A General Neural Network Methodology for Multi-period Portfolio Optimization

by

Chendi Ni

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2024

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	C.W. (Kees) Oosterlee Professor, Mathematics, Utrecht University
Supervisor(s):	Yuying Li Professor, Computer Science, University of Waterloo Peter A. Forsyth Professor Emeritus, Computer Science, University of Waterloo
Internal Member:	Justin Wan Professor, Computer Science, University of Waterloo
Internal Member:	Yaoliang Yu Associate Professor, Computer Science, University of Waterloo
Internal-External Member:	Chengguo Weng Professor, Statistics and Actuarial Science, University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

In this thesis, we propose a neural network methodology for solving the multi-period portfolio optimization problem. Our approach formulates the problem as a stochastic optimal control problem and uses a single neural network model to approximate the optimal control function so that the optimal control can be obtained via solving a single standard optimization problem. Unlike traditional dynamic programming methods, our methodology eliminates the need for high-dimensional expectation evaluations, making it computationally efficient. Moreover, our framework is flexible and not restricted to specific objective functions or data models, allowing it to handle a wide range of portfolio optimization problems. In addition, by designing novel neural network models with suitable activation layers, we ensure that the neural network representation is always a feasible solution to the original optimal control problem so that the complex constrained portfolio optimization problems are converted into computationally feasible unconstrained ones. Furthermore, we provide mathematical proofs that our proposed neural network models are capable of approximating the constrained optimal control arbitrarily well, illustrating the validity and effectiveness of our approach. Finally, through extensive numerical experiments, we demonstrate the empirical performance of our methodology, validating its practical relevance and effectiveness in real-world investment decision-making.

Acknowledgements

I would like to thank:

My supervisors, Yuying Li, and Peter Forsyth, for their support and guidance.

The committee members, Kees Oosterlee, Justin Wan, Yaoliang Yu, and Chengguo Weng, for volunteering their time to participate in my defense.

Past and present SciCom lab members for all the fond memories we shared.

My parents and my wife, Yuhang Ma, for their love and support along the way.

Dedication

To my family.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

"With great risk comes great reward."

Thomas Jefferson

Investing is a balancing act between risk and reward.

A fundamental concept in investing is diversification, which involves spreading investments across various asset classes, industries, and geographic regions to reduce overall portfolio risk. In practice, one simple example of diversification is the constant weight stock-bond portfolio, which allocates a fixed proportion of capital to stocks and bonds. Popular examples include the 50/50 (Graham, 2003; Qian, 2011) and 60/40 portfolios (Ambachtsheer, 1987; Bernstein, 2002).

A constant weight strategy is fundamentally contrarian. When stocks go up in price, they are sold and the proceeds are shifted to the bond portfolio. When stocks decrease in value, bonds are sold, and cash is invested in stocks. This strategy is consequently a "buy low, sell high" policy, which takes advantage of the volatility of stocks.

The allocation between stocks and bonds in a constant-weight portfolio carries significant implications. Stocks, with their potential for growth and capital appreciation, can offer higher returns over the long term but also come with increased volatility and market risk. On the other hand, bonds, with their fixed income structure and relatively lower volatility, provide a lower return but offer greater capital preservation and income stability. The decision regarding the stock-bond split in a constant weight portfolio is crucial in determining the desired overall risk and return profile of the portfolio. Typically, as the allocation to stocks increases, the portfolio achieves higher returns but also possesses greater volatility (Vanguard, 2019).

Constant weight portfolios provide a straightforward method for diversifying risks, making them a popular choice in various applications such as defined contribution (DC) pension plans like 401(k) and RRSP. Additionally, their simplicity and popularity have led many active funds, including some of the largest sovereign wealth funds, to use passive constant weight portfolios as a benchmark for assessing the effectiveness of their active strategies.

For example, the Canada Pension Plan (CPP) uses a base reference portfolio of 85% global equity and 15% Canadian government bonds (CPP Investments, 2022). Another example is the Government Pension Fund Global of Norway (also known as the "Oil Fund") managed by Norges Bank Investment Management (NBIM), which uses a benchmark index consisting of 70% equity index and 30% bond index.[1] The benchmark equity index is constructed based on the market capitalization for equities in the countries included in the benchmark. The benchmark index for bonds specifies a defined allocation between government bonds and corporate bonds, with a weight of 70 percent to government bonds and 30 percent to corporate bonds (Norges Bank, 2022).

However, the excess return that these well-known sovereign wealth funds have achieved over their respective passive benchmark portfolios cannot be described as impressive. In the 2022 fiscal year report, CPP claims to have beaten the base reference portfolio by an annualized 80 bps after fees over the past 5 years (CPP Investments, 2022). On the other hand, NBIM reports a mere average of 34 bps of annual excess return over the benchmark over the last decade (see Table 1.1). It is worth noting that these behemoth funds achieve seemingly meager results by hiring thousands of highly paid investment professionals and spending billions of dollars on day-to-day operations. For example, the CPP 2021 annual report (CPP Investments, 2021) lists personnel costs as CAD 938 million, for 1,936 employees, which translates to average costs of about CAD 500,000 per employee-year.

Year	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	Average
Excess return (%)	0.99	-0.77	0.45	0.15	0.70	-0.30	0.23	0.27	0.74	0.87	0.34

Table 1.1: Norges Bank Investment Management, excess return to benchmark portfolio

The stark contrast between the enormous spending of sovereign wealth funds and the meager outperformance of the funds relative to the passive benchmark portfolios is probably provocative to taxpayers and pensioners who entrust their hard-earned money to these funds. It is indeed difficult for us to comprehend how professional fund managers, given the substantial compensation they receive, are only marginally outperforming a simple constant weight benchmark. After all, considering the active nature of these funds, which allows for a much broader selection of strategies beyond simple constant weight strategies, one would expect more significant outperformance over such benchmarks.

This prompts us to ask the following question: how can we systematically solve the portfolio optimization problem so that an investor can employ a simple asset allocation strategy

[1]The Ministry of Finance of Norway sets the allocation fraction between the equity index and the bond index. It gradually raised the weight for equities from 60% to 70% from 2015-2018.

that consistently beats the benchmark passive portfolios (preferably without spending billions of dollars in personnel costs)?

1.2 Multi-period portfolio optimization

Portfolio optimization problems involve finding the optimal allocation of assets in order to maximize a chosen investment target function. The fundamental work by Markowitz (1952) on optimal single-period mean-variance portfolios establishes the groundwork for modern portfolio optimization.

In practical scenarios, portfolio optimization often takes place over multiple periods, where investors make sequential investment decisions throughout an investment horizon. Consequently, the portfolio optimization framework is extended to accommodate this multi-period setting. In traditional multi-period portfolio optimization literature, a common modeling approach is to assume continuous rebalancing. For instance, Merton (1969) considers a continuous-time asset allocation setting and utilizes stochastic control techniques to derive an optimal portfolio that maximizes a constant relative risk aversion (CRRA) utility function of terminal wealth. Since then, extensive research has been conducted on multi-period portfolio optimization under the continuous rebalancing assumption (Merton, 1971; Browne, 1997; Blanchet-Scalliet et al., 2008; Ang et al., 2014).

However, there are several limitations in these studies. Firstly, while continuous rebalancing assumptions often admit closed-form solutions, the methodologies are typically limited to specific choices of objective functions and require assumptions about the stochastic process models that describe the asset prices. Furthermore, the assumption of continuous rebalancing is not practical.

This thesis focuses on discrete multi-period portfolio optimization problems. Let $[t_0, T]$ denote the investment horizon, and define $\mathcal{T} \subseteq [t_0, T]$ as the rebalancing schedule. Specifically, we consider a discrete schedule with N rebalancing events, i.e.

$$\mathcal{T} = \Big\{ t_0 < t_1 < \cdots < t_N = T \Big\}. \tag{1.1}$$

The multi-period portfolio optimization problem is then formulated as a stochastic optimal control problem, assuming that the control (allocation decision) $\boldsymbol{p}$ is a function of the state variables. The state variable vector at time t_i, denoted by $\boldsymbol{X}(t_i) \in \mathcal{X}$ (the state space), encompasses all relevant information at that time to describe the system. For example, we consider outperforming a benchmark in this thesis, thus the wealth of the actively managed portfolio and the benchmark portfolio serves as state variables. The values of the state variables evolve according to a specific evolution function that depends on the control function, random returns for N_a assets, and investment specifications such as transaction cost, shorting, and cash injections. Mathematically,

$$\boldsymbol{X}(t_i) = f_{evo}\Big(\boldsymbol{X}(t_{i-1}), \boldsymbol{p}(\boldsymbol{X}(t_{i-1})), \boldsymbol{r}(t_{i-1}) \Big), \ \forall i = 1, \cdots, N. \tag{1.2}$$

Here $\boldsymbol{r}(t_{i-1}) \in \mathbb{R}^{N_a}$ is the vector of (random) asset returns from t_{i-1} to t_i. The detailed form of the evolution function f_{evo} relies on the specific investment cases, which will be discussed in detail in later chapters. For now, we assume that all state variables $\boldsymbol{X}(t_i)$ can be computed using (1.2).

Then, the stochastic optimal control problem aims to find the feasible optimal control $\boldsymbol{p}^* : \mathcal{X} \mapsto \mathbb{R}^{N_a}$, under the following optimization problem:

$$\boldsymbol{p}^* = \inf_{\boldsymbol{p} \in \mathcal{A}} \mathbb{E}_{\boldsymbol{p}}^{t_0} \Big[F\big(\boldsymbol{X}(t_0), \cdots, \boldsymbol{X}(t_N)\big) \Big], \tag{1.3}$$

where $\mathcal{A}$ represents the set of feasible controls, and $F(\cdot)$ is an investment performance metric. Notably, the objective function is expressed as an expectation since the state variables are subject to randomness due to the presence of stochastic asset returns. $\mathbb{E}_{\boldsymbol{p}}^{t_0}[\cdot]$ denotes the expectation at time t_0, assuming that the system evolves based on a feasible control $\boldsymbol{p}$.

The literature on solving discrete-time multi-period problems often employs dynamic programming methods, which involve converting the multi-period problem into multiple single-period optimization problems (Mulvey and Vladimirou, 1989; Dantzig and Infanger, 1993; Cariño and Turner, 1998; Cheung and Yang, 2004; Wang and Forsyth, 2010; Dang and Forsyth, 2014). However, for these dynamic programming methods, the computational complexity increases exponentially with the number of state variables and is only tractable when the number of stochastic factors is relatively small.

1.3 The neural network approach

In this thesis, we propose to use a single neural network, denoted as $f_{\boldsymbol{\theta}}$ parameterized by parameter $\boldsymbol{\theta}$, to approximate the control function, i.e. $f_{\boldsymbol{\theta}}(\cdot) \simeq \boldsymbol{p}(\cdot)$. Consequently, we solve a single optimization problem without using dynamic programming.

An important challenge faced by existing literature in multi-period portfolio optimization is the difficulty in handling complex portfolio constraints. The set of feasible controls $\mathcal{A}$ can be intricate, making the optimization problem (1.3) challenging to solve (as we will discuss in detail in later chapters).

A key aspect of our proposed approach is the design of specific neural network models that automatically satisfy the constraints described by $\mathcal{A}$. Instead of tackling the complex constrained problem (1.3), we solve the following parameterized unconstrained optimization problem:

$$\inf_{\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}} \mathbb{E}_{f_{\boldsymbol{\theta}}}^{t_0} \Big[F\big(\boldsymbol{X}(t_0), \cdots, \boldsymbol{X}(t_N)\big) \Big]. \tag{1.4}$$

The solution to the unconstrained problem (1.4) can be easily computed using standard optimization methods such as the stochastic gradient method (SGD) or ADAM (Kingma and Ba, 2014).

It is important to note that, we include time as an input feature to the model, thus making the neural network recurrent. This is different from the stacked neural network approach proposed in Han et al. (2016); Tsang and Wong (2020), which requires a new sub-network at each rebalancing time despite also adopting the idea of approximating the control with neural networks. In comparison, our methodology is more intuitive and computationally efficient. In addition, by including time as an input to the neural network ("global-in-time"), our methodology is suitable for approximating the control functions that are smooth in time.

Recently, researchers have also explored the application of reinforcement learning (RL) methods in multi-period portfolio optimization. For example, Dixon et al. (2020); Park et al. (2020); Lucarelli and Borrotti (2020); Gao et al. (2020) use Q-learning algorithms in which neural networks are used to approximate the action-value function ("Q-function"). However, such methods may be unnecessarily high-dimensional when the state space is relatively low-dimensional (van Staden et al., 2023b).

It is worthwhile emphasizing that much previous work on machine learning (Yang et al., 2018; Lucarelli and Borrotti, 2019; Zhang et al., 2020; Cao et al., 2020; Zhang et al., 2021) for portfolio allocation is based on maximizing common metrics such as the Sharpe ratio. A Sharpe ratio maximizing strategy is the optimal control for a single period mean-variance objective function (Haugh, 2016), but by no means optimal in the multi-period context, where the optimal allocation is dynamic. The approach in this thesis is to specify an objective function, and then determine the optimal control that maximizes this objective function directly.

It is also worth noting that for certain objectives (e.g. the cumulative quadratic difference (CD) objective and the cumulative quadratic shortfall (CS) objective discussed in Chapter 6), our methodology may be formulated to appear similar to policy gradient methods in RL literature (Silver et al., 2014) on a high level. Examples of policy gradient methods in financial problems include Coache and Jaimungal (2021), in which the authors develop an actor-critic algorithm for portfolio optimization problems with convex risk measures. However, there are two main differences between our proposed methodology and policy gradient algorithms. Firstly, we assume that a model (parametric or non-parametric) for the randomness of the environment (i.e., asset returns) over the entire investment horizon is readily available upfront (e.g., through calibration of parametric models or resampling of historical data, which is a common assumption adopted by practitioners when backtesting investment strategies). On the other hand, RL literature often considers an unknown environment, and the algorithms focus on the exploration of the agent to learn from the unknown environment. Such an approach may be unnecessary and complicated for our use case since we do not consider the market impact of rebalancing actions. Secondly, our proposed methodology is not limited to the cumulative reward framework in RL and thus is more general and suitable for problems in which the investment objective cannot be easily expressed in the form of a cumulative reward.

The closest work to this thesis is Li and Forsyth (2019); van Staden et al. (2023b,a). Li and Forsyth (2019) propose a shallow neural network model for representing the allocation

in a long-only DC portfolio but focus on fixed wealth targets. van Staden et al. (2023b) consider a long-only portfolio with the goal of outperforming an elevated target, a concept proposed in Chapter 2 of this thesis. van Staden et al. (2023a) extend the results from van Staden et al. (2023b) by considering the path-dependent outperformance over the elevated target but are still restricted to a long-only setting.

An important advantage of our neural network approach is its flexibility in terms of objective function choices, investment specifications, and data models. Unlike most modeling approaches that are tailored to specific objective functions, investment specifications (e.g. transaction costs, shorting costs, cash injection), or stochastic process data models, our methodology provides a general modeling framework that can adapt to a wide range of investment problems.

Furthermore, we establish mathematical proofs that, under reasonable assumptions about the optimal control which is subject to complex constraints such as bounded leverage, the proposed neural network models are capable of approximating the optimal control arbitrarily well. This provides a theoretical assurance of the validity and effectiveness of the proposed methodology.

Finally, through extensive numerical experiments, we show that the neural network approach produces dynamic strategies that consistently outperform passive benchmarks by specified margins, which showcases the capability of the neural network methodology to achieve desired investment goals and illustrates the immense practical value of the framework.

1.4 Outline

In Chapter 2, we delve into the problem of outperforming a passive benchmark for a long-only defined contribution (DC) pension portfolio. We propose the concept of an elevated target that incorporates both the wealth of the benchmark portfolio and a target outperformance spread and design a novel asymmetric objective function based on this elevated target. We discuss bootstrap resampling as a method of generating data samples and demonstrate its validity by proving the exceedingly low probability of observing repeating sample paths in the data sets. In numerical experiments, we show that the optimal strategy following the learned neural network model achieves desired outperformance over the constant weight benchmark with high probability at the end of the accumulation phase of a pension plan.

In Chapter 3, we provide additional numerical results to examine the robustness of the neural network strategy studied in Chapter 2 from two perspectives. First, we examine the robustness of the neural network methodology with respect to the assumptions and settings of the problem. Particularly, we study the effects of using a different benchmark portfolio (i.e. a target date fund) and an alternative underlying data set, and find that the neural network strategies still achieve the desired outperformance despite the different settings and assumptions. Then, we examine the robustness of the neural network strategy by studying

the performance of the strategy on testing data sets generated from a different distribution from the training data sets. We find that the neural network strategy achieves consistent outperformance over the benchmark portfolio in all scenarios tested.

In Chapter 4, we derive a closed-form solution under certain assumptions to gain insights into the optimal control and find that the closed-form optimal control is contrarian, consistent with observations from numerical results from Chapter 3.

Moving on to Chapter 5, we shift the focus to a multi-period portfolio optimization problem with bounded leverage. Specifically, we aim to replicate the investment behavior of large sovereign wealth funds by allowing the portfolio to have short positions in a selected set of assets, thus enabling leverage in the portfolio. Subsequently, we propose a leverage-feasible neural network (LFNN) model to tackle the optimization problem under leverage constraints. In Chapter 6, we consider a case study that applies the LFNN model to portfolio optimization problems during high inflation regimes and demonstrate that the LFNN model achieves the desired performance outcomes.

In Chapter 7, drawing inspiration from 130/30 portfolios, we consider the portfolio optimization problem with a more relaxed constraint that allows the investment manager to take long or short positions on all assets, as long as the total long position within the portfolio is bounded. To handle the complex constraints involved, we introduce a novel relaxed-constraint neural network (RCNN) model that transforms the constrained optimization problem into an unconstrained one. Moreover, we provide mathematical proof establishing that the RCNN model can effectively approximate the optimal control arbitrarily well. Finally, through numerical experiments, we showcase the advantages of 130/30 portfolios over long-only portfolios, dispelling the misconception that 130/30 portfolios offer little value.

In Chapter 8, we present the concluding remarks for this thesis.

1.5 Contributions

To summarize, this thesis makes the following key contributions:

- We propose a neural network methodology for solving the stochastic optimal control problem in multi-period portfolio optimization. This methodology offers a scalable and efficient alternative to traditional dynamic programming methods, which are computationally intensive.
- The proposed neural network framework is versatile and flexible, as it can be applied to a wide range of investment problems. It is not limited to specific choices of objective functions, investment specifications, or data-generating models, making it a general methodology.

- We develop novel neural network models that convert complex constrained optimization problems into computationally feasible unconstrained standard optimization problems, which enables us to effectively solve traditionally challenging constrained portfolio optimization problems.
- The proposed methodology can be applied to obtain the optimal leveraged portfolios under the multi-period setting, which existing literature is ill-equipped to compute.
- We provide mathematical proofs demonstrating that the proposed neural network models are capable of approximating optimal controls with arbitrary precision. These theoretical validations establish the validity and effectiveness of the proposed approach.
- We propose the concept of the elevated target which leads to the creation of novel objective functions that guide the neural network model to achieve empirically desired relative performance compared to the benchmark portfolio.
- We demonstrate the validity of using bootstrap resampling as a data-generating method by proving the exceedingly low probability of having repeating sample paths in the training and testing data sets.
- Through extensive numerical experiments, we showcase the efficiency and effectiveness of neural network-based strategies in various realistic investment scenarios. These empirical results validate the practical importance and relevance of the proposed methodology in real-world investment decision-making.

Chapter 2

Optimal asset allocation for outperforming a stochastic benchmark target

2.1 Introduction

In this chapter, we introduce a neural network methodology for solving the problem of outperforming a stochastic target.

We begin by introducing the outperformance problem and discussing the current literature on this topic. We then focus on outperformance in the context of outperforming a passive benchmark at the end of the accumulation phase of a defined contribution (DC) pension plan and formally define the outperformance problem.

We propose an objective function for this problem based on the concept of an elevated target. In order to penalize risky strategies, we use an asymmetric weighting objective function.

Since we consider a long-only portfolio, the portfolio optimization problem is essentially an asset allocation problem, in which we need to find the optimal allocation strategy that optimizes the objective function.

We formulate this optimal asset allocation problem as a stochastic optimal control problem where the control represents the allocation strategy. We then propose to use a neural network to approximate the control so that we can obtain the optimal control via solving a standard optimization problem without dynamic programming.

Since we can observe only a single historical path of the capital markets (i.e. the realized historical asset price path), it is necessary to augment this data for the training and testing of the neural network model. We discuss the use of stationary block bootstrap resampling (Politis and Romano, 1994) for generating the training and testing data sets. We then prove

that, for a reasonable choice of bootstrap blocksizes, the probability of observing repeating paths is extremely low. This means that overfitting (i.e. strategies based on "memorizing" specific historical asset price paths in the training set) is extremely unlikely.

In numerical experiments, we consider a 30-year investment horizon with a benchmark portfolio consisting of 50% stocks (a broad stock market index) and 50% bonds (based on a short-term U.S. treasury index) rebalanced annually. We show that the learned adaptive (dynamic) neural network strategy yields significant outperformance compared to the benchmark.

We remark that this chapter is based on the published work of Ni et al. (2022). In summary, in this chapter, we make the following contributions:

- We propose a neural network methodology for solving the outperformance problem under a multi-period setting. This methodology offers a scalable and efficient alternative to traditional dynamic programming methods, which are computationally intensive.
- We propose the concept of an elevated target which leads to the creation of novel objective functions that guide the neural network model to achieve empirically desired relative performance compared to the benchmark portfolio.
- We demonstrate the validity of using bootstrap resampling as a data-generating method by proving the exceedingly low probability of having repeating sample paths in the training and testing data sets.
- In numerical experiments, we demonstrate the efficiency of the proposed neural network by showing that the learned adaptive strategy outperforms the stochastic benchmark target by the specified margin at the end of the accumulation phase of a pension plan. Furthermore, by following the adaptive strategy, an investor can expect to achieve a sufficient replacement rate for retirement.

2.1.1 The outperformance problem

While most existing work on multi-period asset allocations has focused on achieving optimal absolute performance, the allocation problem with the goal of achieving relative outperformance has significant practical importance. This is because, in practice, the performance of a portfolio is often evaluated not only by its absolute performance but also against other benchmark portfolios. Multi-period asset allocation with the goal of optimizing relative performance was first studied by Browne (1999, 2000), in which it is assumed that asset prices follow geometric Brownian motions. Under these assumptions, Browne (1999, 2000) derive closed-form optimal portfolios so that the performance relative to a stochastic benchmark is maximized. The author also considers different investment objectives, such as minimizing the expected time to reach a performance goal and maximizing the utility of relative wealth.

Subsequently, the benchmarked asset allocation problem has been further studied from various perspectives, see, e.g., (Tepla, 2001; Basak et al., 2006; Davis and Lleo, 2008; Lim and Wong, 2010; Bajeux-Besnainou et al., 2013). Tepla (2001) studies the problem of an expected utility maximizing investor with the goal of performing at least as well as a stochastic benchmark. Basak et al. (2006) relax the minimum performance constraints used in Tepla (2001) and a certain shortfall probability is allowed in return for some upside potential. Bajeux-Besnainou et al. (2013) introduce a downside hedging constraint and includes the benchmark in the objective function in a mean-variance framework while avoiding unrestricted losses.

Instead of the classical stochastic control approach, Davis and Lleo (2008) use a risk-sensitive control approach to study the benchmarked asset allocation problem, in which the benchmark follows a variant of the geometric Brownian motion. Lim and Wong (2010) consider more generic price dynamics and general increasing concave objective functions.

More recent studies include Oderda (2015), Al-Aradi and Jaimungal (2018) and Al-Aradi and Jaimungal (2021). In Oderda (2015), under the assumption that stocks follow a geometric Brownian motion and no investing constraints (i.e. infinite leverage and shorting are allowed), the author shows that a portfolio that outperforms the benchmark market capitalization index (under certain criteria) can be constructed by a combination of (i) the benchmark portfolio and (ii) rule-based portfolios, e.g., equal weight and minimum variance portfolios. The determination of the optimal weights for these portfolios is independent of estimates of the expected returns of individual stocks. Hence this outperformance portfolio is robust to uncertainty in the expected return parameters. In Al-Aradi and Jaimungal (2018), optimal stochastic control techniques are also used in this context. Based on several assumptions on the asset return process, Al-Aradi and Jaimungal (2018) formulate the control problem as a Hamilton-Jacobi-Bellman (HJB) partial differential equation (PDE) and are able to obtain a closed-form solution. In Al-Aradi and Jaimungal (2021), the authors assume that the growth rate is stochastic and driven by latent factors, which addresses the shortcoming of assuming a deterministic market return in Al-Aradi and Jaimungal (2018). We remark that all this work is in a continuous-time setting with unconstrained controls. To the best of our knowledge, little work is done on discrete-time rebalancing multi-period asset allocation (with constraints) that focuses on relative performance compared to a benchmark.

A common limitation of the previous work which focuses on outperforming a stochastic benchmark is the lack of consideration of realistic constraints such as no-leverage and no-shorting. Such constraints make finding closed-form solutions difficult, if not impossible. One possible solution is to numerically solve the problem by following the methodology proposed in Dang and Forsyth (2014), in which constraints such as no-shorting, no-leverage, and discrete rebalancing are considered. Dang and Forsyth (2014) propose a method that uses dynamic programming to establish an associated Hamilton-Jacobi-Bellman (HJB) equation that generates the optimal portfolio. However, a numerical HJB equation solution is only practical if there are a small (three or fewer) number of state variables ($\leq$3). In Dang and Forsyth (2014), under discrete rebalancing with two assets and no benchmark strategy, the HJB equation is of dimension two. Note that if discrete rebalancing is assumed, it is not possible to reduce this to a one-dimensional PDE. However, under discrete rebalancing

with two assets and a two-asset benchmark strategy, the PDE problem has four dimensions, as four state variables are needed to represent the amount in each asset for each strategy, between rebalancing times. Similarly, under discrete rebalancing with n assets and a n asset benchmark setting, the problem has $2n$ dimensions. Existing methods (Wang and Forsyth, 2010; Dang and Forsyth, 2014) that convert the problem into an HJB equation are not practical in these cases.

Another common issue with existing approaches is the assumption of parametric stochastic models for asset returns. This, of course, adds challenges, as the parameters can be difficult to estimate accurately (Black, 1993).

2.1.2 Defined contribution (DC) investment plan

In this chapter, we consider a practically relevant and important problem: optimal multi-period asset allocation during the accumulation phase of a DC pension plan. A defined contribution (DC) plan is a retirement plan in which the employer, employee, or both make contributions at regular intervals (e.g. annually). There is no guarantee on the accumulated amount in the plan at the retirement date. In contrast, another type of retirement plan is the defined benefit (DB) plan, which promises to pay a set income when the employee retires. There has been a paradigm shift from DB plans to DC plans in the United States, Canada, the United Kingdom, and Australia, as both the public and private sectors are no longer willing to take on the risks of DB plans. By 2027, all DB plans in the Netherlands will be shifting to DC plans (Pielichata, 2023).

We note that the employee is the investor in the DC plan since she is exposed to the risks of the chosen investment portfolio. In a DC plan, the employee (investor) is often presented with a list of eligible stock and bond funds and then needs to specify how the DC account is to be allocated to each fund. Typically the employee has the opportunity to make contributions to the DC plan (usually a certain percentage of the salary) and change the asset allocation at least yearly. Normally, the DC plan is tax-advantaged, so that there are no tax consequences triggered by the rebalancing action.

In this chapter, we assume the investment horizon for the DC plan is 30 years. Studies have shown that income for a typical employee increases rapidly until the age of 35, then remains mostly unchanged (in real terms) until a few years before retirement, and then decreases due to fewer working hours during the transition to retirement (Cocco et al., 2005; Rupert and Zanella, 2015).

Since total (employee-employer) DC plan contribution is often proportionally tied to overall income, we believe a 30-year investment horizon is reasonable and captures the most stable period in terms of income for a typical employee, during which she can save for retirement most consistently. We remark that the 30-year time horizon is also commonly used in literature in the field of pension studies (O'Donoghue and Rabin, 1998; Booth, 2004; Malliaris and Malliaris, 2008; Looney and Hardin, 2009; Levy, 2016; Blanchett et al., 2017;

Basu and Wiafe, 2017; Brown et al., 2017; Blanchett et al., 2018; Estrada and Kritzman, 2019; Wiafe et al., 2020).

Recently, a popular choice for DC pension investment has been target date funds, in which the investor sets a retirement date and the fund aims to meet certain financial return objectives at the given retirement date. Usually, target date funds take a glide path approach that "glides down" towards a more conservative combination of assets towards the target date (Balduzzi and Reuter, 2012). In a two-asset case of a stock index and a bond index, the glide path strategy often decreases the stock holding over time. Another popular asset allocation strategy for DC plans is the constant proportion strategy, in which the employee invests fixed proportions of the wealth into several assets. This idea can be traced back to Graham (2003). It is shown in (Graf, 2017; Forsyth and Vetzal, 2019) that the final wealth distributions of a constant weight allocation, and any glide path strategy having the same average allocation as the constant weight strategy, are essentially the same. Hence there is little to be gained by using a (deterministic) glide path compared to a constant weight strategy. This theoretical analysis is backed by empirical studies (Basu et al., 2011; Arnott et al., 2013; Esch and Michaud, 2014). We also provide empirical evidence in Section 3.1 to support this argument. Therefore, in this chapter, we set the benchmark target to be the constant proportion strategy as it is easy to understand and implement.

Among the constant proportion strategies, a very popular one is the 50/50 strategy, in which 50% of the wealth is allocated to stocks and 50% of the wealth is allocated to bonds. It is conventional wisdom that a 50/50 portfolio is an appropriate tradeoff between risk and reward for those saving for retirement. Although there has been a popular shift to a 60/40 portfolio (60% in stocks) in recent years, for illustration, we will focus on the 50/50 portfolio in this chapter. In fact, the 50/50 portfolio is recommended by Benjamin Graham in his famous book "The Intelligent Investor" (Graham, 2003) for investors who cannot spare time for deeper investigation into tactical asset allocation. It is also worth noting of the practical applicability of considering a more aggressive benchmark choice, which may result in a riskier allocation strategy with worse tail risk (van Staden et al., 2023a).

We remark that the reason why we only consider two assets is two-fold. Firstly, in practice, retail investors are often choosing between a stock fund and a bond fund. Secondly, the popular constant proportion strategies often only involve two assets. However, we note that the proposed framework is readily applicable to more assets.

Finally, we remark that the stylized DC plan accumulation problem in this chapter is a simplified version of the real-world investment scenario. When making an investment decision in practice, an individual investor will inevitably need to consider some important factors, e.g., medical expenditures, taxes, housing expenses and labor income, and other financial assets, see Duarte et al. (2021).

2.1.3 Optimization with an elevated target

Let the initial time $t_0 = 0$ and consider the rebalancing schedule $\mathcal{T}$:

$$\mathcal{T} = \{t_0 = 0 < t_1 < \ldots < t_N = T\}. \tag{2.1}$$

The fraction of total wealth allocated to each asset is adjusted at times t_n, $n = 0, \ldots, N-1$, with the investment horizon $t_N = T$. We consider an investment problem in N_a assets, an active portfolio, and a benchmark portfolio.

Let $W(t_n)$ and $\hat{W}(t_n)$ denote the wealth of the active portfolio and the benchmark portfolio after cash injections at time t_n. To mimic the typical DC plan account, we assume that $W(t_0) = \hat{W}(t_0) = c(t_0)$, where $\{c(t_n),\ n = 0, \cdots, N\}$ represent an *a priori* specified cash injection schedule which mimics the periodic DC plan contributions.

We denote the allocation at t_n by an allocation vector $p(t_n),\ n = 0, \ldots, N-1$. Given the allocation control vectors $p(t_0), p(t_1), \ldots, p(t_{N-1})$, the statistical properties of the terminal wealth of the adaptive portfolio $W(T)$ can be determined. Similarly, given a benchmark allocation vectors $\hat{p}(t_0), \hat{p}(t_1), \ldots, \hat{p}(t_{N-1})$, the terminal wealth of the benchmark portfolio $\hat{W}(T)$ can also be determined. The time evolution of $W(t)$ and $\hat{W}(t)$ is given by

$$\begin{cases} W(t_{n+1}) = \Big(\sum\limits_{i=1}^{N_a} p_i(t_n) \cdot R_i(t_n)\Big) W(t_n) + c(t_{n+1}), \\ \hat{W}(t_{n+1}) = \Big(\sum\limits_{i=1}^{N_a} \hat{p}_i(t_n) \cdot R_i(t_n)\Big) \hat{W}(t_n) + c(t_{n+1}), \end{cases} \quad n = 0, \cdots, N-1, \tag{2.2}$$

where $R_i(t_n)$ is the return on asset i from t_n to t_{n+1}, and $p_i(t_n), \hat{p}_i(t_n)$ represent the allocation fraction to asset i for the active portfolio and the benchmark portfolio.

Our goal is to design an objective function that produces an adaptive allocation strategy (which dynamically allocates wealth over time) with a superior terminal wealth distribution than the benchmark strategy. The first idea is to minimize some measure of underperformance against the benchmark. A natural choice is to quadratically penalize the underperformance of the terminal wealth of the adaptive strategy compared to a benchmark of a fixed terminal wealth, as in Li and Forsyth (2019). Note, however, that in our case, the benchmark target (i.e. the terminal wealth of the benchmark portfolio) is stochastic. This leads to the following optimization problem ($\mathbb{E}[\cdot]$ is the expectation operator):

$$\min_{p(t_0), p(t_1), \ldots, p(t_{N-1})} \mathbb{E}\left[\min\left(W(T) - \hat{W}(T), 0\right)^2\right]. \tag{2.3}$$

Note that underperformance is quadratically penalized, which will discourage allocation to extremely risky assets. Unfortunately, an optimal solution to (2.3) is trivially the benchmark strategy, i.e. $p(t_n) = \hat{p}(t_n), \forall n$, which indicates the formulation (2.3) does not sufficiently capture the properties of the desired solution.

We propose to generate a more ambitious strategy by including an *elevated target*, $e^{sT} \cdot \hat{W}(T)$, in the objective function, i.e.,

$$\min_{p(t_0),p(t_1),\ldots,p(t_{N-1})} \mathbb{E}\left[\min\left(W(T) - e^{sT} \cdot \hat{W}(T), 0\right)^2\right]. \tag{2.4}$$

Here s is the yearly pre-determined target outperformance spread. Consequently, in an ideal case, the adaptive strategy will have a terminal wealth of $e^{sT} \cdot \hat{W}(T)$ which indicates that the adaptive strategy achieves an annual outperformance spread of return s compared to the benchmark strategy.

However, due to the lack of penalty on the outperformance over the elevated target, (2.4) still tends to yield fairly extreme risk-taking strategies, which results in large excessive terminal wealth as well as large left tails of the terminal wealth distribution. To address this undesirable situation, we introduce penalties for the outperformance of the elevated target.

One natural choice of including penalties on the outperformance part would be a symmetric quadratic penalty objective, i.e.

$$\min_{p(t_0),p(t_1),\ldots,p(t_{N-1})} \mathbb{E}\Big[\left(W(T) - e^{sT} \cdot \hat{W}(T)\right)^2\Big]. \tag{2.5}$$

However, (2.5) penalizes both the outperformance and the underperformance relative to the elevated target equally. In practice, outperformance is preferred over underperformance. Therefore, we introduce the following asymmetric objective, with a linear penalty on outperformance, and a quadratic penalty on underperformance.

$$\min_{p(t_0),p(t_1),\ldots,p(t_{N-1})} \mathbb{E}\left[\Big(\min\left(W(T) - e^{sT} \cdot \hat{W}(T), 0\right)\Big)^2 + \max\left(W(T) - e^{sT} \cdot \hat{W}(T), 0\right)\right]. \tag{2.6}$$

Figure 2.1 illustrates this asymmetric distribution shaping objective function.

By designing this asymmetric objective function, we not only avoid the extreme risk-taking behavior of the strategies but also incentivize a more right-skewed terminal wealth distribution compared to symmetric quadratic penalties as in (2.5), due to the more lenient penalty on outperformance than underperformance. This will be empirically verified by numerical experiments. In numerical tests (details in Appendix A.1), we find that the seemingly minor difference in the asymmetric penalties indeed yields more favorable terminal wealth distributions than the symmetric quadratic penalty objective function (2.5), suggesting that (2.6) is the more suitable objective function to use.

While we choose the objective function (2.6) for outperforming a stochastic target in this chapter, we note that distribution shaping objectives can be problem-dependent. If an investor is concerned with left tail risk, then it may be appropriate to use an objective function that minimizes Conditional Value at Risk (CVaR), for example, see Forsyth and

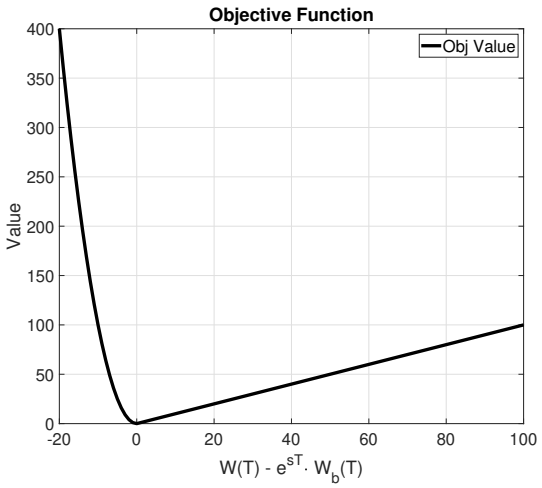


Figure 2.1: Asymmetric distribution shaping objective function with elevated target $e^{sT} \cdot \hat{W}(T)$.

Vetzal (2019); Forsyth (2022). If an investor is concerned with path-dependent performance measures such as draw-down and variation over time, then such measures should be incorporated into the objective function. For example, the quadratic variation penalty used in Al-Aradi and Jaimungal (2018), which is the time-averaged instantaneous volatility relative to a benchmark, can be introduced to penalize the deviation from the benchmark portfolio on a running basis. In fact, in later chapters of this thesis, we will discuss examples with path-dependent objective functions.

The above discussions mainly illustrate the challenges in specifying the appropriate objective functions for obtaining a desirable investment outcome, which is of practical importance to investors. One attractive property of the proposed neural network approach is that it can accommodate any continuously differentiable objective function. In practice, this allows freedom in designing proper objective functions to shape the wealth distribution of the strategies.

2.2 Neural network approach for solving the optimization problem

If we postulate parametric stochastic processes for prices of the traded assets, mathematically, the controls $p(t_0), \cdots p(t_{N-1})$ can be determined using dynamic programming via solving a non-linear HJB PDE (e.g. Al-Aradi and Jaimungal (2018)). In the absence of any closed-form solution, computing a solution to this problem numerically would be costly, particularly when the problem has a high dimension. Consider the simplest allocation problem, for which the portfolio consists of a stock index and a bond index. In the case of discrete rebalancing, the state variables would be the dollar amounts in the bond and stock indices,

for both the adaptive and benchmark portfolios (Dang and Forsyth, 2014). Consequently, even for this comparatively simple case, this would result in a four-dimensional HJB PDE.

Alternatively, recent advancement in reinforcement learning (RL) algorithms provides another direction for numerically solving the controls using dynamic programming. For example, Dixon et al. (2020); Park et al. (2020); Lucarelli and Borrotti (2020); Gao et al. (2020) use Q-learning algorithms to solve the discrete-time multi-period optimal allocation problem. In general, if there are N_a assets to invest in, then the use of Q-learning involves approximation of an action-value function ("Q" function) which is a $(2N_a + 1)$-dimensional function (van Staden et al., 2023b) which represents the conditional expectation of the cumulative rewards at an intermediate state.[1] Meanwhile, the optimal control is a mapping from the state space to the allocation fractions to the assets. If the state space is relatively low-dimensional, then the DP-based approaches are potentially unnecessarily high-dimensional.

Assume that samples of asset returns are available. These samples can come directly from market observations or from simulations of postulated parametric models. Instead of solving $p(t_0), \ldots, p(t_{N-1})$ using dynamic programming, we propose a data-driven approach as follows. We represent the optimal control as a function of feature vector $\boldsymbol{X}(t)$ (of state variables), i.e., at t_n,

$$p(t_n) = p(\boldsymbol{X}(t_n))$$

Example 2.2.1 (Two asset problem with 50/50 benchmark)**.** *In the numerical examples, we will focus on portfolios consisting of two assets: a stock index and a bond index. The benchmark portfolio in this case will be a constant proportion strategy, with* 50% *stocks and* 50% *bonds. We will denote the wealth of the benchmark strategy in this case as* $W_{50/50}(t)$*. For this example, for the stochastic target pension allocation problem, we use three features:*

(i) $W(t)$*, the wealth of the adaptive portfolio at* t*.*

(ii) $W_{50/50}(t)$*, the wealth of the constant proportion portfolio at* t*.*

(iii) $T - t$*, time remaining in the investment period.*

Therefore, the feature vector $\boldsymbol{X}(t) = \big(W(t), W_{50/50}(t), t\big)$*. In the cases that stochastic processes such as multi-dimensional jump-diffusion processes are assumed, then it can be shown (in the absence of transaction costs) that the controls are only a function of these features (Dang and Forsyth, 2014), i.e.* $\boldsymbol{X}(t)$ *represent the state variables.*

We remark that the feature vector $\boldsymbol{X}(t)$ for Example 2.2.1 is different from the features in Samo and Vervuurt (2016) which explicitly include security prices. Instead, at time t, the feature set consists of the accumulated wealth at t from allocation strategy and benchmark strategy, which depends on the returns of traded assets from prior periods. Traded asset

[1] Intuitively, the dimensionality comes from tracking the allocation in the N_a assets for both the active portfolio and benchmark portfolio when evaluating the changes in wealth of both portfolios over one period in the action-value function.

prices are not directly used as features for the neural network model. This is essentially because the stochastic model for asset prices is assumed to be available upfront, and that the market impact from rebalancing of the portfolio is negligible. In addition, since we evaluate the performance of a trading strategy based on the terminal wealth $W(T)$, the trading decision at time t only depends on the current accumulated wealth and return distribution of future trading periods. Unless the asset price has predictability in its future return, including the prices as features is redundant in this context and will likely lead to overfitting of the model. However, in such scenarios, additional investment signals such as price momentum can be incorporated as additional features to our neural network model without technical challenge.

It is worth noting some recent literature also proposes to approximate the optimal control function by neural network functions directly (Han et al., 2016; Tsang and Wong, 2020). However, they propose a stacked neural network approach that essentially uses a sub-network to approximate the control at every rebalancing step. Therefore, the number of neural networks required grows linearly with the number of rebalancing periods, which is computationally inefficient compared to our method which only involves training one single neural network. In addition, the stacked neural network method may not be suitable for approximating control functions that are smooth in time.

In this chapter, we use a two-layer neural network as the functional form to approximate the optimal control. As a result, the goal of the optimization problem is to find the optimal parameters of this shallow neural network.

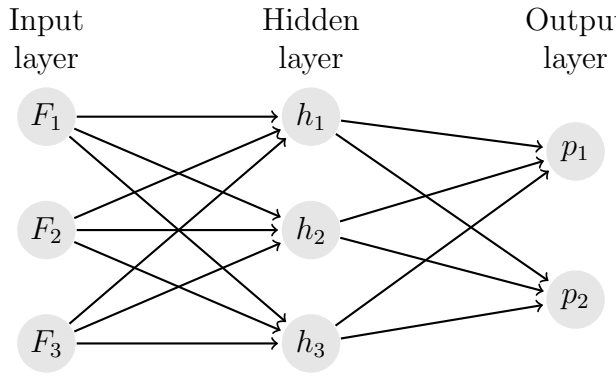


Figure 2.2: A 2-Layer NN representing the control functions

Assume that $h \in R^H$ is the output of the hidden layer. Let the matrix $z \in R^{DH}$ be the weights from the input features $\boldsymbol{X}(t_n) \in R^D$ to the hidden nodes h. We use the sigmoid activation function,

$$\sigma(u) = \frac{1}{1 + e^{-u}} \ ,$$

and have

$$h_j(\boldsymbol{X}(t_n)) = \sigma(\boldsymbol{X}_i(t_n) z_{ij}) \ .$$

Here we use the double summation convention for simplicity, i.e.

$$\boldsymbol{X}_i(t_n)z_{ij} \equiv \sum_{i=1}^{D} \boldsymbol{X}_i(t_n)z_{ij},\ j = 1, ..., H\ .$$

At the output layer, we use the softmax function as the activation function. Let the matrix $x \in R^{HN_a}$ be the weights for the output layer. For asset m, the asset allocation on this asset is given by:

$$p_m(\boldsymbol{X}(t_n)) = \frac{e^{x_{km}h_k(\boldsymbol{X}(t_n))}}{\sum_i e^{x_{ki}h_k(\boldsymbol{X}(t_n))}},\ 1 \le m \le N_a\ .$$

Note that with the softmax activation function, the following constraint is automatically satisfied

$$0 \le p(\boldsymbol{X}(t_n)) \le 1,\ 1^\top p(\boldsymbol{X}(t_n)) = 1\ .$$

This enforces the constraints of no shorting and no leverage. In addition, insolvency cannot occur.

The dynamics of the terminal wealth of the adaptive portfolio then becomes

$$\begin{cases} W(t_{n+1}) = \Big(\sum\limits_{i=1}^{N_a} p_i(\boldsymbol{X}(t_n)) \cdot R_i(t_n)\Big)W(t_n) + c(t_{n+1}), \\ \hat{W}(t_{n+1}) = \Big(\sum\limits_{i=1}^{N_a} \hat{p}_i(t_n) \cdot R_i(t_n)\Big)\hat{W}(t_n) + c(t_{n+1}), \end{cases} \quad n = 0, \cdots, N-1. \tag{2.7}$$

We approximate the expectation in equation (2.6) by a finite number of wealth samples of $W(T)$, computed from return samples of $R(t_n)$ obtained by bootstrapping the historical data. Let $W^\ell(T), \hat{W}^\ell(T)$ be the final wealth samples for the adaptive and benchmark strategies, obtained using equation (2.7), along the ℓ^{th} return sample path $R^\ell(t_n),\ n = 0, 1, \ldots, N-1$.

Denote

$$g(x) \equiv \min\big(x, 0\big)^2 + \max\big(x, 0\big)\ . \tag{2.8}$$

The expectation in equation (2.6) is approximated by

$$\mathbb{E}\Big[g(W(T) - e^{sT} \cdot \hat{W}(T))\Big] \simeq \frac{1}{L}\sum_{\ell=1}^{\ell=L} g(W^\ell(T) - e^{sT} \cdot \hat{W}^\ell(T)) \tag{2.9}$$

Since the approximate function on the right-hand side of (2.9) is a nonconvex, continuous but piecewise differentiable function of the NN weights, solving the optimization problem is challenging.

We recognize however that $\mathbb{E}\Big[g(W(T) - e^{sT} \cdot \hat{W}(T))\Big]$ is a continuously differentiable function of the NN weights assuming that the return distribution is continuous. This motivates us to use the smoothing technique from Alexander et al. (2006). Instead of using

the non-smooth objective function (2.6), in equation (2.9), we replace $g(x)$ by the smoothed approximation $\bar{g}(x)$, to obtain a continuously differentiable approximation,

$$\bar{g}(x) = \begin{cases} x, & \text{if } x > \epsilon \ , \\ \frac{x^2}{4\epsilon} + \frac{1}{2}x + \frac{1}{4}\epsilon, & \text{if } -\epsilon \leq x \leq \epsilon \ , \\ (x+\epsilon)^2, & \text{if } x < -\epsilon \ , \end{cases} \tag{2.10}$$

where ϵ is a predetermined small number. Since we are essentially optimizing the parameters x and z, we write the final problem as

$$\min_{x,z} \ \frac{1}{L} \sum_{\ell=1}^{\ell=L} \bar{g}(W^\ell(T) - e^{sT} \cdot \hat{W}^\ell(T)) \ . \tag{2.11}$$

Similar to Li and Forsyth (2019), we use the trust region optimization method (Coleman and Li, 1996) to solve the resulting optimization problem.

We note that problem (2.11) is an unconstrained optimization problem with $H(D + N_a)$ variables, i.e., the entries of the parameter matrices x and z. More specifically, the optimization method requires the evaluation of the objective function, its derivative with respect to the weight parameters x and z, and the Hessian matrix. Each objective function evaluation costs $\mathcal{O}(H(D+N_a)NL)$, or $\mathcal{O}(L)$ assuming a fixed NN model structure and fixed rebalancing schedule.

For the gradient evaluation, we note that

$$\nabla_{x,z} \left(\frac{1}{L} \sum_{\ell=1}^{\ell=L} \bar{g}(W^\ell(T) - e^{sT} \cdot \hat{W}^\ell(T) \right) \tag{2.12}$$

$$= \frac{1}{L} \sum_{\ell=1}^{\ell=L} \nabla_{W^\ell(T)} \bar{g} \cdot \nabla_{x,z} W^\ell(T). \tag{2.13}$$

Since $\nabla_{W^\ell(T)} \bar{g}$ is a fixed number and only requires constant effort, when calculating the computational cost, we only care about $\nabla_{x,z} W^\ell(T)$. We note the following induction relationship:

$$\nabla_{x,z} W^\ell(t_{n+1}) \tag{2.14}$$

$$= \nabla_{x,z} \Big(p(t_n)^\top R^\ell(t_n) \big(W^\ell(t_n) + q(t_n) \big) \Big) \tag{2.15}$$

$$= \big(\nabla_{x,z} p(t_n) \big)^\top \cdot R^\ell(t_n) W^\ell(t_n) + p(t_n)^\top R^\ell(t_n) \cdot \big(\nabla_{x,z} W^\ell(t_n) \big). \tag{2.16}$$

Since the computational cost of evaluating $\nabla_{x,z} p(t_n)$ is $\mathcal{O}(H(D+N_a))$, we know from (2.14) that the computational cost of evaluating $\nabla_{x,z} W^\ell(T)$ is $\mathcal{O}(H(D+N_a)N)$. Therefore, the total computational cost for evaluating all gradients over L paths is $\mathcal{O}(H(D+N_a)NL)$.

Thus the cost of evaluating the gradient is $\mathcal{O}(H(D+N_a)NL)$. The Hessian matrix used in the optimization is evaluated numerically using the finite difference method and thus has the computational cost of $\mathcal{O}(H^2(D+N_a)^2NL)$. Given the objective function/gradient/Hessian matrix, solving the trust region sub-problem requires $\mathcal{O}(H^3(D+N_a)^3)$. Since we are proposing a shallow network approach, $H(D+N_a)$ is often small. For example, in the two-asset example presented in this chapter, $H(D+N_a) = 15$, and thus the objective function/gradient/Hessian evaluations become the dominant cost and the trust region method optimization evaluation cost is negligible.

2.3 Non-parametric data bootstrap resampling

Success in data-driven learning critically depends on the efficient use of data. Standard machine learning measures success based on testing the model performance on unseen data which are assumed to have the same distribution as the training data. In other words, test results are typically computed based on test samples from the same distributions as training samples.

Unfortunately, for the training of the optimization problem (2.11), we only have access to a single path of historical returns. This lack of data presents a unique challenge in data-driven financial model learning.

For financial model learning and testing, it is a common practice to train and test strategy performance by splitting the historical market data path into two segments - one for training and the other for testing. A critical problem in this approach is insufficient data for robust learning and testing. This is especially problematic in the context of pension planning due to the long-term investment horizon.

Li and Forsyth (2019) use block bootstrap resampling to generate training and testing data in data-driven financial decision learning. Standard block bootstrap resampling is done by dividing the historical market sequential data into blocks with fixed blocksizes and randomly choosing blocks to construct the bootstrap resampled data series. To reduce the impact of a fixed blocksize and to mitigate the edge effects at each block end, the stationary block bootstrap (Patton et al., 2009; Politis and White, 2004) can be used. A single bootstrap resampled path is constructed as follows.

- First, randomly select a block of the historical market data time series with replacement. The blocksize is randomly sampled from a shifted geometric distribution with an expected blocksize $\hat{b}$. The optimal choice for $\hat{b}$ is determined using the algorithm described in (Patton et al., 2009).
- Repeat the previous step and concatenate the new block after the existing data series until the new resampled path has reached the desired length.

- If the selected block exceeds the range of historical data, wrap around the historical data as in the circular bootstrap method (Politis and White, 2004; Patton et al., 2009).

Algorithm 2.3.1 presents the pseudocode for the stationary block bootstrap method.

Algorithm 2.3.1: Pseudocode for stationary block bootstrap

```
/* initialization                                                    */
bootstrap_samples = [ ];
/* loop until the total number of required samples are reached       */
while True do
    /* choose random starting index in [1,...,N], N is the index of the
       last historical sample                                        */
    index = UniformRandom( 1, N );
    /* actual blocksize follows a shifted geometric distribution with
       an expected value of exp_block_size                           */
    blocksize = GeometricRandom( 1 / exp_block_size );
    for i = 0; i < blocksize; i = i + 1 do
        /* if the chosen block exceeds the range of the historical data
           array, do a circular bootstrap                            */
        if index + i > N then
            bootstrap_samples.append( historical_data[ index + i - N ] );
        else
            bootstrap_samples.append( historical_data[ index + i ] );
        end
        if bootstrap_samples.len() == number_required then
            return bootstrap_samples;
        end
    end
end
```

In Li and Forsyth (2019), the training data set is generated using stationary block resampling with one expected blocksize, and the testing data set is generated with a different expected blocksize. As Politis and Romano (1994) point out, changing the expected blocksizes for block bootstrap resampling essentially changes the distribution of the bootstrap resampled data paths. Consequently, such training and testing assessments actually perform out-of-distribution tests.

Intuitively, using the block bootstrap resampling time-series financial market data also seems natural. We have trained a model, considering all permutations of the financial market data with respect to different and random concatenations of time horizons. In addition, testing has been performed on a different sampling distribution of the financial market random horizon concatenations, since the testing data uses a different expected blocksize from

that of the training data. Indeed, evaluating testing performance in this fashion seems to uphold a more stringent standard in comparison to the standard machine learning approach which evaluates testing performance assuming (unseen) testing samples are from the same distribution of the training data.

2.3.1 Probability of repeating paths

Still, one may have concerns that when the training data and testing data are block bootstrap resampled from the same underlying historical market data sequence, one path may appear in both training and testing data sets so that the model may benefit from such an unfair advantage. To address such concerns, we establish a theoretical bound on the probability of training and testing sample sequences being exactly the same and show that the empirical probability is exceedingly low in the experiments.

For a path $\mathcal{P}$, we use the following notations:

$$\begin{aligned} \hat{b} &= \text{expected blocksize in stationary block bootstrap} \\ N &= \text{number of total datapoints in the path} \\ N_{tot} &= \text{number of total datapoints to bootstrap from} \\ \mathcal{P}[i] &= \text{the } i\text{th data point in path } \mathcal{P} \end{aligned} \tag{2.17}$$

We also introduce the following definitions.

Definition 2.3.1. *Assume that a path $\mathcal{P}$ of length N, which contains blocks $[B_1, \dots B_k]$, is resampled from the original data path of length N_{tot}. The* ***decision index list*** *$[I_1, \dots, I_k]$ of the path $\mathcal{P}$ is defined as the list of starting indices of every block in the resampled path with $I_1 = 1$, $I_i = 1 + \sum_{j=1}^{i-1} |B_j|$, $i = 2, \dots k$, where $|B_j|$ denotes the number of points in the block B_j. If I_k is the starting index of the last block in the path, then, for index completeness, we define $I_{k+1} \equiv N + 1$.*

Remark 2.3.1 (Decision index list example)**.** Given a decision index list $[I_1, \dots, I_k]$, associated with a path $\mathcal{P}$, then the data point of the path, which starts at decision index I_i, is $\mathcal{P}[I_i]$.

Definition 2.3.2. *For any two paths $\mathcal{P}_1$ and $\mathcal{P}_2$ of length N, the* ***combined decision index list*** *of $\mathcal{P}_1$ and $\mathcal{P}_2$ is the merged index list (with only a single copy of each index) of the decision index lists of $\mathcal{P}_1$ and $\mathcal{P}_2$. The merged list $[I_1, \dots, I_p]$ retains the order properties of the original lists, i.e. $I_{i+1} > I_i$ and $I_{p+1} = N + 1$.*

Definition 2.3.3. *For any two paths $\mathcal{P}_1$ and $\mathcal{P}_2$, we define $N_{cdi}(\mathcal{P}_1, \mathcal{P}_2)$ as the length of the combined decision index list of $\mathcal{P}_1$ and $\mathcal{P}_2$.*

Lemma 2.3.1. *Consider either the fixed block resampling or stationary resampling from a sequence of N_{tot} distinct observations. Two paths $\mathcal{P}_1$ and $\mathcal{P}_2$ with $[I_1, I_2, \ldots, I_{cdi}]$ as the combined decision index list are identical if and only $\mathcal{P}_1[I_j] = \mathcal{P}_2[I_j]$ at any I_j, $j = 1, \ldots, N_{cdi}$.*

Proof. First, $\mathcal{P}_1$ equals to $\mathcal{P}_2$ clearly implies that $\mathcal{P}_1[I_j] = \mathcal{P}_2[I_j]$ at any I_j, $j = 1, \ldots, N_{cdi}$. Conversely, assume that $\mathcal{P}_1[I_j] = \mathcal{P}_2[I_j], j = 1, \ldots, N_{cdi}$. For any j, $j = 1, \ldots, N_{cdi}$, the entire segment $\mathcal{P}_1[I_j], \ldots, \mathcal{P}_1[I_{j+1} - 1]$ is from the same resampled subblock of the original data. Similarly, the entire segment $\mathcal{P}_2[I_j], \ldots, \mathcal{P}_2[I_{j+1} - 1]$ is from the same resampled subblock of the original data. Since $\mathcal{P}_1[I_j] = \mathcal{P}_2[I_j]$, then $\mathcal{P}_1[I_j], \ldots, \mathcal{P}_1[I_{j+1}-1]$ and $\mathcal{P}_2[I_j], \ldots, \mathcal{P}_2[I_{j+1}-1]$ are identical. Thus, the entire paths $\mathcal{P}_1$ and $\mathcal{P}_2$ are identical.

□

Theorem 2.3.1. *Consider generating a sequence of N data points using fixed block resampling from a sequence of N_{tot} distinct observations. Let path $\mathcal{P}_1$ be a bootstrap resampled path with a fixed blocksize of b_1 and path $\mathcal{P}_2$ be a bootstrap resampled path with a fixed blocksize of b_2. Then the probability of $\mathcal{P}_1$ and $\mathcal{P}_2$ being identical is $(\frac{1}{N_{tot}})^{lcm(\frac{N}{b_1}, \frac{N}{b_2})}$, where $lcm(a, b)$ is the least common multiple of integer a, b.*

Proof. Let I denote the combined decision index list of $\mathcal{P}_1$ and $\mathcal{P}_2$, with N_{cdi} denoting the total number of combined decision points and I_j denoting the jth index within I.

From Lemma 2.3.1, two paths are identical if and only if $\mathcal{P}_1[I_j] = \mathcal{P}_2[I_j]$ at any I_j, $j = 1, \ldots, N_{cdi}$.

For any $j = 1, \ldots, N_{cdi}$, since each starting point of either $\mathcal{P}_1$ or $\mathcal{P}_2$ is chosen independently with equal probability $\mathbb{P}(\mathcal{P}_1[I_j] = \mathcal{P}_2[I_j]) = \frac{1}{N_{tot}}$. Then,

$$\begin{aligned}\mathbb{P}(\mathcal{P}_1[I_j] = \mathcal{P}_2[I_j], j = 1, \ldots, N_{cdi}(\mathcal{P}_1, \mathcal{P}_2)) &= \prod_{j=1}^{N_{cdi}(\mathcal{P}_1, \mathcal{P}_2)} \mathbb{P}(\mathcal{P}_1[I_j] = \mathcal{P}_2[I_j]) \\ &= \Big(\frac{1}{N_{tot}}\Big)^{N_{cdi}(\mathcal{P}_1, \mathcal{P}_2)}.\end{aligned}$$

Since $N_{cdi}(\mathcal{P}_1, \mathcal{P}_2) = lcm(\frac{N}{b_1}, \frac{N}{b_2})$, the probability of $\mathcal{P}_1$ and $\mathcal{P}_2$ being identical is $(\frac{1}{N_{tot}})^{lcm(\frac{N}{b_1}, \frac{N}{b_2})}$.

□

Example 2.3.1. To put Theorem 2.3.1 into perspective, assume a fixed blocksize for the training paths of 6 months, and a fixed blocksize for the testing path of 24 months (or 2 years). Consider a 30-year investment horizon of monthly return paths randomly generated from historical monthly data over 90 years, i.e. $N = 30 \times 12 = 360$ and $N_{tot} = 90 \times 12 = 1080$. Then the probability of a training path being identical to a testing path is $(\frac{1}{1080})^{lcm(\frac{360}{6}, \frac{360}{24})} = (\frac{1}{1080})^{60} < 10^{-180}$. Assume that we use a total of 100,000 training paths in the training data and 10,000 testing paths in the testing data. By the union bound, the probability of the existence of a pair of identical training and testing paths is bounded by $100,000 \times 10,000 \times 10^{-180} = 10^{-171}$.

Next, we consider the stationary block bootstrap resampling, in which the blocksizes are randomly generated from a shifted geometric distribution. We first establish a few properties and lemmas to set up the proof for the main theorem.

Properties 2.3.1 (Properties of a Geometric Distribution)**.** *Suppose the integer $m > 0$ is drawn from a shifted geometric distribution, with $\mathbb{E}[m] = 1/p$, then*

$$\begin{aligned} \mathbb{P}[m = k] &= (1-p)^{k-1}p \\ \mathbb{P}[m \geq k] &= (1-p)^{k-1} \ . \end{aligned} \tag{2.18}$$

We rewrite equation (2.18) in a form amenable to manipulation. Let

$$(1-p) = e^{-\lambda} \ , \tag{2.19}$$

so that equation (2.18) becomes

$$\begin{aligned} \mathbb{P}[m = k] &= e^{-\lambda k}(e^{\lambda} - 1) \\ \mathbb{P}[m \geq k] &= e^{-\lambda(k-1)} \\ \lambda &= -\log[1-p] \ . \end{aligned} \tag{2.20}$$

Denote the expected blocksize by $\hat{b}$, then in our case, $p = 1/\hat{b}$, and consequently

$$\lambda = -\log\left[1 - \frac{1}{\hat{b}}\right] \ . \tag{2.21}$$

Lemma 2.3.2. *Suppose $[I_1, \ldots, I_k]$ is the decision index list of a block resampled path of length N with the expected blocksize of $\hat{b}$. Then the probability of the decision index list $[I_1, \ldots, I_k]$ occurring is $e^{-\lambda(N-1)}(e^{\lambda} - 1)^{k-1}$, with $\lambda = -\log[1 - \frac{1}{\hat{b}}]$.*

Proof. By definition, $I_{j+1} > I_j$ for any $j = 1, \ldots, k-1$, and $I_{k+1} = N + 1$. The probability of path $\mathcal{P}$ having $[I_1, \ldots, I_k]$ as the decision index list is equal to the probability of path $\mathcal{P}$ having the first block with blocksize of $I_2 - I_1$, ..., the kth block with blocksize of $I_{k+1} - I_k$. Denote the blocks of path $\mathcal{P}$ as $B_1, \ldots, B_k$. According to Properties 2.3.1,

$$\mathbb{P}(\mathit{blocksize}(B_j) = I_{j+1} - I_j) = \begin{cases} e^{-\lambda(I_{j+1} - I_j)}(e^{\lambda} - 1), & \text{if } j < k \\ e^{-\lambda(I_{k+1} - I_k - 1)}, & \text{if } j = k \end{cases}$$

The probability of path $\mathcal{P}$ having $[I_1, \ldots, I_k]$ as the decision index list is

$$\prod_{j=1}^{k} \mathbb{P}(\mathit{blocksize}(B_j) = I_{j+1} - I_j) = e^{-\lambda(I_{k+1} - I_1 - 1)}(e^{\lambda} - 1)^{k-1} = e^{-\lambda(N-1)}(e^{\lambda} - 1)^{k-1}.$$

□

Lemma 2.3.2 shows that the probability of a stationary block resampled path $\mathcal{P}$ with an expected blocksize of $\hat{b}$ having a decision index list is uniquely determined by the expected blocksize $\hat{b}$, the path length N, and the length of the decision index list k.

Lemma 2.3.3. *Suppose two paths $\mathcal{P}_1$ and $\mathcal{P}_2$ of the length N are generated by stationary block bootstrap resampling with the expected blocksizes of $\hat{b}_1$ and $\hat{b}_2$ respectively. Then*

$$\begin{aligned}\mathbb{P}(N_{cdi}(\mathcal{P}_1,\mathcal{P}_2)=k) &= \binom{N-1}{k-1}e^{-(\lambda_1+\lambda_2)(N-1)}(e^{\lambda_1+\lambda_2}-1)^{k-1} \\ &\lambda_1=-\log\left[1-\frac{1}{\hat{b}_1}\right]\ ;\ \lambda_2=-\log\left[1-\frac{1}{\hat{b}_2}\right]\ . \qquad (2.22)\end{aligned}$$

Proof. Let $f(\hat{b},n)$ denote the occurrence probability of a stationary block resampled path of length N with the expected blocksize of $\hat{b}$ and a decision index list of length n (this is given by Lemma 2.3.2).

Suppose $[I_1,\ldots,I_k]$ is a combined index list of any two paths $\mathcal{P}_1$ and $\mathcal{P}_2$. Let v be the number of overlapped indices and i be the number of non-overlapped indices for $\mathcal{P}_1$ respectively, corresponding to $[I_1,\ldots,I_k]$.

Enumerating the possible values for v, the number of overlapped indices and values for i, the number of non-overlapped indices in $\mathcal{P}_1$, the probability of a combined decision index list $[I_1,\ldots,I_k]$ occurring equals

$$\sum_{v=1}^{k}\Big(\binom{k-1}{v-1}\sum_{i=0}^{k-v}\binom{k-v}{i}f(\hat{b}_1,v+i)f(\hat{b}_2,k-i)\Big). \qquad (2.23)$$

Note that

$$\begin{aligned}&\sum_{v=1}^{k}\Big(\binom{k-1}{v-1}\sum_{i=0}^{k-v}\binom{k-v}{i}f(\hat{b}_1,v+i)f(\hat{b}_2,k-i)\Big)\\ =&\sum_{v=1}^{k}\Big(\binom{k-1}{v-1}\sum_{i=0}^{k-v}\binom{k-v}{i}e^{-\lambda_1(N-1)}(e^{\lambda_1}-1)^{v+i-1}e^{-\lambda_2(N-1)}(e^{\lambda_2}-1)^{k-i-1}\Big)\\ =&e^{-(\lambda_1+\lambda_2)(N-1)}\sum_{v=1}^{k}\Big(\binom{k-1}{v-1}\big(e^{\lambda_1+\lambda_2}-e^{\lambda_1}-e^{\lambda_2}+1\big)^{v-1}\Big(\sum_{i=0}^{k-v}\binom{k-v}{i}(e^{\lambda_1}-1)^i(e^{\lambda_2}-1)^{k-v-i}\Big)\Big)\\ =&e^{-(\lambda_1+\lambda_2)(N-1)}\sum_{v=1}^{k}\Big(\binom{k-1}{v-1}\big(e^{\lambda_1+\lambda_2}-e^{\lambda_1}-e^{\lambda_2}+1\big)^{v-1}\Big(e^{\lambda_1}+e^{\lambda_2}-2\Big)^{k-v}\Big)\\ =&e^{-(\lambda_1+\lambda_2)(N-1)}(e^{\lambda_1+\lambda_2}-1)^{k-1}.\end{aligned}$$

Since there are $\binom{N-1}{k-1}$ combinations of the decision index list of length k, we conclude

$$\mathbb{P}(N_{cdi}(\mathcal{P}_1,\mathcal{P}_2)=k)=\binom{N-1}{k-1}e^{-(\lambda_1+\lambda_2)(N-1)}(e^{\lambda_1+\lambda_2}-1)^{k-1}.$$

□

Using Lemma 2.3.1 and Lemma 2.3.3, we establish the probability of two paths generated with stationary block bootstrap resampling being identical.

Theorem 2.3.2. *Consider generating a sequence of N data points using stationary block resampling from a sequence of N_{tot} distinct observations. Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two paths generated from the stationary block bootstrap resampling from this observation sequence with the expected blocksizes of $\hat{b}_1$ and $\hat{b}_2$ respectively, and both have a length of N. The probability of $\mathcal{P}_1$ and $\mathcal{P}_2$ being identical is*

$$\frac{1}{N_{tot}}\Big((1-\frac{1}{\hat{b}_1})(1-\frac{1}{\hat{b}_2})+\frac{\frac{1}{\hat{b}_1}+\frac{1}{\hat{b}_1}-\frac{1}{\hat{b}_1\hat{b}_2}}{N_{tot}}\Big)^{N-1}.$$

Proof. Using Lemma 2.3.1, $\mathcal{P}_1=\mathcal{P}_2$ if and only if the observations from $\mathcal{P}_1$ and $\mathcal{P}_2$ are equal at each of the indexes in the combined decision index list. Thus,

$$\mathbb{P}\big(\mathcal{P}_1=\mathcal{P}_2|N_{cdi}(\mathcal{P}_1,\mathcal{P}_2)=k\big)=\left(\frac{1}{N_{tot}}\right)^k.$$

Additionally, following Lemma 2.3.3, we have

$$\begin{aligned}
\mathbb{P}(\mathcal{P}_1=\mathcal{P}_2)=&\sum_{k=1}^{N}\mathbb{P}\big(N_{cdi}(\mathcal{P}_1,\mathcal{P}_2)=k\big)\cdot\mathbb{P}\big(\mathcal{P}_1=\mathcal{P}_2|N_{cdi}(\mathcal{P}_1,\mathcal{P}_2)=k\big)\\
=&\sum_{k=1}^{N}\binom{N-1}{k-1}e^{-(\lambda_1+\lambda_2)(N-1)}(e^{\lambda_1+\lambda_2}-1)^{k-1}(\frac{1}{N_{tot}})^k\\
=&\frac{e^{-(\lambda_1+\lambda_2)(N-1)}}{N_{tot}}\sum_{k=1}^{N}\binom{N-1}{k-1}\Big(\frac{e^{\lambda_1+\lambda_2}-1}{N_{tot}}\Big)^{k-1}\\
=&\frac{e^{-(\lambda_1+\lambda_2)(N-1)}}{N_{tot}}\Big(1+\frac{e^{\lambda_1+\lambda_2}-1}{N_{tot}}\Big)^{N-1}\\
=&\frac{1}{N_{tot}}\Big(e^{-(\lambda_1+\lambda_2)}+\frac{1-e^{-(\lambda_1+\lambda_2)}}{N_{tot}}\Big)^{N-1}\\
=&\frac{1}{N_{tot}}\Big((1-\frac{1}{\hat{b}_1})(1-\frac{1}{\hat{b}_2})+\frac{\frac{1}{\hat{b}_1}+\frac{1}{\hat{b}_1}-\frac{1}{\hat{b}_1\hat{b}_2}}{N_{tot}}\Big)^{N-1}.
\end{aligned}$$

□

Example 2.3.2. Consider the following example. If the training paths are bootstrap resampled with an expected blocksize of 6 months (0.5 years) and the testing paths with an expected blocksize of 24 months (2 years), and $N=30\times 12=360$ (30-year horizon) and

$N_{tot} = 90 \times 12 = 1080$ (90 years of historical monthly return data). If the training data set consists of a total of 100,000 training paths and the testing data set consists of 10,000 testing paths, by union bound, the probability of observing a pair of training and testing paths being identical is bounded by $100,000 \times 10,000 \times 8.737 \times 10^{-39} < 10^{-29}$.

Therefore, even when the training set and testing set are generated from the same data sequence, the probability of observing the same path in the training and testing data set is near zero. This suggests that using the block bootstrap resampling to generate training and testing data sets is a robust method for enhancing data for the learning framework.

Remark 2.3.2. Under stationary block bootstrap, a path is likely to have large actual blocksizes even if the expected blocksize is relatively small, which can result in a higher probability of observing two identical paths than under fixed block bootstrap. For example, a path with an expected blocksize of 10 years has a 5% probability of only containing one block of 30 years, which increases the probability of one path being identical to another path, according to Theorem 2.3.2.

2.4 Numerical experiments

In this section, we evaluate the performance of the proposed data-driven approach for outperforming a stochastic target in the context of the accumulation phase of a 30-year DC pension plan. In the numerical experiments, we focus on portfolios with only two assets: a stock index and a bond index, as described in Example 2.2.1. The benchmark portfolio is a constant weight strategy, which is rebalanced to 50% bonds and 50% stocks annually. We denote the wealth of the benchmark strategy at time t by $W_{50/50}(t)$.

Before we discuss the numerical results, we first recall the concept of *partial stochastic dominance*, which is useful for the evaluation of the relative performance of investment strategies.

Suppose two investment strategies A and B are evaluated on a set of data samples under the same investment scenario. We consider the cumulative distribution functions (CDF) of terminal wealth W associated with both strategies. Specifically, we denote the CDF of strategy A by $\text{CDF}_A(W)$ and that of strategy B by $\text{CDF}_B(W)$. Let W_T be the random wealth at time T and W be a possible wealth realization, then we can interpret $\text{CDF}_A(W)$ as

$$\text{CDF}_A(W) \quad = \quad Prob(W_T \leq W) \ . \tag{2.24}$$

Following Atkinson (1987); van Staden et al. (2021), we define partial first-order stochastic dominance.

Definition 2.4.1 (Partial first-order stochastic dominance)**.** *Given an investment strategy A which generates a CDF of terminal wealth W given by $CDF_A(W)$, and a strategy B with*

$CDF_B(W)$, then strategy A partially stochastically dominates strategy B (to first order) in the interval (W_{lo}, W_{hi}) if

$$CDF_A(W) \quad \leq \quad CDF_B(W), \ \forall W \in (W_{lo}, W_{hi}) \tag{2.25}$$

with strict inequality for at least one point in (W_{lo}, W_{hi}).

The arguments for relaxing the usual definition of stochastic dominance are given in (Atkinson, 1987; van Staden et al., 2021). Given some initial wealth W_0, if $W_{hi} \gg W_0$, then an investor may not be concerned that strategy A underperforms strategy B at these very high wealth values. In this case, the investor is already fabulously wealthy. Suppose that $W_{lo} \ll W_0$, and assume $\text{CDF}_A(W_{lo}) = \text{CDF}_B(W_{lo})$. As an extreme example, suppose $W_{lo} =$ two cents. The fact that strategy B has a higher probability of ending up with one cent compared with strategy A is cold comfort, and not particularly interesting. On the other hand, suppose $\text{CDF}(W_{lo}) \ll 1$. Again, an investor may not be interested in these events with exceptionally low probabilities.

Remark 2.4.1. (Intuition behind partial stochastic dominance). Strategy A stochastically dominates Strategy B if Definition 2.4.1 holds with (W_{lo}, W_{hi}) = (-∞, +∞). In this case, any investor who prefers more wealth rather than less wealth will prefer strategy A.
Partial stochastic dominance generalizes this so that Strategy A will be preferred over Strategy B, except for cases which the investor does not think are relevant. This is a practically useful definition since pure first-order stochastic dominance is rare.

2.4.1 Original data and augmentation

Our main objective here is to consider the core allocation problem between a risky asset (i.e. stocks) and a defensive asset (i.e. bonds).

To that end, we use monthly historical data from the Center for Research in Security Prices (CRSP) from January 1, 1926, to December 31, 2015.[2]. Specifically, we use the CRSP 3-month Treasury bill (T-bill) index and the CRSP cap-weighted total return index. The latter index includes all distributions for all domestic stocks trading on major U.S. exchanges. Since both indexes are in nominal terms, we adjust them for inflation using the U.S. CPI index, also supplied by CRSP. We use real indexes since investors saving for retirement should be focused on real (not nominal) wealth goals. Note that Li and Forsyth (2019), in the context of a fixed (non-stochastic) target-based objective function, have also tested the use of the CRSP capitalization-weighted index (as the risky asset) and the ten-year treasury bond index (as the defensive asset). The control strategies are qualitatively similar

[2]More specifically, results presented here were calculated based on data from Historical Indexes, ©2015 Center for Research in Security Prices (CRSP), The University of Chicago Booth School of Business. Wharton Research Data Services was used in preparing this chapter. This service and the data available thereon constitute valuable intellectual property and trade secrets of WRDS and/or its third-party suppliers.

for either choice of the risky and the defensive asset. We have also carried out similar tests for the stochastic benchmark objective function. The results, using a ten-year treasury as the defensive asset, can be found in Section 3.2. For simplicity here, we will focus on the CRSP index and the 3-month T-bill case.

For the stock index and bond index, Table 2.1 shows the optimal expected blocksize for each index estimated from the historical data. When using the resampling method in the proposed data-driven NN approach, we simultaneously sample the same time block across all asset data sets (i.e. the stock index and bond index). Since the optimal blocksize varies with the index, it is not clear which blocksize to use since we need to simultaneously resample both indices. Consequently, we will carry out tests with a variety of blocksizes, in the ranges reported in Table 2.1.

Data Series	Optimal expected block size $\hat{b}$ (months)
Real 3-month T-bill index	50.1
Real CRSP cap-weighted index	1.8

Table 2.1: Optimal expected blocksize $\hat{b} = 1/v$ when the blocksize follows a geometric distribution $Pr(b = k) = (1 - v)^{k-1}v$. The algorithm in Patton et al. (2009) is used to determine $\hat{b}$.

2.4.2 Experiment setting

As discussed in Section 2.1.2, we use an example of an investor in a DC plan to illustrate the application of the data-driven methodology. In the numerical experiment, we assume an investor starts with zero wealth (balance) in the DC plan and makes a real cash injection of 10 per year[3] for 30 years. At the beginning of every year, the investor has the choice to rebalance the DC portfolio and change the allocation weights to a stock index fund and a bond index fund. The market data is generated following the methodology in Section 2.4.1.

Here we list the parameters used in training and testing the proposed data-driven approach:

- L: a total of $L = 100,000$ bootstrap paths are used for training;
- L_{test}: a total of $L_{test} = 10,000$ paths are bootstrap resampled from a different expected blocksize than the training data for testing the strategy performance;
- $W(0)$: initial wealth is $W(0) = 0$;
- T: the entire investment period is $T = 30$ years;

[3]We will use thousands of dollars as the unit of wealth

- N: the entire period is divided into $N = 30$ periods. At the beginning of each period rebalancing occurs, i.e., annual rebalancing;
- q: annual cash injection is $q = 10$;
- s: the annual target outperformance rate $s = 1\%$ for calculating the elevated target $e^{sT} \cdot W_{50/50}(T)$, where $W_{50/50}(T)$ is the terminal wealth of the constant proportion portfolio;
- 3 features:
  - $T - t$: time remaining in the investment period,
  - $W(t)$: wealth of the adaptive portfolio at time t,
  - $W_{50/50}(t)$: wealth of the constant proportion portfolio at time t.

We remark that in this numerical example, we are assuming annual contributions of a fixed dollar amount, in real terms (inflation-adjusted). We are aware that many pensioners care about the replacement rate (percentage of annual employment income replaced by retirement income) which measures how well retirees can maintain their lifestyles in retirement. In fact, depending on specific assumptions about the salary, we can scale up the cash contribution number to estimate whether a good replacement rate can be achieved. We present a realistic example at the end of Section 2.4.6 which shows that an investor can expect to achieve a reasonable level of replacement income in 30 years following the DC plan strategy.

It is also worth noting that we do not consider transaction costs in the experiment. The reasoning for this assumption is two-fold. Firstly, we consider index funds as investment assets, which usually enjoy low transaction fees. Secondly, the portfolio does not have frequent rebalancing. Particularly, in the DC plan example, we consider yearly rebalancing, and the transaction cost associated with the rebalancing is unlikely to cause any meaningful impact on the strategy.

2.4.3 Strategy performance and characteristics

We now evaluate the performance of the optimal adaptive strategy trained on bootstrap resampled data. First, we show the performance of the optimal adaptive strategy trained on the bootstrap resampled data with the expected blocksize $\hat{b} = 0.5$ years, and tested on bootstrap resampled data with expected blocksize of $\hat{b} = 2$ years, which is the average optimal blocksize.[4]

[4]In fact, the strategy performance using alternative training-testing expected blocksize pairs is qualitatively similar (see Section 3.4).

Training Results on Bootstrap Data: Expected Blocksize $\hat{b} = 0.5$ years					
Strategy	$E(W_T)$	$std(W_T)$	$median(W_T)$	$Pr(W_T < median(W_T^{CP}))$	$Pr(W_T < median(W_T^{NN}))$
constant proportion($p = 0.5$)	678	276	624	0.50	0.84
adaptive	963	474	913	0.27	0.50
Testing Results on Bootstrap Data: Expected Blocksize $\hat{b} = 2$ years					
Strategy	$E(W_T)$	$std(W_T)$	$median(W_T)$	$Pr(W_T < median(W_T^{CP}))$	$Pr(W_T < median(W_T^{NN}))$
constant proportion($p = 0.5$)	679	267	629	0.50	0.84
adaptive	962	449	921	0.26	0.50

Table 2.2: Terminal wealth statistics of the optimal adaptive strategy, trained on bootstrap resampled data with blocksize $\hat{b} = 0.5$ years and tested on bootstrap resampled data with blocksize $\hat{b} = 2$ years.

Table 2.2 reports performance statistics and the probability of the terminal wealth being less than the median of the terminal wealth of both strategies. From Table 2.2 , we observe that

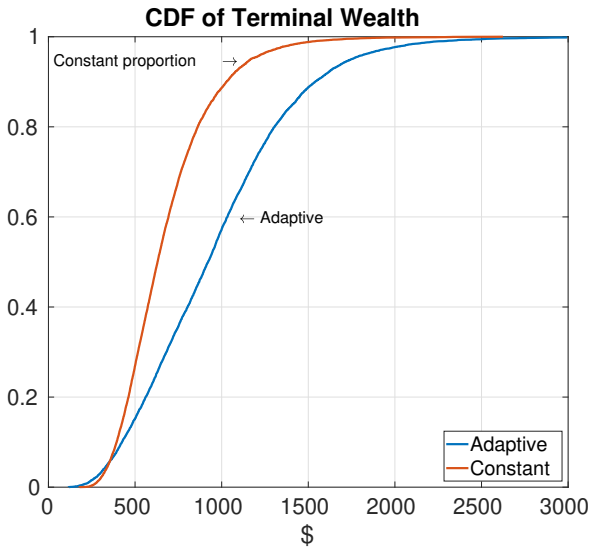
- The median and mean terminal wealth of the optimal adaptive strategy are significantly higher than the constant proportion strategy.
- The optimal adaptive strategy has only 26% probability of achieving a lower terminal wealth than the median terminal wealth of the constant proportion strategy ($median(W_T^{CP})$), while the constant proportion strategy has an 84% probability of achieving a lower terminal wealth than the median terminal wealth of the NN adaptive strategy ($median(W_T^{NN})$).

It is also worth noting that the standard deviation of the terminal wealth of the optimal adaptive strategy is higher than the standard deviation of the terminal wealth of the constant proportion strategy. In the context of dynamic trading, a higher standard deviation does not imply that the performance of the strategy is poor. In fact, we can observe from Figure 2.3a that the distribution of the terminal wealth of the optimal adaptive strategy is significantly more right-skewed. A higher standard deviation of terminal wealth is desirable in the right-skewed situation (van Staden et al., 2021). This illustrates why standard deviation and Sharpe Ratio are poor measures of risk for inherently non-linear strategies (Lhabitant, 2000). In fact, the optimal adaptive dynamic strategy has properties in common with option-based strategies.

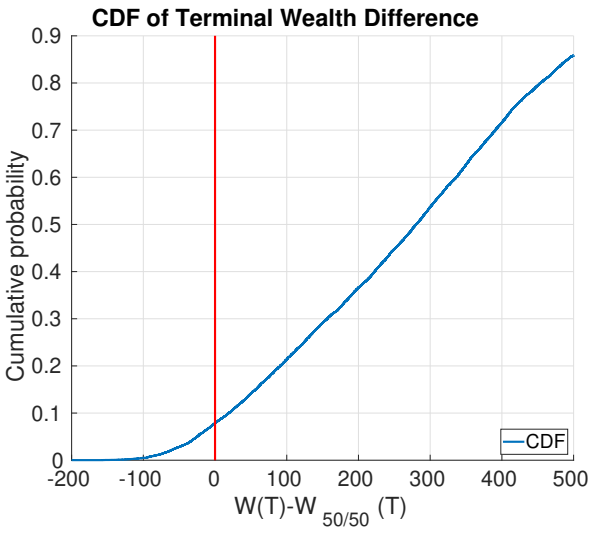
In addition, Figure 2.3a shows that the adaptive strategy achieves partial stochastic dominance over the constant proportion strategy, which indicates that the adaptive strategy has a more favorable terminal wealth distribution.

We note that the terminal wealth distribution of the optimal adaptive strategy has a slightly worse left tail than the constant proportion strategy. The 5% VaR of terminal wealth (i.e. the 5th percentile of terminal wealth distribution) is 326 for the optimal adaptive strategy and 338 for the constant proportion strategy.[5] In fact, from Table 2.3 we can observe

[5]We measure quantiles of the terminal wealth, not losses. Hence a larger value of VaR is more desirable, i.e. has less risk.



(a) CDF of terminal wealth for adaptive strategy and constant proportion strategy



(b) CDF of terminal wealth difference between adaptive and constant proportions strategy

Figure 2.3: Histogram of terminal wealth $W(T)$ (adaptive) and $W_{50/50}(T)$ (constant proportion) and CDF of wealth difference $W(T) - W_{50/50}(T)$ based on the testing data (bootstrap data with $\hat{b}$=2 years)

that the adaptive strategy has worse 5% and 1% VaR and CVaR (Conditional Value at Risk, i.e. the average terminal wealth value in the tail percentiles) than the constant proportion strategy.

VaR and CVaR on Testing Data				
Strategy	5% VaR	5% CVaR	1% VaR	1% CVaR
constant proportion($p = 0.5$)	338	294	265	238
adaptive	326	253	201	169

Table 2.3: VaR and CVaR of the terminal wealth of the adaptive strategy and constant proportion strategy.

These tail events occur when the bootstrapped paths correspond to consistently bearish market periods when stocks underperform bonds for a long period of time. Recall that the objective function in (2.6) is to determine a strategy with the terminal wealth achieving a certain premium over the benchmark strategy, rather than to optimize the tail risk metrics of the adaptive strategy such as VaR and CVaR.

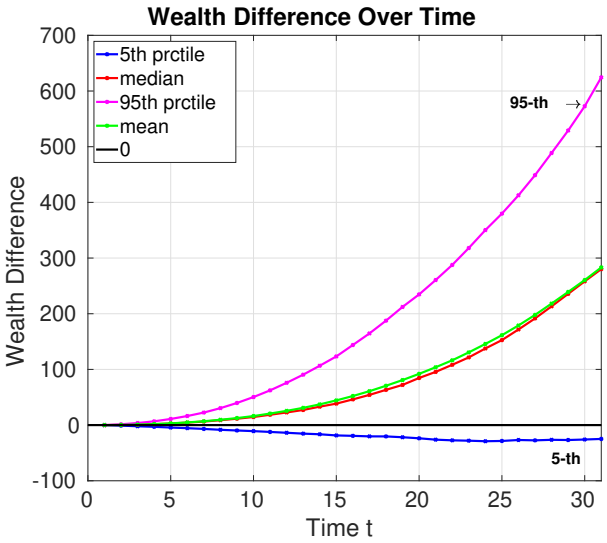
In other words, the objective function is designed to optimize the pathwise terminal wealth difference between the adaptive strategy and the constant proportion strategy, hence the idea of "beating the stochastic benchmark target". Figure 2.3b shows the cumulative

distribution function (CDF) of the wealth difference $W(T) - W_{50/50}(T)$ that provides a more direct comparison between the optimal adaptive strategy and the constant proportion strategy along the same paths. From Figure 2.3b, we can observe that the probability of the optimal adaptive strategy underperforming the constant proportion strategy is less than 10%. When underperformance occurs, the magnitude of underperformance is small compared to the magnitude of outperformance.

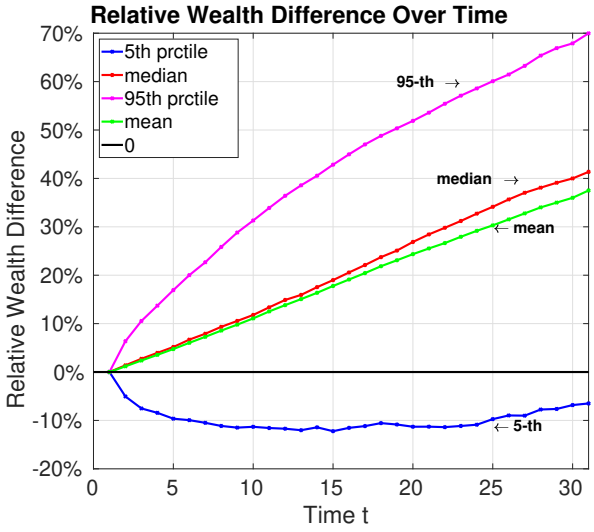
If reducing the tail risk has a higher priority in the investment plan, a tail-risk measure such as CVaR can be included in the objective function accordingly. This, of course, will produce a lower probability of pathwise outperformance over the benchmark strategy. However, the proposed framework can be similarly adopted by including suitable optimization methods for CVaR optimization, see, e.g., Alexander et al. (2006); van Staden et al. (2021); Forsyth (2022).

So far, we have analyzed and compared the overall performance based on terminal wealth. Next, we provide more detailed comparisons of the various characteristics of the strategies.

Since the objective function for the optimal control (2.6) is defined in terms of the terminal wealth, we examine how the optimal adaptive strategy performs over the entire investment period.



(a) Percentiles of wealth difference $W(t) - W_{50/50}(t)$ over time



(b) Percentiles of relative wealth difference $\frac{W(t)-W_{50/50}(t)}{W_{50/50}(t)}$ over time

Figure 2.4: Wealth difference and relative wealth difference over time: $W(t)$ denotes the optimal adaptive is wealth and $W_{50/50}(t)$ denotes the benchmark

Figure 2.4 graphs the average and various percentiles of the wealth difference $W(t) - W_{50/50}(t)$ and the relative wealth difference $\frac{W(t)-W_{50/50}(t)}{W_{50/50}(t)}$ over the investment horizon. From Figure 2.4, we observe that

- With a high probability, the optimal adaptive strategy achieves higher wealth than the constant proportion strategy over time.
- The outperformance of the optimal adaptive strategy in terms of the relative wealth difference is not as significant as the wealth difference in dollar values.

The observations indicate that a larger outperformance of the optimal adaptive strategy often occurs when the constant proportion strategy performs well. Nevertheless, the outperformance of the optimal adaptive strategy in terms of the relative wealth difference is still very impressive with a median value of almost 40% at the terminal stage. Of course, if we are primarily interested in relative outperformance, it is a simple matter to alter the objective function to focus on achieving this goal.

Figure 2.4 shows that, even though the objective function only targets the wealth difference of the portfolios at the terminal time, without having any direct restrictions on the wealth of the optimal adaptive strategy in the interim period, the adaptive strategy still manages to have a statistically higher wealth throughout the entire investment period.

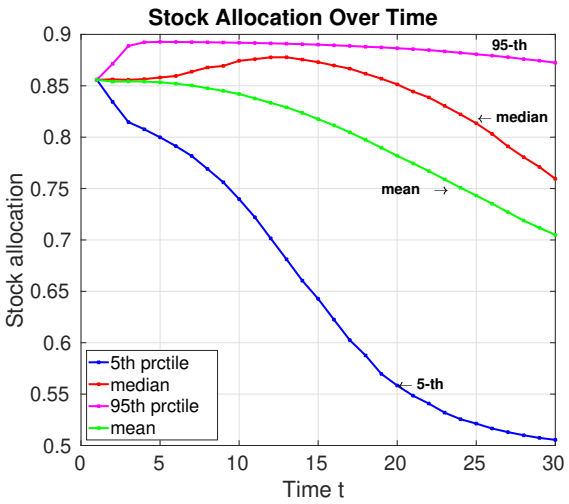
Remark 2.4.2. (Probability of underperformance). Note that any attempt to beat a benchmark has some positive probability of underperformance. To assert otherwise is to postulate an arbitrage opportunity, i.e. go long the active portfolio and short the benchmark. If such a possibility existed, the author of this thesis would be extremely rich. The readers may draw their own conclusions.

We further examine the characteristics of the optimal adaptive strategy. Figure 2.5a shows different percentiles of the stock allocation of the optimal adaptive strategy over time. We observe that

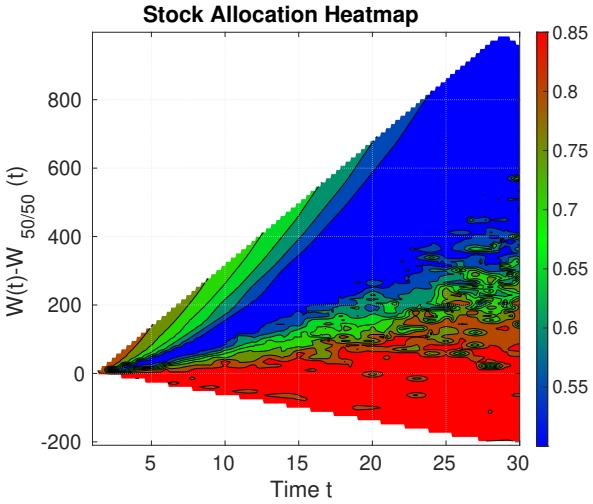
- In general, the stock allocation (fraction of wealth invested in stocks) decreases when approaching the end of the investment horizon.
- The stock allocation almost always stays above the benchmark allocation of 50%.

With a red-blue color scheme, Figure 2.5b shows the heatmap of the stock allocation with respect to time t and the wealth difference $W(t) - W_{50/50}(t)$. Darker shades of the red color indicate more allocation in stocks and darker shades of the blue color indicate more allocation in bonds.

From Figure 2.5b, we observe that when $W(t) - W_{50/50}(t)$ is positive and large (optimal adaptive strategy outperforming), the allocation of wealth to the stock becomes small. The intuitive explanation is that the optimal adaptive strategy tends to decrease the wealth allocation to stocks once it has established an advantage over the benchmark constant proportion strategy. This also explains why the stock allocation almost always stays above 50%. In most cases where the optimal adaptive strategy has established an advantage over the constant proportion strategy (as we have observed in Figure 2.4), decreasing the stock



(a) Percentiles of the fraction invested in stocks over time for the adaptive strategy



(b) Heatmap, fraction invested in stocks for the adaptive strategy

Figure 2.5: Fraction invested in stocks over time for the optimal adaptive strategy: percentiles and the heatmap

allocation to 50% to maintain the same allocation strategy as the 50/50 constant proportion strategy locks in the outperformance.

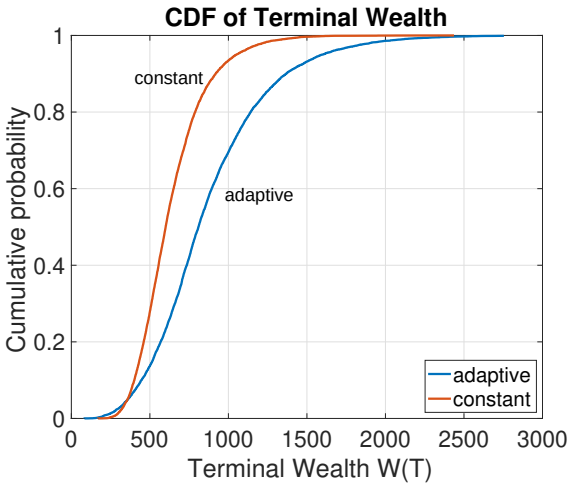
On the other hand, when $W(t) - W_{50/50}(t)$ is negative (i.e. the adaptive strategy underperforms), the optimal policy allocates more wealth to stocks. This is because the stock index has a higher expected return than the bond index. To eventually outperform the constant proportion strategy, the adaptive strategy invests more wealth in stocks, in an attempt to make up for the lost ground.

In fact, the optimal adaptive strategy appears to be a contrarian strategy, following which an investor buys and sells in opposition to the prevailing sentiment at the time.

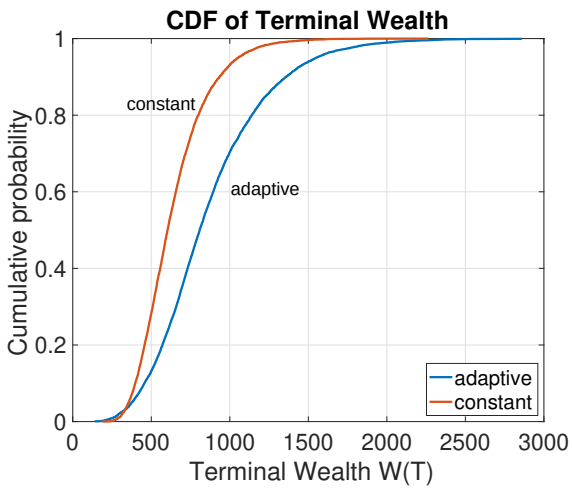
2.4.4 The three-asset case

While we only discuss the investment scenario with two assets in Section 2.4.2, the proposed framework can be easily extended to more assets. Here, we present results from an example with three assets - the capitalization-weighted CRSP stock index, the 3-month T-bill index, and the 10-year T-bond index. We choose the benchmark to be a 40/30/30 split constant proportion strategy, where 40% of the wealth is allocated to the cap-weighted stock index, 30% to the 3-month T-bill index, and 30% to the 10-year T-bond index.

We train the neural network model on bootstrap resampled data with an expected blocksize of 0.5 years, with the proposed asymmetric objective function (2.6). We then test the learned adaptive strategy on bootstrap resampled data with an expected blocksize of 2 years.

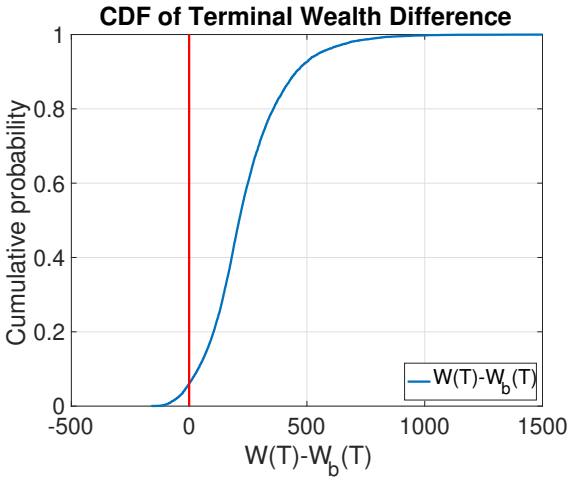


(a) Training on bootstrap data with $\hat{b}$=0.5 years

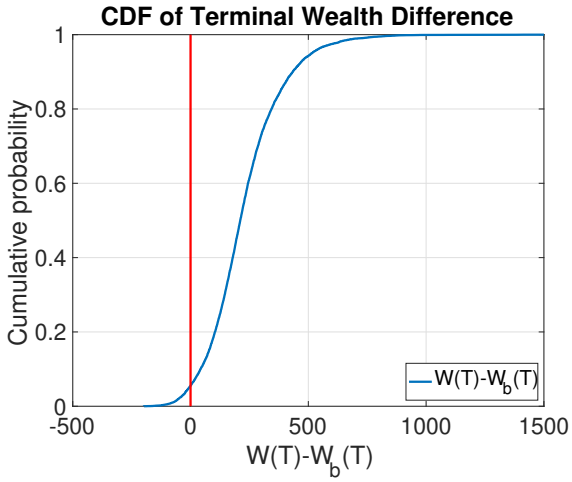


(b) Testing on bootstrap data with $\hat{b}$=2 years

Figure 2.6: CDF of terminal wealth for the 3 asset case



(a) Training on bootstrap data with $\hat{b}$=0.5 years



(b) Testing on bootstrap data with $\hat{b}$=2 years

Figure 2.7: CDF of terminal wealth difference for the 3 asset case, where $W_b(T)$ indicate the terminal wealth of the constant proportion benchmark strategy

We can observe from Figure 2.6 that the adaptive strategy produces a consistently more right-skewed distribution of the terminal wealth compared with the constant proportion benchmark strategy. The path-wise comparison of terminal wealth difference also shows consistent outperformance compared to the adaptive strategy in both training and testing.

The framework can easily include more assets. However, the choice of which assets to use, especially considering the recent interest in factor indexes, is beyond the scope of this thesis.

2.4.5 Forward-looking bias in bootstrap resampling?

In Section 2.4.2, both training and testing data sets are generated from bootstrap resampled data from a single historical return path from 1926-2015. A possible criticism of such an approach is that both the training data and testing data share the same information source. In particular, is it possible for the training data to have a forward-looking bias?

We argue that there is no forward-looking bias in the described training and testing data generation process. Recall that in the experiments, training data and testing data have different expected blocksizes and thus different distributions. Specifically, when bootstrap resampling randomly with different expected blocksizes, the ordering of blocks of data points is randomly shuffled and sequential ordering information is destroyed. Further, Theorem 2.3.1 and 2.3.2 show that the probability of an entire path in the training data set reappearing in the testing data set is vanishingly small. This is due to the random block resampling nature of the bootstrap algorithm.

Nonetheless, to provide additional evidence of robustness, we compare the following two different cases:

Case #1: (Overlap case). We train the adaptive strategy on bootstrap resampled data from the entire historical path from 1926 to 2015. We test the strategy on bootstrap resampled data from the last 30 years of the historical path from 1986-2015. There is an overlap between the underlying historical path for training and testing (1986-2015). We show that such overlap does not introduce an advantage in terms of the strategy performance by comparing it with case #2 - the *non-overlap* case.

Case #2: (Non-overlap case). We train the adaptive strategy on bootstrap resampled data from the first 60 years of the historical path from 1926 to 1985. We test the strategy on the same bootstrap resampled data generated from the last 30 years of the historical path from 1986-2015 as in case #1. Consequently, there is no overlap between the underlying historical paths we use for generating training data and testing data at all.

Figure 2.8 and Figure 2.9 show these two cases schematically. Case #2 is the more stringent test case as there are zero overlaps between the underlying historical data for the generation of the training set and the testing set.

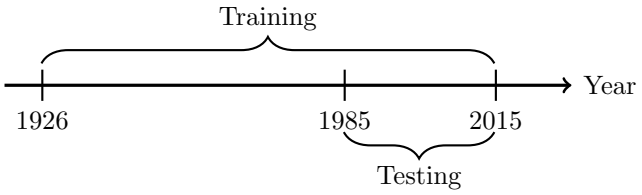


Figure 2.8: Case #1: use historical data from 1926-2015 for generating training data, and 1986-2015 for testing. There is an overlap between the underlying historical paths for training and testing.

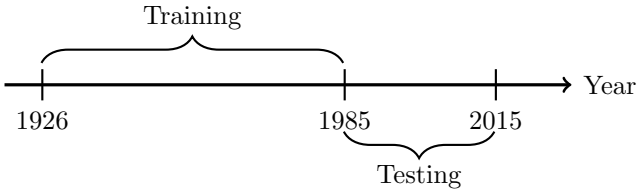


Figure 2.9: Case #2: "non-overlap" case where underlying market data for training and testing data has no overlaps. Case #2 uses the same testing data set as case #1.

Note that, for Case #1 and #2, the underlying historical data for testing data only covers a 30-year window. Recall that our investment horizon in our previous experiments was $T = 30$ years. In order to obtain more meaningful block bootstrap resampling results, we reduce the investment horizon to $T = 15$ years for both cases in this section.

We first compare the CDF of the terminal wealth difference for the two cases. From Figure 2.10, we can observe that Case #1 (the *overlap* case) and Case #2 (the *non-overlap* case) produce almost identical CDF curves, which validates the argument that forward-looking bias is not a concern in our approach. Despite using the entire historical period as the underlying data for training, case #1 does not yield a superior CDF than Case #2, in which the underlying market data for training data and testing data have no overlaps.

We further examine the percentiles of the terminal wealth difference. From Figure 2.11, we can observe that both cases have almost identical wealth differences in different percentiles, except that Case #2 has slightly better tail risk control (fifth percentile) than Case #1. This actually further proves that the overlap does not introduce performance advantage as the *non-overlap* case actually has less tail risk.

In Figure 2.12, we compare the actual strategies, i.e., stock allocations of both cases. This time we can observe some differences between Case #1 and Case #2. From the median and mean plot, we can observe that Case #2 tends to de-risk (decrease allocation in the stocks) more aggressively over time than Case #1. However, the difference between allocation strategies is not significant. In fact, the average stock holding over time is quite similar for both cases. In addition, we have already observed similar strategy performances in terms of terminal wealth distributions from Figure 2.10 and Figure 2.11.

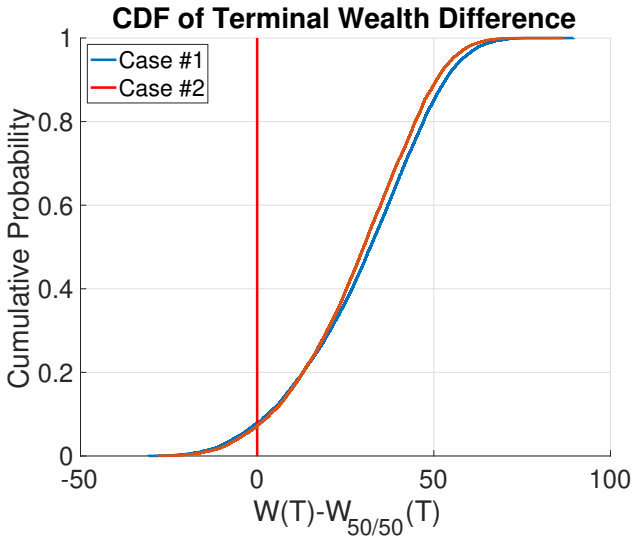


Figure 2.10: CDF of wealth difference $W(T) - W_{50/50}(T)$, testing result

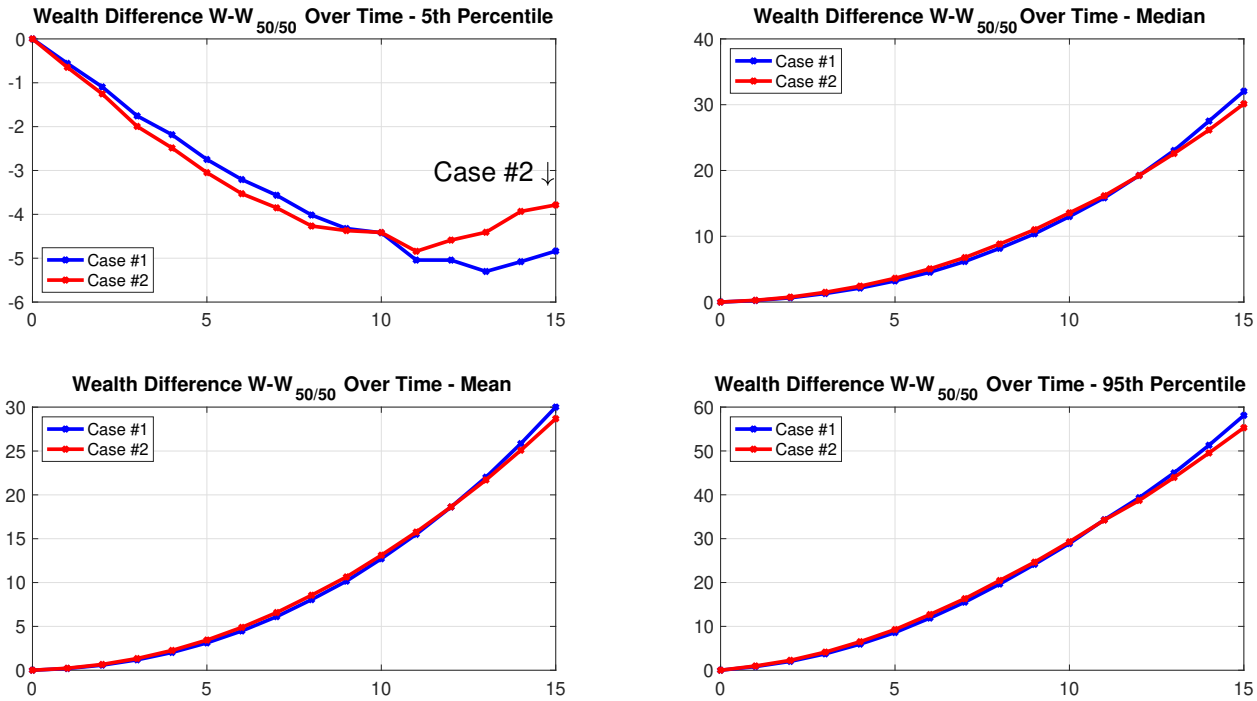


Figure 2.11: Percentiles of wealth difference $W(T) - W_{50/50}(T)$ for the two cases, testing results

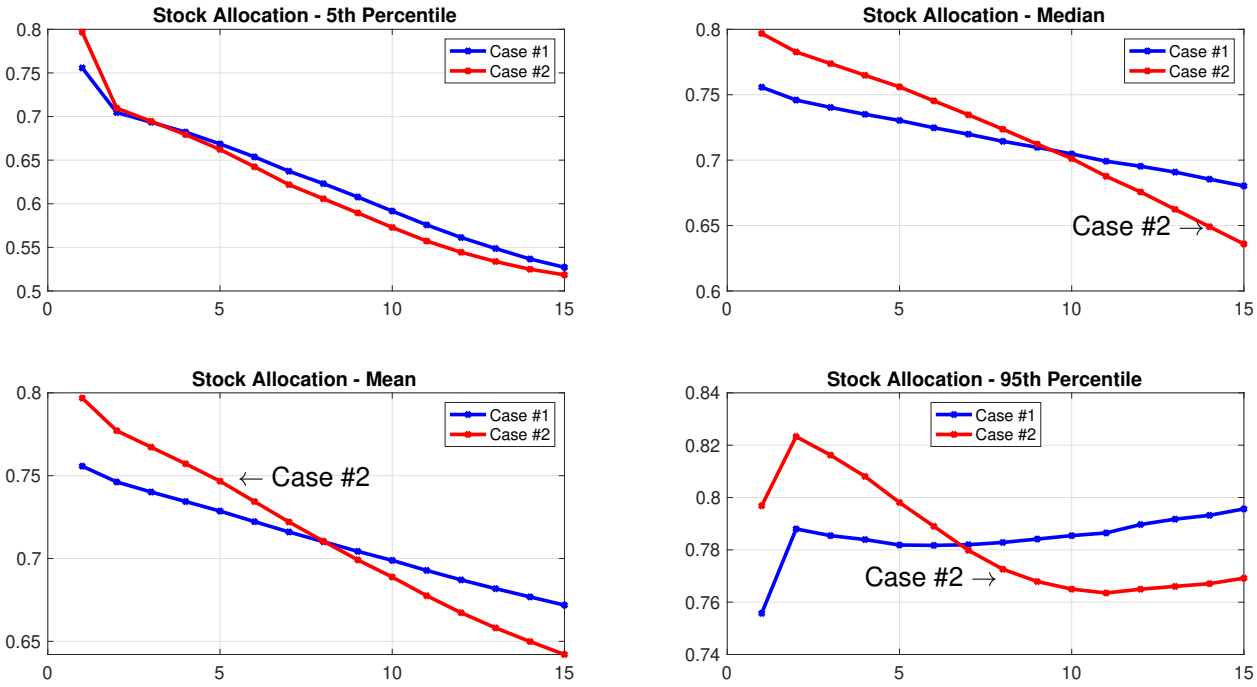


Figure 2.12: Stock allocation for the two cases, testing results

In conclusion, the results shown in this section further illustrate that forward-looking bias is not a concern in the bootstrap resampling method.

2.4.6 Replacement rate example

One common measure to determine the effectiveness of a pension plan is the replacement rate, which is defined as the percentage of annual employment income replaced by retirement income. Often, the retirement income consists of two parts: the social benefits and retirement saving accounts (DC plans and tax-free investment accounts). Typically, 70% is accepted as an adequate level of replacement rate (Booth, 2004; Biggs and Springstead, 2008)

In Canada, the social benefits include the Canada Pension Plan (CPP) and Old Age Security (OAS), and in the U.S. it would be Social Security. In fact, the social benefit is a significant part of the income of retirees. In Canada, the average CPP and OAS payment amount to $20,000 per year (MacColl, 2023), which translates to 40% of the replacement rate based on the average income of $49,000 in Canada (Statistics Canada, 2022). In the United States, an earlier study (Biggs and Springstead, 2008) shows that Social Security benefits provide about 40% of replacement income. However, a more recent study by Ghilarducci et al. (2017) shows that the replacement rate from Social Security for middle-income employees (annual income of $40,000 - $115,000) is only 29%. Nevertheless, Social Security is still a significant source of retirement income in the United States.

Consider the example of an employee making $75,000 per year in Canada (which is well

above the national average of $49,000) and contributing $10,000 per year (total employee and employer contribution) to the savings plan. According to the numerical results in Table 2.2, the employee can expect a median terminal wealth of over $900,000 following the adaptive strategy in the DC plan. A 4% annual withdrawal (Bengen, 1994) out of the terminal balance of $900,000 gives $36,000, which accounts for 48% replacement income. If we assume this employee receives the average CPP and OAS of $20,000, i.e. a replacement rate of 26% (note that this is a very conservative assumption, since $75,000 annual income is well above the national average, the actual government benefits this employee receives will be higher than average), the total retirement income of this employee will be $56,000, which is a 75% replacement rate. Similarly, a U.S. employee earning $75,000 annually will also be able to achieve more than 70% replacement income under the assumption that Social Security provides 29% of replacement rate. In fact, average American employees aged between 55-64 have an average balance of $100,000 in all retirement saving accounts combined, and having $900,000 balance in the DC account is enough to provide adequate replacement income, according to the analysis in Ghilarducci et al. (2017).

2.4.7 Historical backtest performance

As a special out-of-sample test that might be of interest to some investors, we consider the actual realized historical path from 1985 to 2015 to backtest the performance of the optimal adaptive strategy. We note that the historical path is not a path in the training data set.

From Figure 2.13, we observe that the optimal adaptive portfolio always maintains a higher wealth than the constant proportion strategy over the entire investment period. While optimizing the performance of the adaptive strategy on a specific path is not the goal of our study, it is still quite interesting to observe that the optimal adaptive strategy does better than the constant proportion strategy in real history.

Note that the adaptive strategy does show a large drawdown in 2002 and 2008. However, our objective function is posed in terms of the outperformance of the terminal wealth. We observe that the adaptive strategy outperforms, in the sense that its wealth is always above the benchmark wealth, even in 2002 and 2008. It is, of course, possible to add penalties on drawdowns in the objective function. However, this would result in less favorable terminal statistics.

The solid line without markers in Figure 2.13 illustrates the time evolution of the stock allocation on the historical path, which shows a clear contrarian pattern. When the adaptive strategy performs poorly, such as in 2002 and 2008, the strategy allocates more wealth to stocks. When the adaptive strategy performs well, the strategy decreases allocation to stocks and invests more in bonds.

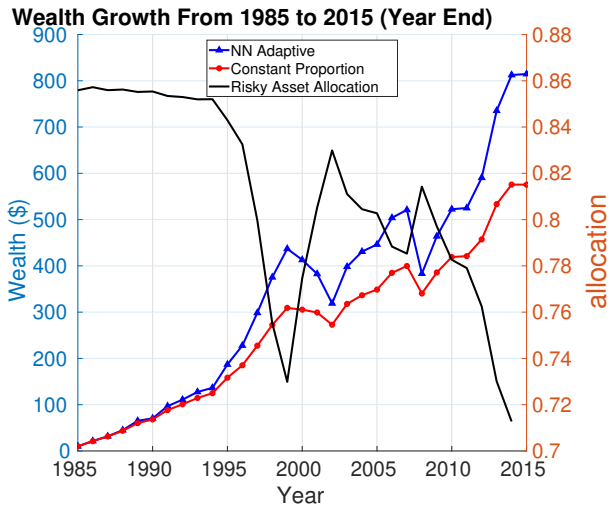


Figure 2.13: Backtest of strategy performance over the historical period from 1985-2015 (single path)

2.4.8 Comparison with the 80/20 constant proportion strategy

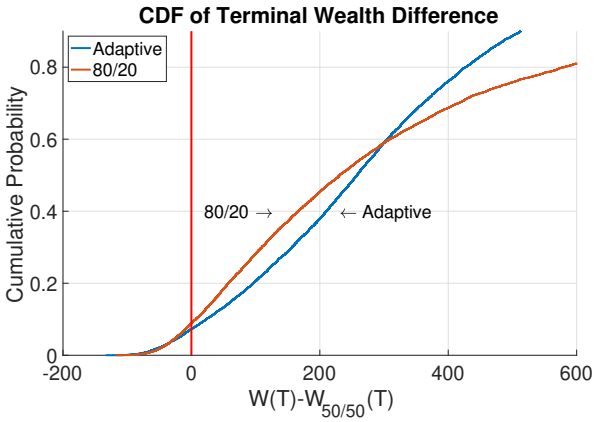
In Section 2.4.3, we have observed that while the stock allocation from the optimal adaptive strategy varies over time, its average over time is about 80%. Naturally, one may question how the optimal adaptive strategy compares with the 80/20 constant proportion strategy which invests a constant 80% of the wealth in the stocks and 20% in the bonds.

To answer this question, we compare the optimal adaptive strategy with the 80/20 constant proportion strategy. Recall that in Section 2.4.3, the optimal adaptive strategy is trained on bootstrap resampled data with the expected blocksize of 0.5 years and the test data set is bootstrap resampled data with the expected blocksize of 2 years. We compare the optimal adaptive strategy and 80/20 strategy on the same test data set.

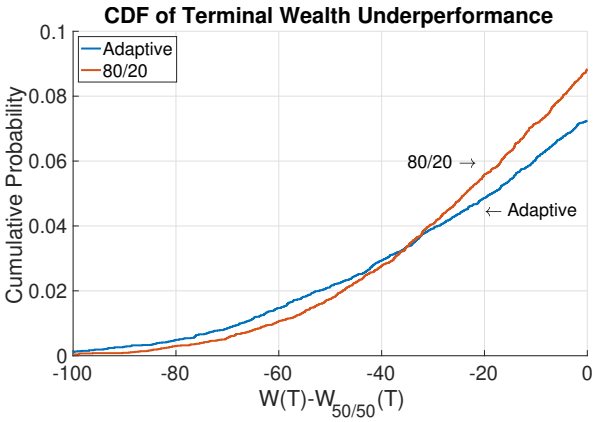
In Figure 2.14, we plot CDFs of $W_{NN}(T) - W_{50/50}(T)$ and $W_{80/20}(T) - W_{50/50}(T)$, i.e., the wealth difference of the optimal neural network adaptive strategy and the 80/20 strategy from the 50/50 strategy respectively.

We observe that the optimal adaptive strategy controls tail risk better than the 80/20 strategy. Specifically, the probability of the optimal adaptive strategy underperforming the 50/50 strategy is lower than that of the 80/20 strategy. When underperformance against the 50/50 strategy occurs, the magnitude of underperformance for the optimal adaptive strategy is slightly smaller than the magnitude of underperformance for the 80/20 strategy, as shown in Figure 2.14b.

It is worth noting that the 80/20 strategy has more upside than the optimal adaptive strategy. However, less upside is a natural result of our choice of the double-sided penalty objective function. As reflected in the asymmetric objective function (2.6), our goal is not to achieve extremely large outperformance over the 50/50 strategy, but to reach the elevated



(a) CDF of terminal wealth difference $W(T) - W_{50/50}(T)$, $W(T)$ is either $W_{NN}(T)$ or $W_{80/20}(T)$



(b) CDF of terminal wealth difference - enlarged for underperformance

Figure 2.14: CDF of wealth difference of both strategies (optimal adaptive and 80/20 constant proportion) over the 50/50 strategy

target with high probability and to control the downside risk. The optimal adaptive strategy achieves those goals better than the 80/20 strategy. To better demonstrate this, we plot the following CDF of outperformance of both strategies over the elevated target $e^{sT} \cdot W_{50/50}(T)$, in Figure 2.15a.

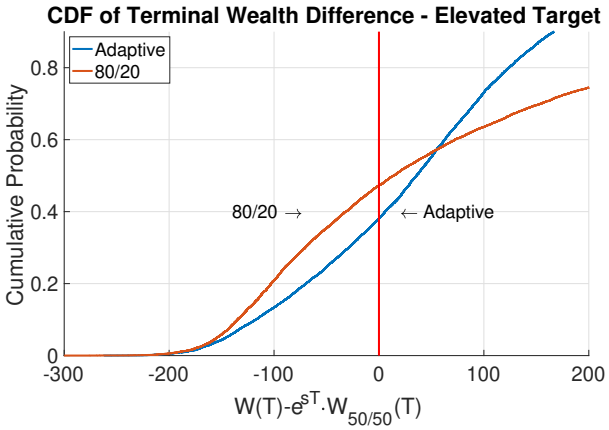
We can observe that the optimal adaptive strategy has a lower probability of underperforming the elevated target (37.3%) than the 80/20 strategy (46.8%). This means the optimal adaptive strategy is more likely to reach the elevated target and thus achieve the pre-determined annual outperformance spread.

Moreover, we observe from the enlarged CDF plot in Figure 2.15b that the optimal adaptive strategy controls underperformance better than the 80/20 strategy, in the sense that the optimal adaptive strategy underperforms less than the 80/20 strategy when the elevated target is not met.

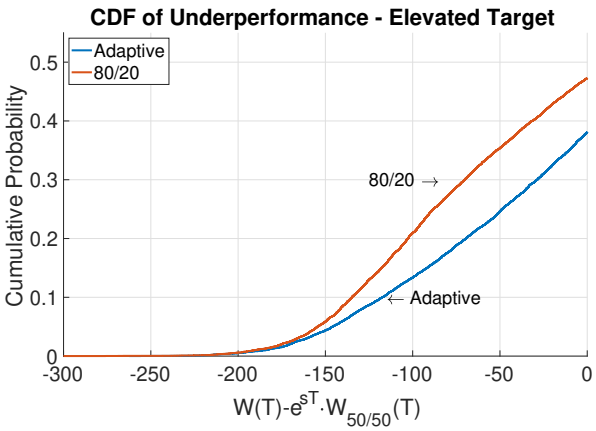
In summary, the optimal adaptive strategy is better at tracking the elevated target as desired than the 80/20 strategy, despite that both strategies have a similar average stock allocation over time.

2.5 Conclusions

In this chapter, we proposed a data-driven framework for computing the optimal asset allocation for outperforming a stochastic benchmark target based on market asset return observations. The dynamic asset allocation problem is solved directly assuming a neural network representation for the optimal control, without using dynamic programming.



(a) CDF of terminal wealth difference $W(T) - e^{sT} \cdot W_{50/50}(T)$, $W(T)$ is either $W_{NN}(T)$ or $W_{80/20}(T)$



(b) CDF of terminal wealth difference over the elevated target - enlarged for underperformance

Figure 2.15: CDF of wealth difference of both strategies (optimal adaptive and 80/20 constant proportion) over the elevated target $e^{sT} \cdot W_{50/50}(T)$

In addition, we designed an asymmetric distribution shaping objective function which is capable of producing an optimal strategy that can yield significantly larger median terminal wealth than the target, with only a small probability (and magnitude) of underperformance. We emphasize that the methodology can encompass a wide class of objective functions, which can be tailored to the risk preferences of individual investors.

We used block bootstrap resampling to augment historical financial market data. The training data and testing data are generated by block bootstrap resampling from market asset returns. This leads to a data-driven approach for determining the optimal dynamic asset allocation, avoiding the need to postulate a parametric asset price model as well as model parameter estimations. We further provided mathematical justifications for using block bootstrap resampling to generate both training and testing data sets.

The proposed method was illustrated in the DC pension allocation problem, which is a practically relevant and important problem on its own. We evaluated and analyzed the performance of the optimal neural network adaptive strategy based on the CRSP 3-month Treasury bill (T-bill) index for the defensive asset and the CRSP cap-weighted total return index for the risky asset from *1926:1-2015:12*. We showed that the adaptive strategy following the neural network model achieves the desired outperformance over the stochastic benchmark target, and is financially intuitive and easily implementable, thus demonstrating the empirical importance of the proposed methodology.

Chapter 3

Robustness of the neural network methodology

In this chapter, we discuss the robustness associated with the proposed neural network methodology for solving the portfolio optimization problem for outperforming a stochastic benchmark. While robustness can be discussed from various perspectives, we focus on the following two angles.

(i) The robustness of the neural network methodology - whether the neural network methodology produces a strategy that outperforms the benchmark strategy when certain settings and assumptions of the investment problem are modified.

(ii) The robustness of the neural network strategy - whether the neural network strategy displays consistent outperformance over the benchmark strategy on different testing data sets.

We first discuss the robustness of the neural network methodology.

In Chapter 2, we demonstrated the performance of the neural network model under a DC pension plan example where the objective is to outperform a 50/50 constant weight benchmark portfolio. While the 50/50 benchmark portfolio is a common choice of a DC plan portfolio, the target date fund concept has gained much popularity in recent years. As of 2021, the target date fund size has reached $3.27 trillion in assets (Correia, 2022). A target date fund typically sets a fixed target date for retirement, and the fund reduces stock allocation following a deterministic schedule as the date approaches the target date. For example, a common glide-path strategy allocates a stock allocation fraction of $\frac{110-\text{age}}{100}\%$.

Due to the recent popularity of target date funds, we choose target date funds as the benchmark and show that the neural network strategy still outperforms the benchmark. This demonstrates the robustness of the neural network method with respect to the choice of the benchmark strategy, i.e., the neural network methodology is capable of producing strategies that consistently outperform the benchmark for various choices of the benchmark strategy.

In the numerical experiments in Chapter 2, we used the historical returns of the cap-weighted U.S. stock index and 3-month U.S. T-bill as the underlying source of data. To assess how the proposed neural network methodology generalizes on other data sources, we consider the alternative data set that uses the equal-weighted U.S. stock index and 10-year U.S. T-bond index. By training and testing the neural network model on an alternative set of asset returns, we show that the neural network strategy outperforms the benchmark portfolio under this alternative data set with different choices of underlying assets, demonstrating that the methodology is robust to the choices of underlying data sources.

We then discuss the robustness of the neural network strategies. While typical machine learning applications are built upon the assumption that the testing data set shares the same distribution as the training data set, it is of practical interest to practitioners to have some out-of-distribution guarantee since this assumption is sometimes too good to be true. Therefore, we conduct experiments to train and test the neural network model on data sets from different distributions, hoping to provide some assurance of the robustness of the relative performance of the strategy for practitioners.

In the first experiment, we train the neural network model on a synthetic data set generated from a parametric model and test the learned strategy on bootstrap resampled data. Even though the data-generating process (i.e. data distribution) varies greatly, we show that the learned neural network strategy still manages to outperform the benchmark portfolio on testing data.

As Politis and Romano (1994) point out, changing the blocksize of the stationary block bootstrap resampling algorithm changes the distribution of the generated data samples. In the second experiment, we train the neural network on bootstrap resampled data with a certain blocksize and test the learned strategy on bootstrap resampled data with different blocksizes. We show that the neural network strategy trained on a certain blocksize performs consistently well on testing data generated from different blocksizes.

In the final experiment, we train the neural network model on bootstrap resampled data from the historical returns but test on bootstrap resampled data assuming future stock returns are reduced by 300 bps relative to historical returns. This test is to address some concerns practitioners have regarding the potential long-term effect of COVID-19 on the global economy. Surprisingly, under such adverse assumptions, we show that the neural network strategy still achieves a more desirable terminal wealth distribution than the benchmark on the testing data set.

In summary, in this chapter, we make the following contributions.

- We demonstrate the robustness of the neural network approach, by showing that with different choices of the benchmark portfolio (i.e. target date funds) or underlying data sources, the neural network model achieves desirable outperformance.
- We demonstrate the robustness of the neural network strategies by showing that the learned strategy achieves consistent outperformance on testing data sets from different distributions than the training data sets.

3.1 Alternative benchmark: target date funds

In recent years, target date funds (also known as life-cycle funds) have gained much popularity among investors. Target date funds operate with the premise that the investor retires at a certain target date. For target date funds, the asset allocation is adjusted as the calendar time gets closer to the target date. Often, the fund allocates between a stock fund and a bond fund. As reported in Correia (2022), the size of the target date funds in the U.S. has reached $3.27 trillion in total assets, as of the end of 2021.

Typically, target date funds employ a deterministic *glide-path* style of asset allocation, in which the fund maintains a high percentage of stock allocation in the earlier phase of the investment. As time goes by, the stock allocation decreases and the bond allocation increases. For example, the 40-year Vanguard target date fund (Donaldson et al., 2015) starts with a 90% stock allocation for the first 15 years, and gradually decreases the stock allocation to 50% at year 40 (the decrease is almost linear as observed from Figure 3.1).[1] A major part of target date funds' popularity comes from this glide-path design, as it fits well with the common belief that younger investors can better withstand market risk than older investors.

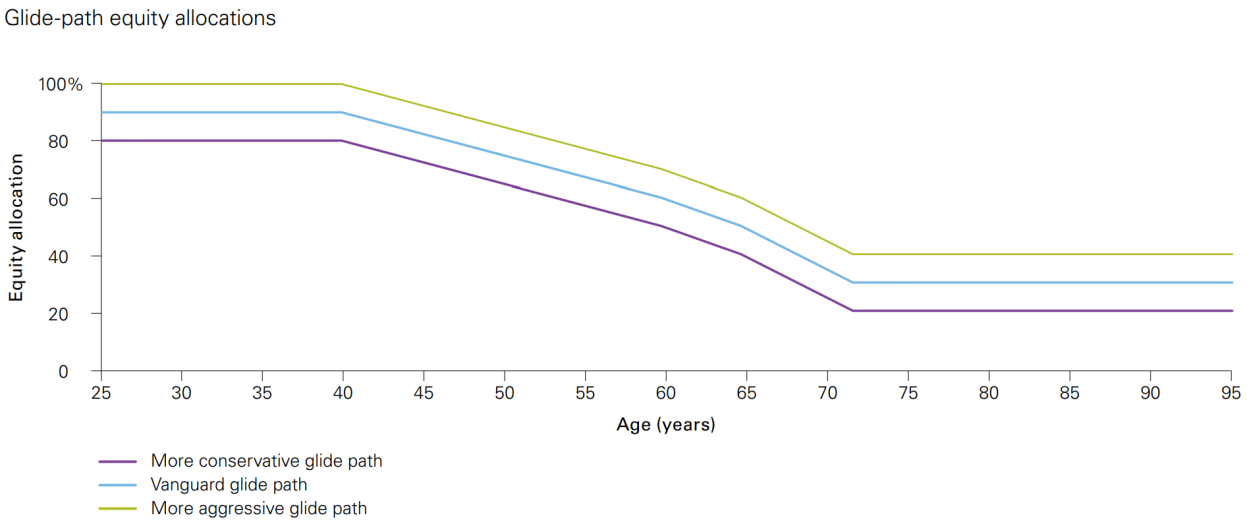


Figure 3.1: Target date fund stock allocation from Vanguard (Donaldson et al., 2015)

3.1.1 Target date fund and constant weight portfolios

However, recent research, based on empirical (Arnott et al., 2013; Esch and Michaud, 2014) and theoretical work (Graf, 2017; Forsyth and Vetzal, 2019) suggests that the purported

[1]For simplicity, we have lumped together U.S. and International stocks as an allocation to stocks, and the total allocation to U.S. bonds, international bonds and TIPS as an allocation to bonds

advantages of target date funds may have been oversold. Particularly, this research indicates that the terminal wealth distributions of a deterministic glide path and a constant proportion strategy with the same expected terminal wealth are virtually indistinguishable.

In order to confirm this analysis, we have determined that a constant weight strategy with 73% in stocks and 27% in bonds has approximately the same expected terminal wealth as the Vanguard glide path in Figure 3.1 (evaluated on bootstrap resampled data).

We then empirically compute the terminal wealth cumulative distribution function of the Vanguard target date fund and 73/27 constant proportion strategy using historical bootstrapped resampled data. From Figure 3.2, we can observe that the terminal wealth distributions of the two strategies are almost identical. Therefore, outperforming a target date fund in terms of terminal wealth distribution is essentially the same problem as outperforming a constant proportion strategy, if the evaluation is based on the terminal wealth distribution. However, the observation of identical CDFs of wealth is not expected to hold at intermediate times.

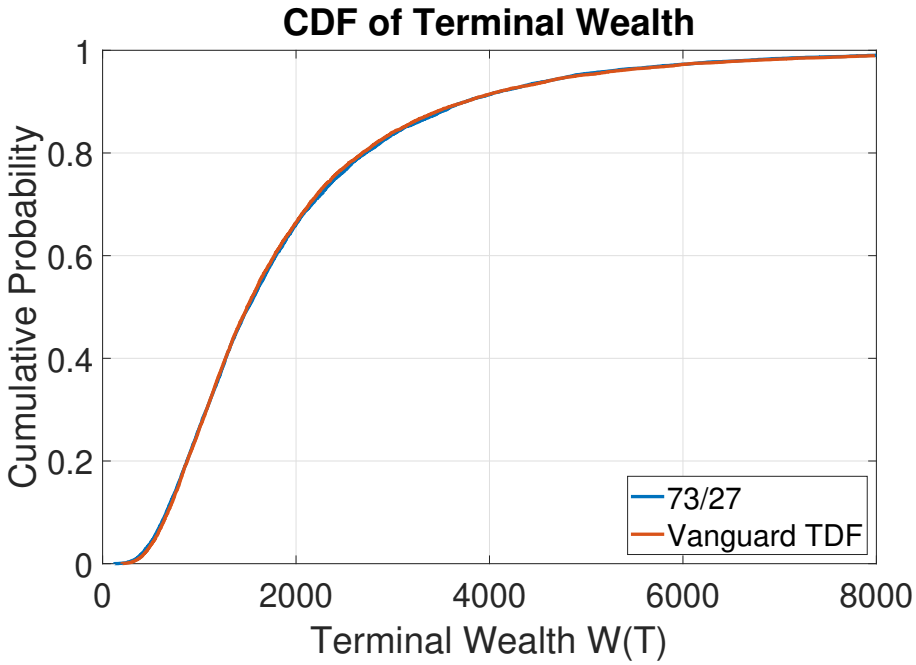


Figure 3.2: CDF of terminal wealth of Vanguard TDF and 73/27 strategy

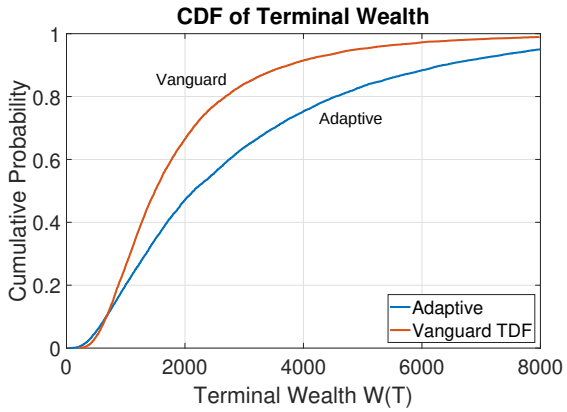
3.1.2 Outperforming the Vanguard target date fund

Nevertheless, outperforming the Vanguard target date fund or the 73/27 constant proportion strategy appears to be very challenging if we retain the constraint that use of leverage is not permitted. This is simply because such strategies are already heavy in stocks, and thus inevitably the learned strategy needs to be heavier in stocks in order to achieve a higher

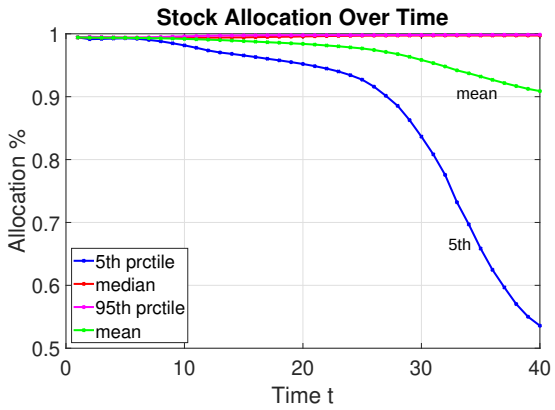
expected terminal wealth, but the no-leverage constraint imposes an upper bound on the stock holdings.

As an experiment, we train the model on bootstrap resampled data with the Vanguard target date fund as the benchmark and set the outperformance spread in the elevated target to be 50 basis points. The learned adaptive strategy does show a more attractive terminal wealth distribution compared to the Vanguard target date fund, as the CDF of the adaptive strategy shown in Figure 3.3a demonstrates partial stochastic dominance over the target date fund, with a more right-skewed CDF and a slightly worse left tail. However, we can also observe from Figure 3.3b that the learned strategy has a median stock allocation of almost 100%, and a mean allocation above 90%. In other words, half of the time the strategy simply allocates all wealth to the stock, which makes the learned adaptive strategy seem quite trivial and not as adaptive as we expect.[2] This happens simply because stock-heavy allocation nature of the Vanguard target date fund leaves little room for improvement and thus forces the adaptive strategy to go all stock so that the outperformance spread of 50 bps can be achieved.

We remark that, in terms of terminal wealth distribution, we could expect more interesting results if we allowed the use of leverage. However, this is usually not advisable in a retirement savings account. We will consider leveraged portfolios in later chapters.



(a) CDF of terminal wealth - adaptive strategy vs Vanguard TDF



(b) Stock allocation of learned adaptive strategy

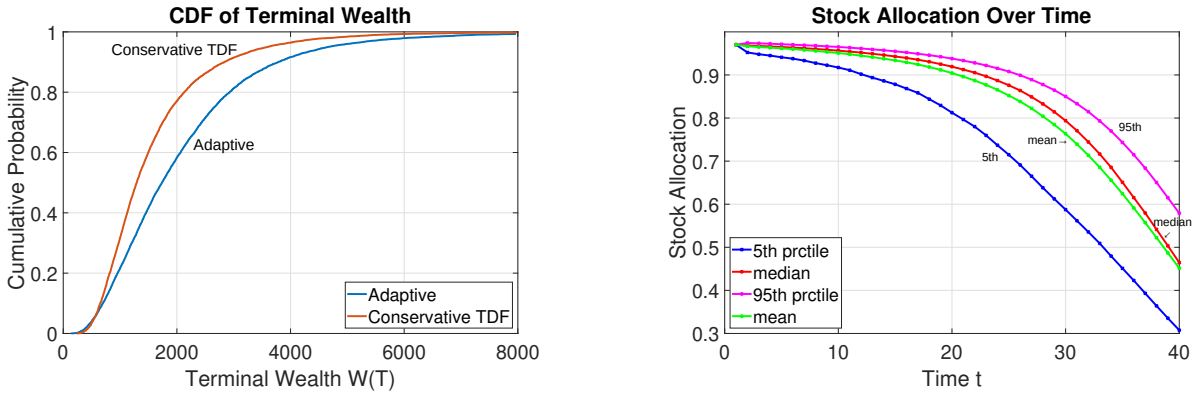
Figure 3.3: Testing results on bootstrap resampled data with $\hat{b} = 2$ years. The neural network is trained on bootstrap resampled data with $\hat{b} = 0.5$ years.

[2]We remark that such a strategy is still better than a strategy that is always 100% stock allocation, when evaluated under the two-sided objective function (2.6) and in terms of tail risk.

3.1.3 Outperforming a conservative target date fund

In order to illustrate the capability of our proposed framework in a more meaningful way, we choose a more conservative target date fund as the benchmark since it provides more room for improvement. The target date of the benchmark strategy is set to be 40 years from initiation. In the first 15 years, the fund allocates 80% in stocks and 20% in bonds. After the first 15 years, the stock allocation linearly decreases to 40% at the target date, while the bond allocation increases accordingly. In other words, this benchmark strategy shifts the stock allocation of the Vanguard 40-year target date fund down by 10%. We remark that this conservative benchmark strategy is used in the Vanguard report (Donaldson et al., 2015) and described as the *more conservative* target date fund. We also note that even for this so-called conservative target date fund, the time-average stock allocation over 40 years is still about 68% and thus has a substantial amount of market exposure.

In the next experiment, we set the outperformance target spread s in the objective function (2.6) to be 50 basis points. As in Section 2.4.3, the parameters of the neural network are trained on bootstrap resampled data with the expected blocksize of 0.5 years, and tested on bootstrap resampled data with an expected blocksize of 2 years. The only difference here is that the benchmark strategy is a target date fund, instead of a constant proportion strategy.



(a) CDF of terminal wealth - adaptive strategy vs conservative TDF

(b) Stock allocation of learned adaptive strategy

Figure 3.4: Testing results on bootstrap resampled data with $\hat{b} = 2$ years. The neural network is trained on bootstrap resampled data with $\hat{b} = 0.5$ years. The benchmark is the conservative Vanguard strategy.

First and foremost, we can observe from Figure 3.4a that the learned adaptive strategy achieves partial stochastic dominance over the conservative target date fund in terms of the terminal wealth distribution. When we examine the actual allocation of the adaptive strategy in Figure 3.4b, we can see that the adaptive strategy tends to hold more stocks in the earlier periods. This establishes an early advantage over the benchmark conservative

target date fund. Once the advantage is established, the adaptive strategy de-risks (shifts to bonds) more aggressively compared to the linear decrease in stock allocation in the target date fund.

Such asset allocation behavior also explains why it was difficult for the framework to learn an interesting strategy when benchmarking with the more aggressive Vanguard target date fund. The default Vanguard target date fund starts with 90% stock allocation and forces the adaptive strategy to full stock allocation so that the adaptive strategy can establish an early advantage.

In conclusion, we have shown that:

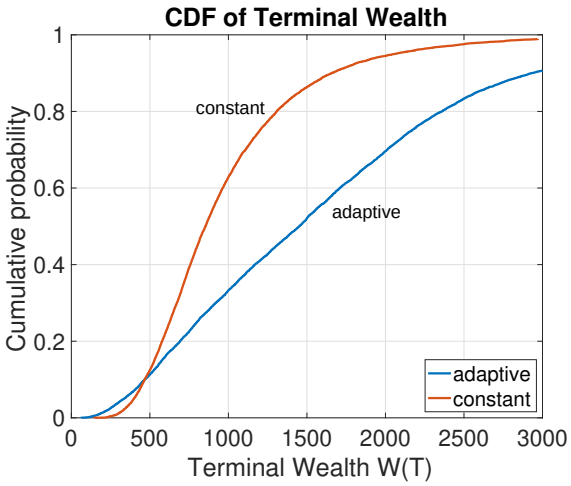
- Outperforming the Vanguard target date fund will lead to an almost all stock strategy, as the Vanguard target date fund is stock-heavy and leaves little room for learning a non-trivial adaptive strategy (assuming that a no-leverage constraint is imposed).
- When choosing a more conservative target date fund as the benchmark strategy, we are able to learn a non-trivial adaptive strategy that outperforms the benchmark target date fund with high probability and has a better terminal wealth distribution. Note that the more conservative glide path still has a time-averaged allocation fraction in stocks of about 68%. Overall, this illustrates the robustness of the neural network approach with respect to a different benchmark portfolio for achieving the desired outperformance over the benchmark.

3.2 Results from alternative data sets

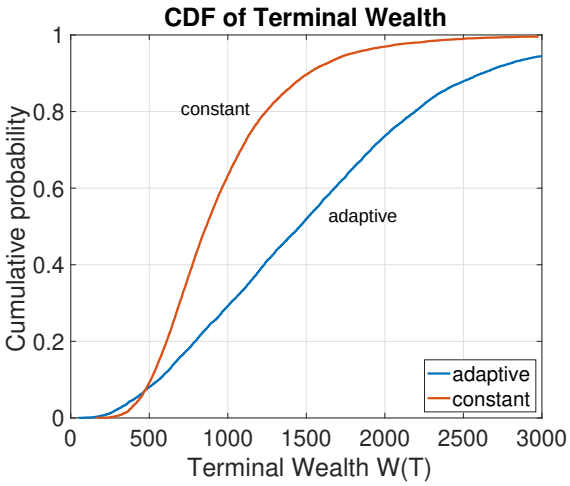
In this section, we show the results based on alternative historical data sets - the equal-weighted CRSP stock index and the 10-year treasury bond index, instead of using a mix of the cap-weighted CRSP stock index and 3-month T-bill index as in Chapter 2. Historically, the equal-weighted stock index has shown higher average return and volatility than the cap-weighted stock index, which has been attributed to equal-weighted portfolios having higher exposure to value, size, and market factors (Plyakha et al., 2021). Similarly, the 10-year T-bond has not always had the same behavior as the 3-month T-bill. Therefore, it will be interesting to see if the neural network methodology yields similar performances on this alternative data set. Same as in numerical experiments in Chapter 2, we revert back to using the 50/50 strategy as the benchmark.

We train the neural network model on bootstrap resampled data from the alternative data sets with an expected blocksize of 0.5 years, with the proposed asymmetric objective function (2.6). We then test the learned adaptive strategy on bootstrap resampled data from the alternative data sets with an expected blocksize of 2 years.

From Figure 3.5, we can clearly see that the learned adaptive strategy achieves partial stochastic dominance over the benchmark in both training and testing. From Figure 3.6, we

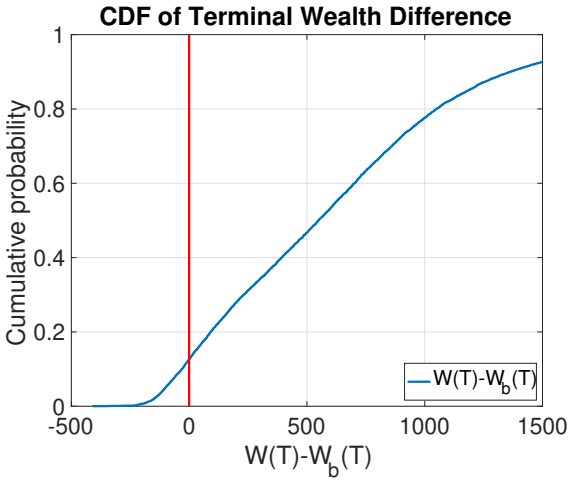


(a) Training on bootstrap data with $\hat{b}$=0.5 years

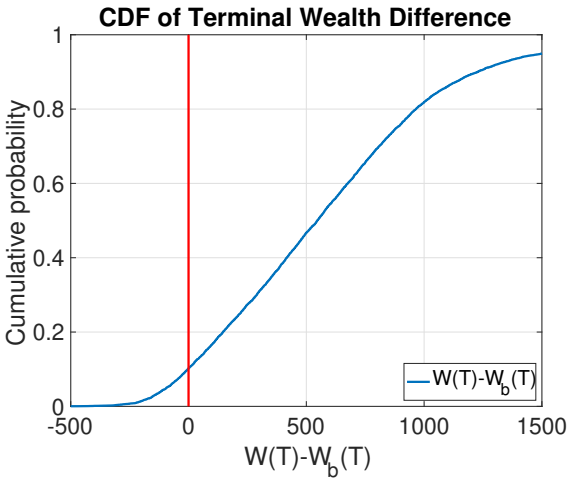


(b) Testing on bootstrap data with $\hat{b}$=2 years

Figure 3.5: CDF of terminal wealth - equal-weighted stock index and 10-year T-bond index



(a) Training on bootstrap data with $\hat{b}$=0.5 years



(b) Testing on bootstrap data with $\hat{b}$=2 years

Figure 3.6: CDF of terminal wealth difference - equal-weighted stock index and 10-year T-bond index, where $W_b(T)$ indicate the terminal wealth of the constant proportion benchmark strategy

can see that the adaptive strategy outperforms the benchmark strategy with more than 90% probability in both training and testing. Such results demonstrate that the framework is capable of learning a good adaptive strategy that outperforms the benchmark strategy with different underlying historical data sets.

3.3 Strategy trained on synthetic data

In this section, we generate synthetic data from a parametric model calibrated to historical data. We then test the strategy on bootstrap resampled data. Clearly, the synthetic data from the parametric model will have a different distribution compared to the resampled data.

The synthetic data is generated based on a jump-diffusion stochastic process. Let $S(t)$ and $B(t)$ respectively denote the wealth invested in the stocks and bonds at time t, $t \in [0, T]$. Specifically, we will assume that $S(t)$ represents the unit amount invested in a broad stock market index (CRSP cap-weighted index), while $B(t)$ is the unit amount invested in short-term default-free government bonds (in our case, the 3-month T-bill).

Recall that $t^- = t - \epsilon, \epsilon \to 0^+$, i.e. t^- is the instant of time before t, and let ψ be a random number representing a jump multiplier. When a jump occurs, $S(t) = \xi S(t^-)$. Allowing discontinuous jumps lets us explore the effects of severe market crashes on stock holding and nonnormal returns. We assume that ξ follows a double exponential distribution (Kou, 2002; Kou and Wang, 2004). If a jump occurs, p_{up} is the probability of an upward jump, while $1 - p_{up}$ is the chance of a downward jump. The density function for $y = \log \xi$ is

$$f(y) = p_{up}\eta_1 e^{-\eta_1 y}\mathbf{1}_{y\geq 0} + (1 - p_{up})\eta_2 e^{\eta_2 y}\mathbf{1}_{y\leq 0}. \tag{3.1}$$

For future reference, note that

$$E[y = \log \xi] = \frac{p_{up}}{\eta_1} - \frac{(1 - p_{up})}{\eta_2},\ E[y = \xi] = \frac{p_{up}\eta_1}{\eta_1 - 1} + \frac{(1 - p_{up})\eta_2}{\eta_2 - 1}. \tag{3.2}$$

We assume that $S(t)$ evolves according to

$$\frac{dS(t)}{S(t^-)} = (\mu - \lambda E[\xi - 1])dt + \sigma dZ + d\Big(\sum_{i=1}^{\pi_t}(\xi_i - 1)\Big), \tag{3.3}$$

where μ is the (uncompensated) drift rate, σ is the volatility, dZ is the increment of a Wiener process, π_t is a Poisson process with positive intensity parameter λ, and ξ_i are i.i.d. positive random variables having distribution (3.1). Moreover, ξ_i, π_t, and dZ are assumed to all be mutually independent.

We assume that the dynamics of the amount $B(t)$ invested in the defensive asset are

$$dB(t) = rB(t)dt, \tag{3.4}$$

where r is the (constant) rate. This is obviously a simplification of the real bond market. We remind the reader that, ultimately, our neural network method is entirely data-driven and is agnostic to the data-generating model.

Based on (3.3) and (3.4), we use the methods in (Dang and Forsyth, 2016) to calibrate the process parameters. We use a threshold technique (Cont et al., 2011) to identify jump frequency and distribution, and the methods in (Dang and Forsyth, 2016) to determine the remaining parameters. Annualized estimated parameters for the cap-weighted stock index are provided in Table 3.1.

μ	σ	λ	p_{up}	η_1	η_2	r
Real CRSP Cap-Weighted Stock Index and 3-month T-bill Index						
.08889	.14771	.32222	0.27586	4.4273	5.2613	0.00827

Table 3.1: Estimated annualized parameters for double exponential jump-diffusion model. Cap-weighted index, deflated by the CPI. Sample period 1926:1 to 2015:12.

We then generate the synthetic data based on the parametric model with the calibrated parameters through Monte Carlo simulations.

We test the performance of the strategy trained on synthetic data on historical bootstrap data with expected blocksize $\hat{b} = 2$ years, against the benchmark constant proportion 50/50 strategy. Note that the testing performance with other expected blocksizes is very similar to each other so we only show results for $\hat{b} = 2$ years.

Training Results on Synthetic Data : Market Cap Weighted					
Strategy	$E(W_T)$	$std(W_T)$	$median(W_T)$	$Pr(W_T < median(W_T^{CP}))$	$Pr(W_T < median(W_T^{NN}))$
constant proportion($p = 0.5$)	714	383	630	0.50	0.82
adaptive	1019	651	930	0.29	0.50
Testing Results on Bootstrap Data with Expected Blocksize = 2 years					
Strategy	$E(W_T)$	$std(W_T)$	$median(W_T)$	$Pr(W_T < median(W_T^{CP}))$	$Pr(W_T < median(W_T^{NN}))$
constant proportion($p = 0.5$)	679	267	630	0.50	0.84
adaptive	944	431	912	0.26	0.50

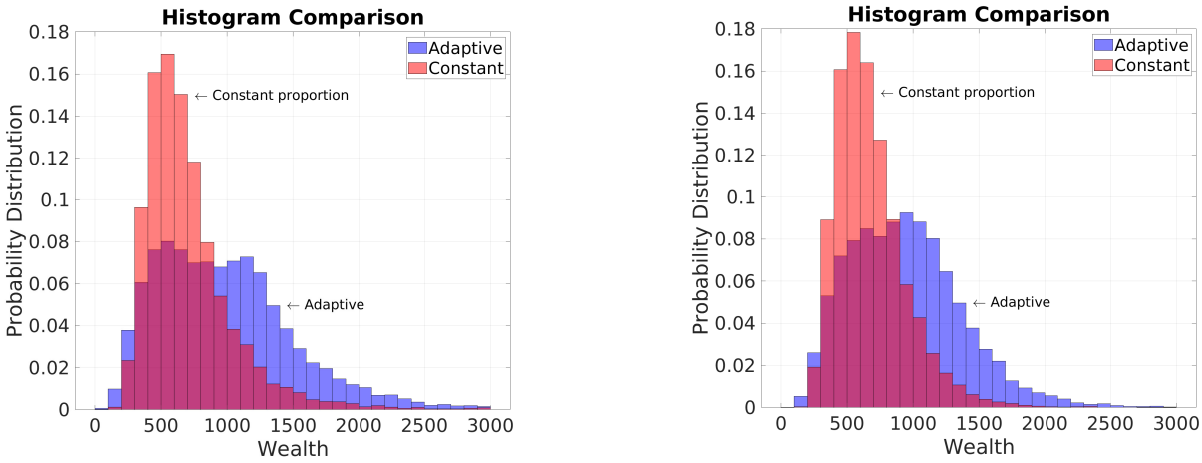
Table 3.2: Terminal wealth statistics of the adaptive strategy trained on synthetic data and tested on bootstrap resampled data with expected blocksize $\hat{b} = 2$ years

Table 3.2 shows that the adaptive strategy learned from synthetic data performs well on the test set of bootstrap resampled data. The adaptive strategy has significantly higher median and mean terminal wealth than the constant proportion strategy in both training and testing.

We do notice that in the testing results, the adaptive strategy has slightly lower mean and median terminal wealth, as well as a lower standard deviation than in training results. This is hardly surprising since the training and test data have different distributions. However, the adaptive strategy robustly outperforms the benchmark.

We observe from Figure 3.7 that the terminal wealth distributions of the adaptive strategy are consistently right-skewed and have similar shapes in training and testing, which indicates

that the NN strategy similarly outperforms the constant proportion strategy in both training and testing.



(a) Training on synthetics data

(b) Testing on bootstrap data with $\hat{b}$=0.5

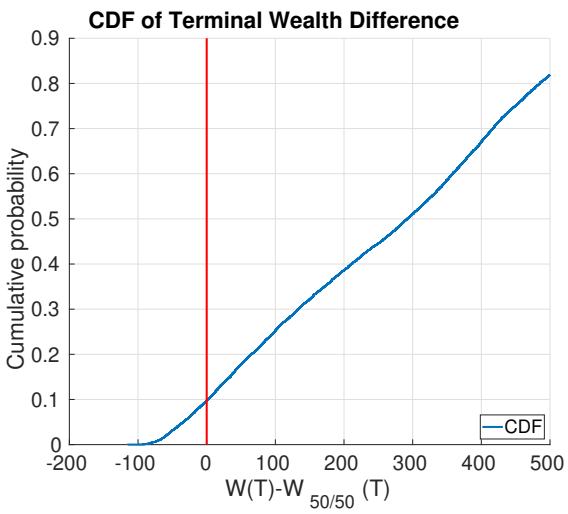
Figure 3.7: Histogram of terminal wealth. Model trained on synthetic data and tested on bootstrap resampled data with expected blocksize of 2 years

We also plot the CDF of the wealth difference between the adaptive strategy and the 50/50 strategy, $W(T) - W_{50/50}(T)$, to give a more direct comparison between the adaptive strategy and constant proportion strategy on the same paths.

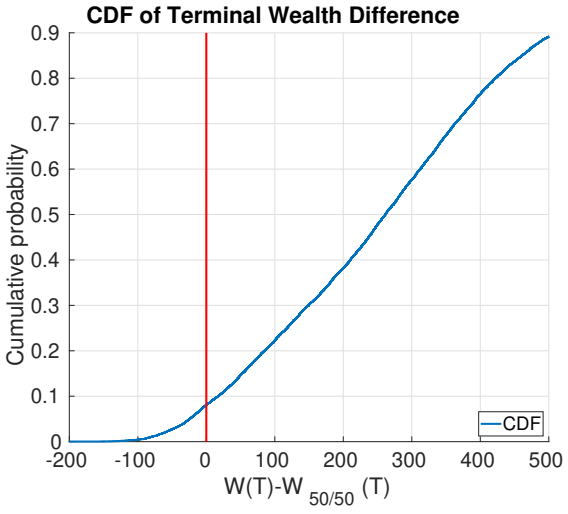
From Figure 3.8 we can see that the probability of the adaptive strategy underperforming the constant proportion strategy is less than 10% for both training and testing. When underperformance occurs, the scale of underperformance is small compared to the scale of potential outperformance. Therefore, we conclude that the adaptive strategy controls tail risks better in both training and testing, despite the fact that the training data set is synthetically generated and the testing data set is bootstrap resampled data.

In terms of the allocation strategy, we can see from 3.9 that this policy is consistent with the results in Figure 2.5b (bootstrap resampling case) in the sense that the learned strategy is a contrarian strategy that takes more risk when behind, and de-risks when ahead. We do want to point out that the heatmap in Figure 3.9b is not as smooth as the heatmap in the training case in Figure 3.9a.

We believe that this is due to the fact that, in the testing case, the strategy itself is learned from synthetic data, which has a different distribution compared with the bootstrap resampled data used in testing.

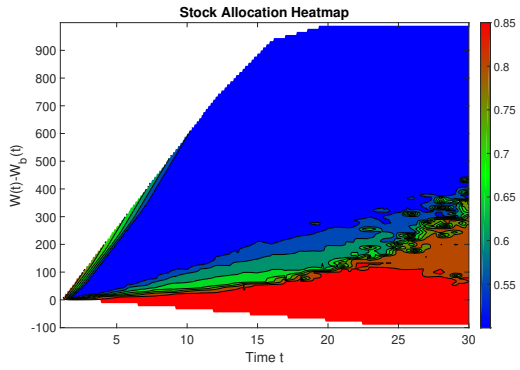


(a) Training on synthetics data

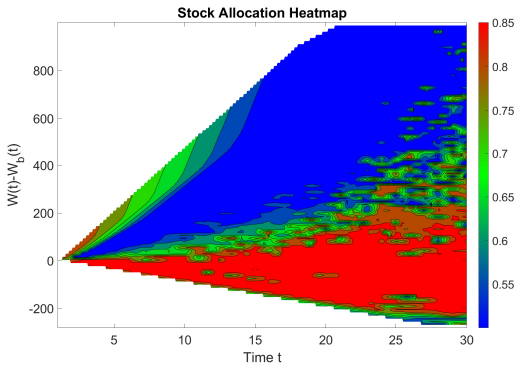


(b) Testing on bootstrap data with $\hat{b}$=2 years

Figure 3.8: CDF of terminal wealth difference $W(T) - W_{50/50}(T)$



(a) Stock allocation heatmap - training



(b) Stock allocation heatmap - testing

Figure 3.9: Stock allocation heatmap w.r.t. wealth difference

3.4 Strategy performance with different blocksizes

In this section, we test the adaptive strategy learned on bootstrap resampled data with a given blocksize on bootstrap resampled data with different blocksizes.

For illustration, we only show the testing results of the strategy learned on bootstrap resampled data with an expected blocksize of 0.5 years, where test data sets are bootstrap resampled data with blocksizes ranging from 1-10 years.

Training Results on Bootstrap Data with Expected Blocksize = 0.5 : Market Cap Weighted					
Strategy	$E(W_T)$	$std(W_T)$	$median(W_T)$	$Pr(W_T < median(W_T^{CP}))$	$Pr(W_T < median(W_T^{NN}))$
constant proportion($p = 0.5$)	678	276	624	0.50	0.86
adaptive	963	474	913	0.26	0.50
Testing Results on Bootstrap Data: Market Cap Weighted					
Strategy	$E(W_T)$	$std(W_T)$	$median(W_T)$	$Pr(W_T < median(W_T^{CP}))$	$Pr(W_T < median(W_T^{NN}))$
Expected Blocksize $\hat{b} = 1$ years					
constant proportion($p = .5$)	674	273	624	0.50	0.84
NN adaptive	955	466	909	0.27	0.50
Expected Blocksize $\hat{b} = 2$ years					
constant proportion($p = .5$)	676	263	631	0.50	0.84
NN adaptive	958	445	917	0.26	0.50
Expected Blocksize $\hat{b} = 5$ years					
constant proportion($p = .5$)	669	244	626	0.50	0.85
NN adaptive	953	409	915	0.24	0.50
Expected Blocksize $\hat{b} = 8$ years					
constant proportion($p = .5$)	669	233	632	0.50	0.87
NN adaptive	960	393	928	0.23	0.50
Expected Blocksize $\hat{b} = 10$ years					
constant proportion($p = .5$)	667	223	635	0.50	0.88
NN adaptive	961	383	928	0.22	0.50

Table 3.3: Terminal wealth statistics of the adaptive strategy trained on bootstrap resampled data with expected blocksize $\hat{b} = 0.5$ years. Tested on bootstrap resampled data with blocksizes from 1 to 10 years.

We can observe from Table 3.3 that

- The mean and the median terminal wealth of the adaptive strategy remain similar across different blocksizes.
- The adaptive strategy has a more favorable terminal wealth distribution as it is more likely to achieve a terminal wealth higher than the median terminal wealth of the constant proportion strategy.

We note that training and testing on data sets using different blocksize combinations produces qualitatively similar results. Specifically, we have also trained the neural network strategy with other blocksizes from 1-10 years, and have tested the strategy with blocksizes ranging from 0.5-10 years. Across all the tests, we observe consistent testing performance on out-of-distribution data sets (i.e. different blocksizes). The tables for additional numerical results are included in Appendix A.2.

3.5 Strategy performance under reduced stock market returns

The outbreak of the global COVID-19 pandemic has led to some concerns about the recovery of the global economy and expectations of lower future returns, especially in the stock markets. Historically, the real (geometric) returns from the U.S. equities have been around 6.6% (Dimson et al., 2020). Recent industry reports, however, estimate the future real (geometric) returns from the U.S. stock market to drop to as low as 3.8%(AQR, 2021), which is almost 300 basis points less than the average historical returns.

We remark that a lower level of stock returns does not change the main observations in Chapter 2. Specifically, in the context of outperforming a stochastic benchmark strategy, lower stock market returns adversely affect the performance of the benchmark strategy as well as the learned adaptive strategy.

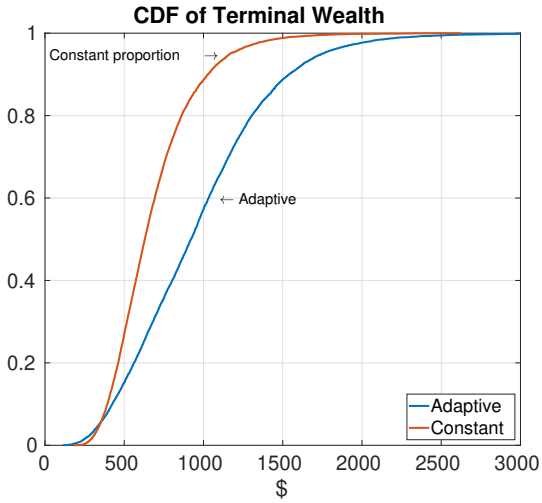
We proceed to test for robustness as follows. We train the neural network using block bootstrap resampling of the historical data. However, for testing data, we artificially reduce all historical returns by 300 bps and then use bootstrap resampled returns from this artificial data set.

Testing Results on Bootstrap Data with Original Historical Data					
Strategy	$E(W_T)$	$std(W_T)$	$median(W_T)$	$Pr(W_T < median(W_T^{CP}))$	$Pr(W_T < median(W_T^{NN}))$
constant proportion($p = 0.5$)	679	267	629	0.50	0.84
adaptive	962	449	921	0.26	0.50
Testing Results on Bootstrap Data with Historical Stock Returns Reduced by 300 bps					
Strategy	$E(W_T)$	$std(W_T)$	$median(W_T)$	$Pr(W_T < median(W_T^{CP}))$	$Pr(W_T < median(W_T^{NN}))$
constant proportion($p = 0.5$)	520	213	480	0.50	0.73
adaptive	648	344	599	0.36	0.50

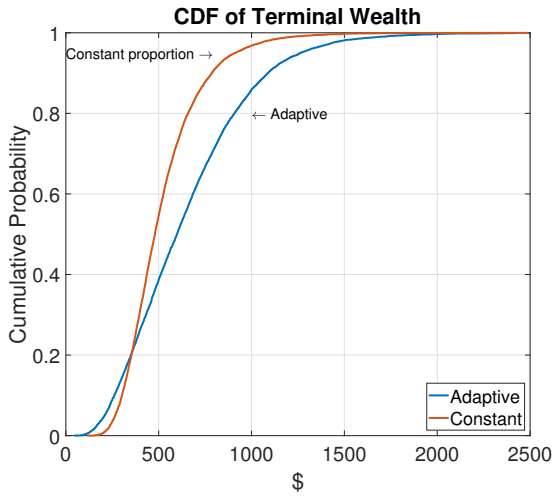
Table 3.4: Terminal wealth statistics of the optimal adaptive strategy. Table shows a comparison between testing results on bootstrap data of original historical data and historical data with stock returns adjusted by -300 bps.

In Table 3.4, we have included typical statistics on the terminal wealth in the case of reduced stock returns. We also include the original results from Section 2.4.3 for comparison. As can be seen from Table 3.4, while the terminal wealth levels of the adaptive strategy drop, the terminal wealth of the benchmark constant proportion strategy also drops significantly.

From Figure 3.10, we observe that the learned adaptive strategy has a more right-skewed terminal wealth distribution compared to the benchmark constant proportion strategy. In addition, we observe from Figure 3.11 that the adaptive strategy has a high chance of beating the benchmark strategy in pathwise comparisons. We note that, when the stock returns are adjusted for -300 bps, the advantage of the adaptive strategy decreases slightly. As can be observed from Figure 3.11, the adaptive strategy has only less than 10% of the probability of underperforming the benchmark constant proportion strategy with the original historical data, but this probability of underperforming the benchmark strategy increases to around

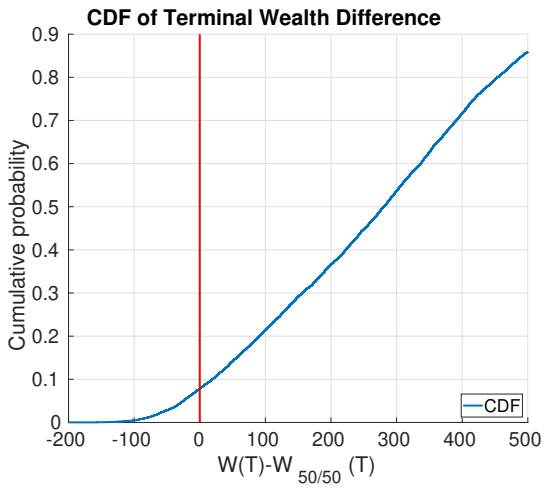


(a) Testing results with original historical data

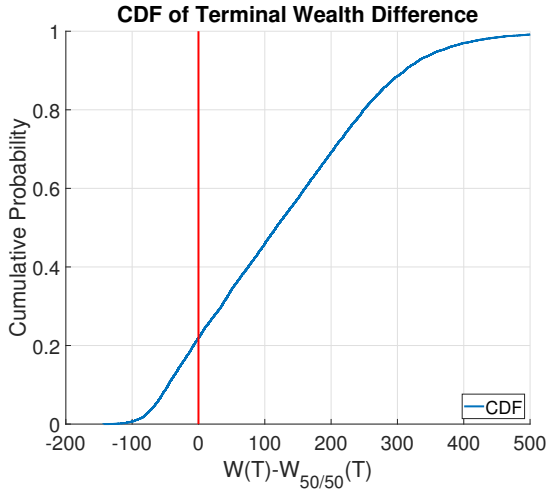


(b) Testing results with stock returns reduced by 300 bps

Figure 3.10: CDF of Terminal Wealth



(a) Testing results with original historical data



(b) Testing results with stock returns reduced by 300 bps

Figure 3.11: CDF of Terminal Wealth Difference

20% in the case of reduced stock returns. We believe that this is due to the narrower gap between stock returns and bond returns, which adversely affects the adaptive strategy since it usually starts off with a higher allocation in stocks.

However, even in such an adverse scenario, we still see the clear outperformance of the adaptive strategy as it has a more favorable terminal wealth distribution and a high chance of beating the benchmark. This alleviates the potential concern of the proposed methodology in an environment of lower stock market returns, and also shows the robustness of the neural network strategy with respect to out-of-sample data sets.

3.6 Conclusions

In this section, we discussed the robustness of the proposed neural network method from two perspectives.

We first illustrated the robustness of the neural network methodology by showing that the methodology can produce strategies that outperform benchmarks when settings and assumptions in the numerical experiments (e.g. choice of benchmark portfolio and underlying data source) are modified.

We then demonstrated the robustness of the neural network strategy by conducting several experiments and showing that the learned strategy achieves consistent outperformance over the benchmark on out-of-sample testing data sets, addressing the concerns of practitioners.

Chapter 4

Closed-form solution for outperforming a stochastic target under path-dependent objectives

4.1 Introduction

In previous chapters, we obtained numerical solutions for the problem of outperforming a benchmark, in terms of the terminal (i.e. at the end of the investment horizon) wealth. We did note that, even though the objective function was based only on terminal wealth, the strategy seemed to perform reasonably well during the entire investment period.

This leads us naturally to ask the following questions. If we can change the objective function to encourage outperformance at all times (i.e. not just at the terminal date), would it be possible to achieve even better outperformance over time? How would this affect the performance of the strategy at terminal time?

To enhance our understanding of this new objective function, we will develop closed-form solutions to a simplified problem. Usually, closed-form solutions require unrealistic assumptions, e.g. infinite leverage is allowed, and trading continues even if insolvent (i.e. negative portfolio wealth).

Nevertheless, the closed-form solutions are a useful guide to developing an understanding of numerically computed optimal controls. In later chapters, we will focus on numerical solutions for realistically constrained controls. We will see that the intuition obtained from the closed-form solution holds for the more realistic solutions.

The derivation of closed-form solutions for multi-period portfolio optimization problems, specifically for outperforming a benchmark target, typically relies on specific choices of asset price models and objective functions. For instance, Browne (1999, 2000) assume geometric Brownian motion (GBM) for asset prices and derive closed-form optimal portfolios under

various investment objectives, such as minimizing the expected time to reach a performance goal and maximizing relative wealth utility. Subsequently, the benchmarked asset allocation problem has been explored from different perspectives such as different market model assumptions and objective functions, see, e.g., Tepla (2001); Basak et al. (2006); Yao et al. (2006); Zhao (2007); Davis and Lleo (2008); Lim and Wong (2010); Bajeux-Besnainou et al. (2013); Oderda (2015); Zhang and Gao (2017); Al-Aradi and Jaimungal (2018); Nicolosi et al. (2018); Bo et al. (2021).

However, most existing literature lacks the ability to handle cash injections, a common feature in investment funds. van Staden et al. (2023b) address this limitation by employing the Hamilton-Jacobi-Bellman (HJB) technique to solve the optimal control problem. Nonetheless, they assume an asset price model where one of the assets is riskless. In this chapter, we extend the findings of van Staden et al. (2023b) by considering the assumption that all asset prices follow double-exponential jump-diffusion models, which are more appropriate for volatile economic situations with substantial uncertainty, such as the current post-COVID environment. We consider all asset returns (e.g. stock and bond) to be inflation-adjusted. Sudden changes in inflation can have a large effect on bond values, as we have seen recently.

In Chapter 2 and Chapter 3, we focused solely on the problem of outperforming a stochastic target based on terminal wealth. Specifically, we examined a DC plan setting, making the focus on terminal wealth (at retirement) suitable. However, many investors have psychological difficulty following a strategy that may be optimal in the long run but has temporary underperformance. Consequently, an objective function based solely on the terminal wealth of the portfolio becomes inappropriate in such cases.

Starting from this chapter, our focus shifts to outperforming a stochastic target with path-dependent investment objectives. Intuitively, we seek investment strategies that outperform the benchmark portfolio throughout the investment horizon.

We begin by formalizing the outperformance problem, with a specific emphasis on the intermediate performance of portfolios. We then explore several options for path-dependent objectives and discuss the cumulative difference (CD) objective function, which measures the deviation from the elevated target throughout the entire investment phase. Subsequently, we derive the closed-form solution under the CD objective function and examine the properties of this solution. Finally, we discuss the clipped-form, an approximate solution based on the closed-form solution which satisfies the realistic constraints. This is a feasible, but almost surely, sub-optimal approximation.

To summarize, our contributions in this chapter can be outlined as follows:

(i) We derive the closed-form solution to the problem of outperforming a passive benchmark portfolio under the CD objective function.

(ii) We show that the optimal strategy is contrarian in nature. In addition, it utilizes a target wealth level higher than the elevated benchmark as the decision variable for asset allocation.

(iii) We propose an approximate form of the closed-form solution that satisfies realistic constraints.

4.2 Mathematical formulation

In this section, we mathematically formulate the problem of outperforming a benchmark throughout the investment horizon. Let $[t_0(=0), T]$ denote the investment horizon, and let $W(t)$ denote the wealth (value) of the portfolio actively managed by the manager at time $t \in [t_0, T]$. We refer to the actively managed portfolio as the *active portfolio*. Furthermore, let $\hat{W}(t)$ denote the wealth of the benchmark portfolio at time $t \in [t_0, T]$. To ensure a fair assessment of the relative performance of the two portfolios, we assume both portfolios start with an equal initial wealth amount $w_0 > 0$, i.e., $W(t_0) = \hat{W}(t_0) = w_0 > 0$.

For simplicity, we assume that both the active portfolio and the benchmark portfolio can allocate among the same set of N_a assets. Let vector $\boldsymbol{S}(t) = (S_i(t) : i = 1, \cdots, N_a)^\top \in \mathbb{R}^{N_a}$ denote the asset prices of the N_a underlying assets at time $t \in [t_0, T]$. In addition, let vectors $\boldsymbol{p}^{(t)} = (p_i^{(t)} : i = 1, \cdots, N_a)^\top \in \mathbb{R}^{N_a}$ and $\hat{\boldsymbol{p}}^{(t)} = (\hat{p}_i^{(t)} : i = 1, \cdots, N_a)^\top \in \mathbb{R}^{N_a}$ denote the allocation fractions to the N_a underlying assets at time $t \in [t_0, T]$, respectively, for the active portfolio and the benchmark portfolio. In the remainder of this thesis, we consider a passive benchmark portfolio, i.e. $\hat{\boldsymbol{p}}^{(t)}$ is a pre-defined constant vector independent of time such that $\hat{\boldsymbol{p}}^{(t)} \equiv \hat{\boldsymbol{p}}$. It is possible to let $\hat{\boldsymbol{p}}^{(t)} \equiv \hat{\boldsymbol{p}}(t)$, i.e., a glide path type benchmark strategy. However, this results in complex solutions which are not amenable to building intuition. In any case, constant weight benchmarks are very common in practice.

From an optimal control perspective, the allocation vector $\boldsymbol{p}^{(t)}$ is regarded as the value of the feedback control of the system (which affects the outcome of the system, i.e. the value evolution of the portfolio), and that $\boldsymbol{p}$ is assumed to be a function of the state variables which are the portfolio values and time. Mathematically, $\boldsymbol{p}^{(t)} = \boldsymbol{p}(\boldsymbol{X}(t)) = (p_i(\boldsymbol{X}(t)) : i = 1, \cdots, N_a)^\top \in \mathbb{R}^{N_a}$, where $\boldsymbol{X}(t) \in \mathcal{X} \subseteq \mathbb{R}^{N_x}$ is the vector representing the state variables at time t. In this thesis, we consider the particular problem of outperforming a passive portfolio, in which $\boldsymbol{X}(t) = \big(t, W(t), \hat{W}(t)\big)^\top$, i.e., the state variables are the wealth of both portfolios and time. Our goal is to find the optimal control function (i.e. allocation strategy) $\boldsymbol{p} : \mathcal{X} \mapsto \mathbb{R}^{N_a}$ so that the relative performance of the active portfolio over the benchmark portfolio is maximized.

We assume that the active portfolio and the benchmark portfolio follow the same rebalancing schedule denoted by $\mathcal{T} \subseteq [t_0, T]$. In the case of discrete rebalancing, $\mathcal{T} \subset [t_0, T]$ is a discrete set. In the case of continuous rebalancing, $\mathcal{T} = [t_0, T]$, i.e., rebalancing occurs continuously throughout the entire investment horizon.

Additionally, we assume both portfolios follow the same deterministic sequence of cash injections, defined by the set $\mathcal{C} = \{c(t),\ t \in \mathcal{T}_c\}$, where $\mathcal{T}_c \subseteq [t_0, T]$ is the schedule of the cash injections. When $\mathcal{T}_c$ is a discrete injection schedule, $c(t)$ is the amount of cash injection at t. In the case of continuous cash injections, i.e., $\mathcal{T} = [t_0, T]$, $c(t)$ is the rate of cash injection at

t, i.e., the total cash injection amount during $[t, t+dt]$ is $c(t)dt$, where dt is an infinitesimal time interval. For simplicity, we assume that $\mathcal{T}_c = \mathcal{T}$, so that the cash injections schedule is aligned with the rebalancing schedule. At $t \in \mathcal{T}$, $W(t)$ and $\hat{W}(t)$ always denote the wealth after the cash injection (assuming there is a cash injection event happening at t).

Denote $\mathcal{A}$ as the set of admissible strategies, which reflects the investment constraints on the controls. We assume that admissibility can vary with state and let $\{\mathcal{X}_i\colon\ i = 1, \cdots, k\}$ be a partition of $\mathcal{X}$ (the state variable space), i.e.,

$$\begin{cases} \bigcup_{i=1}^{k} \mathcal{X}_i = \mathcal{X}, \\ \mathcal{X}_i \bigcap \mathcal{X}_j = \varnothing, \forall 1 \le i < j \le k, \end{cases} \tag{4.1}$$

and $\{\mathcal{Z}_i \subseteq \mathbb{R}^{N_a}\colon\ i = 1, \cdots, k\}$ be the pre-determined corresponding value sets of feasible controls.

We say that a control $\boldsymbol{p} : \mathcal{X} \mapsto \mathbb{R}^{N_a}$ is an admissible strategy, i.e., $\boldsymbol{p} \in \mathcal{A}$, if and only if

$$\text{(Feasibility of control):} \quad \boldsymbol{p}(\boldsymbol{x}) \in \mathcal{Z}_i, \forall \boldsymbol{x} \in \mathcal{X}_i,\ \forall i \in \{1, \cdots, k\}. \tag{4.2}$$

Consider a discrete rebalancing schedule $\mathcal{T} = \{t_j,\ j = 0, \cdots, N\}$ with N rebalancing events, where $t_0 < t_1 < \cdots < t_N = T$.[1] Then, the wealth evolution of the active portfolio and the benchmark portfolio can be described by the equations

$$\begin{cases} W(t_{j+1}) = \Big(\sum\limits_{i=1}^{N_a} p_i(\boldsymbol{X}(t_j)) \cdot \frac{S_i(t_{j+1}) - S_i(t_j)}{S_i(t_j)} \Big) W(t_j) + c(t_{j+1}),\ j = 0, \cdots, N-1, \\ \hat{W}(t_{j+1}) = \Big(\sum\limits_{i=1}^{N_a} \hat{p}_i \cdot \frac{S_i(t_{j+1}) - S_i(t_j)}{S_i(t_j)} \Big) \hat{W}(t_j) + c(t_{j+1}),\ j = 0, \cdots, N-1. \end{cases} \tag{4.3}$$

In the continuous rebalancing case, $\mathcal{T} = [t_0, T]$. Let $dS_i(t)$ denote the instantaneous change in price for asset i, $i \in [1, \cdots, N_a]$.[2] Then, at $t \in \mathcal{T} = [t_0, T]$, the wealth dynamics of the active portfolio and the benchmark portfolio, following their respective strategies $\boldsymbol{p}$ and $\hat{\boldsymbol{p}}$, can be described by the equations

$$\begin{cases} dW(t) = \Big(\sum\limits_{i=1}^{N_a} p_i(\boldsymbol{X}(t)) \cdot \frac{dS_i(t)}{S_i(t)} \Big) W(t_j) + c(t)dt, \\ d\hat{W}(t) = \Big(\sum\limits_{i=1}^{N_a} \hat{p}_i \cdot \frac{dS_i(t)}{S_i(t)} \Big) \hat{W}(t_j) + c(t)dt. \end{cases} \tag{4.4}$$

Let sets $\mathcal{W}_{\boldsymbol{p}} = \{W(t), t \in \mathcal{T}\}$ and $\hat{\mathcal{W}}_{\hat{\boldsymbol{p}}} = \{\hat{W}(t), t \in \mathcal{T}\}$ represent the trajectories of the portfolio values for the active portfolio and the benchmark portfolio, respectively,

[1] Technically, at $t = t_0$, the manager makes the initial asset allocation, rather than a "rebalancing" of the portfolio. However, despite the different purposes, a rebalancing of the portfolio is simply a new allocation of the portfolio wealth. Therefore, for notational simplicity, we include t_0 in the rebalancing schedule.

[2] For illustration purposes, here we assume $S_i(t), i \in [1, \cdots, N_a]$ follow standard diffusion processes, i.e., no jumps. We will discuss the case with jumps in detail in Section 4.3.

following the dynamics specified in equation (4.3) or (4.4). We introduce an investment metric denoted by $F(\mathcal{W}_{\boldsymbol{p}}, \hat{\mathcal{W}}_{\hat{\boldsymbol{p}}}) \in \mathbb{R}$, which quantifies the relative performance of the active portfolio in relation to the benchmark portfolio based on their respective value trajectories.

In this article, we assume the asset prices $\boldsymbol{S}(t) \in \mathbb{R}^{N_a}$ are stochastic. Consequently, the value trajectories $\mathcal{W}_{\boldsymbol{p}}$, $\hat{\mathcal{W}}_{\hat{\boldsymbol{p}}}$, and the performance metric $F(\mathcal{W}_{\boldsymbol{p}}, \hat{\mathcal{W}}_{\hat{\boldsymbol{p}}})$ are also stochastic. Therefore, when investment managers aim to optimize an investment metric, the evaluation commonly involves calculating the expectation of the random metric.

Let $\mathbb{E}_{\boldsymbol{p}}^{(t_0,w_0)}[F(\mathcal{W}_{\boldsymbol{p}}, \hat{\mathcal{W}}_{\hat{\boldsymbol{p}}})]$ denote the expectation of the performance metric F, given a specific initial wealth value $w_0 = W(0) = \hat{W}(0)$ at time $t_0 = 0$, and evaluated on the wealth trajectories following the admissible investment strategies $\boldsymbol{p} \in \mathcal{A}$ and the benchmark investment strategy $\hat{\boldsymbol{p}}$. As we assume the benchmark strategy to be predetermined, we keep the benchmark strategy $\hat{\boldsymbol{p}}$ implicit in this notation for simplicity. Subsequently, we try to solve the following stochastic optimization (SO) problem:

$$\text{(Stochastic optimization problem):} \quad \inf_{\boldsymbol{p}\in\mathcal{A}} \mathbb{E}_{\boldsymbol{p}}^{(t_0,w_0)}\big[F(\mathcal{W}_{\boldsymbol{p}}, \hat{\mathcal{W}}_{\hat{\boldsymbol{p}}})\big]. \tag{4.5}$$

In previous chapters, we have focused on measuring the performance of the portfolio at the terminal time of the investment horizon. As discussed in the introduction to this chapter, it is often desirable to have smooth outperformance during the entire investment horizon. Therefore, starting from this chapter, we evaluate the performance of portfolios from a path-dependent perspective.

We consider the following cumulative quadratic tracking difference (CD) objectives proposed in van Staden et al. (2023a):

$$(CD(\beta)): \quad \begin{cases} \inf\limits_{\boldsymbol{p}\in\mathcal{A}} \mathbb{E}_{\boldsymbol{p}}^{(t_0,w_0)}\left[\int_{t_0}^{T}\Big(W(t) - e^{\beta t}\hat{W}(t)\Big)^2 dt\right], \text{ if } \mathcal{T} = [t_0, T], & (4.6) \\ \inf\limits_{\boldsymbol{p}\in\mathcal{A}} \mathbb{E}_{\boldsymbol{p}}^{(t_0,w_0)}\left[\sum\limits_{t\in\mathcal{T}\cup\{T\}}\Big(W(t) - e^{\beta t}\hat{W}(t)\Big)^2 \Delta t\right], \text{ if } \mathcal{T} \subseteq [t_0, T], \mathcal{T} \text{ discrete.} & (4.7) \end{cases}$$

Here, β is the annualized performance premium rate, and $e^{\beta t}\hat{W}(t)$ is the elevated target used in the asymmetric objective function (2.6) proposed in Chapter 2. Note that objective (4.6) is for the continuous rebalancing case, and (4.7) for discrete rebalancing. Both (4.6) and (4.7) measure the cumulative deviation of the wealth of the active portfolio relative to the elevated target, along the rebalancing schedule throughout the entire investment horizon. (4.7) is clearly a discrete approximation of (4.6).

4.3 Closed-form solution for CD problem

In this section, we present the closed-form solution to the CD problem (4.6) under the following assumptions.

Assumption 4.3.1. *(Two assets, no friction, unlimited leverage, trading continues in insolvency, constant rate of cash injection). The active portfolio and the benchmark portfolio have access to two underlying assets, a stock index, and a constant-maturity bond index. Both portfolios are rebalanced continuously, i.e., $\mathcal{T} = [t_0, T]$. There are no transaction costs and no leverage limit. Furthermore, we assume that trading continues in the event of insolvency, i.e., when $W(t) < 0$ for some $t \in [t_0, T]$. Finally, we assume both portfolios receive constant cash injections with an injection rate of c, which means during any time interval $[t, t + \Delta t] \subseteq [t_0, T], \forall \Delta t > 0$, both portfolios receive cash injection amount of $c\Delta t$.*

Remark 4.3.1. (Remark on Assumption 4.3.1) For illustration purposes, we assume only two underlying assets. However, the technique for deriving the closed-form solution can be extended to multiple assets. We remark that unlimited leverage is unrealistic, and is only assumed for deriving the closed-form solution. We also acknowledge that it is not realistic to assume that the manager can continue to trade and borrow when insolvent. However, this assumption is typically required for obtaining closed-form solutions, see Zhou and Li (2000); Li and Ng (2000) for similar assumptions in the case of a multi-period mean-variance asset allocation problem.

Assumption 4.3.2. *(Fixed-mix benchmark strategy). We assume that the benchmark strategy is a fixed-mix strategy (also known as the constant weight or constant proportion strategy). We assume the benchmark always allocates a constant fraction of $\hat{\varrho}$ ($\in \mathbb{R}$) of the portfolio wealth to the stock index, and a constant fraction of $1 - \hat{\varrho}$ to the bond index. Let $\hat{\boldsymbol{\varrho}} = (\hat{\varrho}, 1 - \hat{\varrho})^\top \in \mathbb{R}^2$ denote the vector of allocation fractions to the stock index and the bond index.*

Finally, we assume the stock index price and bond index price follow the jump-diffusion processes described below.

Assumption 4.3.3. *(Jump-diffusion processes). Let $S_1(t)$ and $S_2(t)$ denote the deflated (adjusted by inflation) price of the stock index and the bond index at time $t \in [t_0, T]$. We assume $S_i(t)$, $i \in \{1, 2\}$ follow the jump-diffusion processes*

$$\frac{dS_i(t)}{S_i(t^-)} = (\mu_i - \lambda_i \kappa_i + r_i \cdot \mathbf{1}_{S_i(t^-)<0})dt + \sigma_i dZ_i(t) + d\Big(\sum_{k=1}^{\pi_i(t)} (\xi_i^{(k)} - 1)\Big),\ i = 1, 2. \tag{4.8}$$

Here μ_i is the (uncompensated) drift rate, σ_i is the diffusive volatility, $Z_1(t), Z_2(t)$ are correlated Brownian motions, where $\mathbb{E}[dZ_1(t) \cdot dZ_2(t)] = \rho dt$. Constant r_i is the borrowing premiums when $S_i(t^-)$ is negative.[3] *$\pi_i(t)$ is a Poisson process with positive intensity parameter λ_i. $\{\xi_i^{(k)},\ k = 1, \cdots, \pi_i(t)\}$ are i.i.d. positive random variables that describe jump multipliers associated with the assets. If a jump occurs for asset i at time $t \in (t_0, T]$, its*

[3]Intuitively, there is a premium for shorting an asset. In the closed-form solution derivation, we assume $r_i = 0$. Normally, this is relevant for the case of borrowing cash (i.e. shorting the bond) to buy stocks.

underlying price jumps from $S_i(t^-)$ *to* $S_i(t) = \xi_i \cdot S_i(t^-)$.[4] $\kappa_i = \mathbb{E}[\xi_i - 1]$. ξ_i *and* $\pi_i(t)$ *are independent of each other. Moreover,* $\pi_1(t)$ *and* $\pi_2(t)$ *are assumed to be independent.*[5]

Remark 4.3.2. (Motivation for jump-diffusion model). The assumption of stock index price following a jump-diffusion model is common in the financial mathematics literature (Merton, 1976; Kou, 2002). In addition, we follow the practitioner approach and directly model the returns of the constant maturity bond index as a stochastic process (Lin et al., 2015; MacMinn et al., 2014). As in MacMinn et al. (2014), we also assume that the constant maturity bond index follows a jump-diffusion process. During uncertain market schemes, such as the current high inflation regime in the market (as of the writing of this thesis), central banks often make rate hikes to curb inflation, which causes sudden jumps in bond prices (Lahaye et al., 2011). For an empirical discussion of the appropriateness of a jump-diffusion model for inflation-adjusted bond returns, see Forsyth et al. (2020).

Under the jump-diffusion model (4.8), the wealth processes for the active portfolio and benchmark portfolio are

$$\begin{cases} dW(t) = \Big(\sum_{i=1}^{2} p_i(\boldsymbol{X}(t^-)) \cdot \frac{dS_i(t)}{S_i(t^-)} \Big) W(t_j^-) + cdt, \\ d\hat{W}(t) = \Big(\sum_{i=1}^{2} \hat{p}_i \cdot \frac{dS_i(t)}{S_i(t^-)} \Big) \hat{W}(t_j^-) + cdt, \end{cases} \tag{4.9}$$

where $t \in (t_0, T]$, $W(t_0) = \hat{W}(t_0) = w_0$ and $X(t^-) = (t, W(t^-), \hat{W}(t^-))^\top \in \mathbb{R}^3$ is the state variable vector.

We now derive the closed-form solution of the CD problem (4.6) under Assumption 4.3.1, 4.3.2 and 4.3.3. We first present the verification theorem for the HJB integro-differential equation (PIDE) satisfied by the value function and the optimal control of the CD problem (4.6).

Theorem 4.3.1. *(Verification theorem for CD problem (4.6). For a fixed* $\beta > 0$*, assume that for all* $(t, w, \hat{w}, \hat{\varrho}) \in [t_0, T] \times \mathbb{R}^3$*, there exists a function* $V(t, w, \hat{w}, \hat{\varrho}) : [t_0, T] \times \mathbb{R}^3 \mapsto \mathbb{R}$ *and* $\boldsymbol{p}^*(t, w, \hat{w}, \hat{\varrho}) : [t_0, T] \times \mathbb{R}^3 \mapsto \mathbb{R}^2$ *that satisfy the following two properties. (i)* V *and* $\boldsymbol{p}^*$ *are sufficiently smooth and solve the HJB PIDE (4.10), and (ii) the function* $\boldsymbol{p}^*(t, w, \hat{w}, \hat{\varrho})$ *attains the pointwise infimum in (4.10) below*

$$\begin{cases} \frac{\partial V}{\partial t} + (w - e^{\beta t}\hat{w})^2 + \inf\limits_{\boldsymbol{p} \in \mathbb{R}^2} H(\boldsymbol{p}; t, w, \hat{w}, \hat{\boldsymbol{\varrho}}) = 0, \\ V(T, w, \hat{w}, \hat{\varrho}) = 0, \end{cases} \tag{4.10}$$

[4]For any functional $\psi(t)$, we use the notation $\psi(t^-)$ as shorthand for the left-sided limit $\psi(t^-) = \lim_{\Delta t \downarrow 0} \psi(t - \Delta t)$.

[5]See Forsyth (2020) for the discussion on the empirical evidence for stock-bond jump independence. Also note that the assumption of independent jumps can be relaxed without technical difficulty if needed (Kou, 2002), but will significantly increase the complexity of notations.

where

$$\begin{aligned} H(\boldsymbol{p};t,w,\hat{w},\hat{\boldsymbol{\varrho}}) =& \big(w\cdot\boldsymbol{\alpha}^\top\boldsymbol{p}+c\big)\cdot\frac{\partial V}{\partial w}+\big(\hat{w}\cdot\boldsymbol{\alpha}^\top\hat{\boldsymbol{\varrho}}+c\big)\cdot\frac{\partial V}{\partial \hat{w}}-\Big(\sum_i\lambda_i\Big)\cdot V(t,w,\hat{w},\hat{\varrho})\\ &+\frac{w^2}{2}\cdot\big(\boldsymbol{p}^\top\boldsymbol{\Sigma}\boldsymbol{p}\big)\cdot\frac{\partial^2 V}{\partial w^2}+\frac{\hat{w}^2}{2}\cdot\big(\hat{\boldsymbol{\varrho}}^\top\boldsymbol{\Sigma}\hat{\boldsymbol{\varrho}}\big)\cdot\frac{\partial^2 V}{\partial \hat{w}^2}+w\hat{w}\cdot\big(\boldsymbol{p}^\top\boldsymbol{\Sigma}\hat{\boldsymbol{\varrho}}\big)\cdot\frac{\partial^2 V}{\partial w\partial\hat{w}}\\ &+\sum_i\lambda_i\int_0^\infty V(w+p_iw(\xi-1),\hat{w}+\hat{p}_i\hat{w}(\xi-1),t,\hat{\varrho})f_{\xi_i}(\xi)d\xi. \end{aligned}\tag{4.11}$$

Here $\boldsymbol{\alpha}=(\mu_1-\lambda_1\kappa_1,\mu_2-\lambda_2\kappa_2)^\top$ *is the vector of (compensated) drift rates,* $\boldsymbol{\Sigma}=\begin{bmatrix}\sigma_1^2 & \rho\sigma_1\sigma_2\\ \rho\sigma_1\sigma_2 & \sigma_2^2\end{bmatrix}$ *is the covariance matrix, and* f_{ξ_i} *is the density function for* ξ_i*.*

Then, under Assumption 4.3.1, 4.3.2 and 4.3.3, V *is the value function and* $\boldsymbol{p}^*$ *is the optimal control for the CD problem (4.6).*

Sketch of proof. At any state $(t,w,\hat{w})\in[t_0,T]\times\mathbb{R}^2$, define the value function $V(t,w,\hat{w},\hat{\boldsymbol{\varrho}})$ to the CD problem (4.6) as

$$V(t,w,\hat{w},\hat{\boldsymbol{\varrho}})=\inf_{\boldsymbol{p}}\Big\{\mathbb{E}_{\boldsymbol{p}}\Big[\int_t^T\big(W(s)-e^{\beta s}\hat{W}(s)\big)^2ds\Big|W(t)=w,\hat{W}(t)=\hat{w}\Big]\Big\}.\tag{4.12}$$

By the dynamic programming principle, we have

$$\begin{aligned} V(t,w,\hat{w},\hat{\boldsymbol{\varrho}})=&\inf_{\boldsymbol{p}}\Big\{\mathbb{E}_{\boldsymbol{p}}\Big[\Big(V(t+\Delta t,W(t+\Delta t),\hat{W}(t+\Delta t),\hat{\boldsymbol{\varrho}})\\ &+\int_t^{t+\Delta t}\big(W(s)-e^{\beta s}\hat{W}(s)\big)^2ds\Big)\Big|W(t)=w,\hat{W}(t)=\hat{w}\Big]\Big\} \end{aligned}\tag{4.13}$$

Rearrange equation (4.13) to obtain

$$\inf_{\boldsymbol{p}}\Big\{\mathbb{E}_{\boldsymbol{p}}\Big[\Big(dV(t,w,\hat{w},\hat{\boldsymbol{\varrho}})+\int_t^{t+\Delta t}\big(W(s)-e^{\beta s}\hat{W}(s)\big)^2ds\Big)\Big|W(t)=w,\hat{W}(t)=\hat{w}\Big]\Big\}=0\tag{4.14}$$

Then, apply Itô's lemma with jumps (Cont et al., 2011), substitute dW and $d\hat{W}$ terms with (4.9), and take limits as $\Delta t\downarrow 0$, we obtain (4.10).

Note that the above results merely serve as an intuitive guide to obtain (4.10). A formal proof proceeds by using a suitably smooth test function (Øksendal and Sulem, 2007). □

Define several auxiliary variables

$$\begin{cases} \kappa_i^{(2)}=\mathbb{E}\big[(\xi_i-1)^2\big], & (\sigma_i^{(2)})^2=(\sigma_i)^2+\lambda_i\kappa_i^{(2)},\ i\in\{1,2\},\\ \vartheta=\sigma_1\sigma_2\rho-(\sigma_2^{(2)})^2, & \gamma=(\sigma_1^{(2)})^2+(\sigma_2^{(2)})^2-2\sigma_1\sigma_2\rho,\\ \phi=\frac{(\mu_1-\mu_2)(\mu_1-\mu_2+\vartheta)}{\gamma}, & \eta=\frac{(\mu_1-\mu_2+\theta)^2}{\gamma}-(\sigma_2^{(2)})^2, \end{cases}\tag{4.15}$$

then we have the following proposition regarding the optimal control of problem (4.6).

Proposition 4.3.1. *(CD-optimal control). Suppose Assumption 4.3.1, 4.3.2 and 4.3.3 are applicable, then the optimal control fraction of the wealth of the active portfolio to be invested in the stock index for the $CD(\beta)$ problem (4.6) is given by $p^*(t,w,\hat{w},\hat{\varrho}) \in \mathbb{R}$, where*

$$p^*(t,w,\hat{w},\hat{\varrho}) = \frac{1}{W^*(t)}\left[\frac{(\mu_1-\mu_2)}{\gamma}h(t;\beta,c)+\frac{(\mu_1-\mu_2+\vartheta)}{\gamma}\Big(g(t;\beta)\hat{W}(t)-W^*(t)\Big)+g(t;\beta)\hat{W}(t)\cdot\hat{\varrho}\right]. \tag{4.16}$$

Here $W^*(t)$ denotes the wealth process of the active portfolio from (4.4) following control $\boldsymbol{p}^*(t,W^*(t),\hat{W}(t),\hat{\varrho}) = \Big(p^*(t,W^*(t),\hat{W}(t),\hat{\varrho}),1-p^*(t,W^*(t),\hat{W}(t),\hat{\varrho})\Big)^\top$, where p^* is the optimal stock allocation described in (4.16), and $\hat{W}(t)$ is the wealth process of the benchmark portfolio following the fixed-mixed strategy described in Assumption 4.3.2. Here, h and g are deterministic functions of time,

$$g(t;\beta) = -\frac{D(t;\beta)}{2A(t)}, \qquad h(t;\beta,c) = -\frac{B(t;\beta,c)}{2A(t)}, \tag{4.17}$$

where A, D and B are deterministic functions defined as

$$A(t) = \frac{e^{(2\mu_2-\eta)(T-t)}-1}{(2\mu_2-\eta)}, \qquad D(t;\beta) = 2e^{\beta T}\Big(\frac{e^{-\beta(T-t)}-e^{(2\mu_2-\eta)(T-t)}}{2\mu_2-\eta+\beta}\Big), \tag{4.18}$$

and

$$\begin{aligned} B(t;\beta,c) = {} & \frac{2c}{2\mu_2-\eta}\Big(\frac{e^{(2\mu_2-\eta)(T-t)}-e^{(\mu_2-\phi)(T-t)}}{\mu_2+\phi-\eta}-\frac{e^{(\mu_2-\phi)(T-t)}-1}{\mu_2-\phi}\Big) \\ & + \frac{2ce^{\beta T}}{2\mu_2-\eta+\beta}\Big(\frac{e^{(\mu_2-\phi)(T-t)}-e^{-\beta(T-t)}}{\mu_2-\phi+\beta}-\frac{e^{(2\mu_2-\eta)(T-t)}-e^{(\mu_2-\phi)(T-t)}}{\mu_2+\phi-\eta}\Big). \end{aligned} \tag{4.19}$$

Proof. As β and c are fixed parameters, in this proof, we omit the dependence of B and D on them for notational simplicity.

The quadratic source term $\big(w-e^{\beta t}\hat{w}\big)^2$ in Theorem (4.10) suggests the following *ansatz* for the value function V in Theorem 4.10 of the form

$$V(t,w,\hat{w}) = A(t)w^2+B(t)w+C(t)+\hat{A}(t)\hat{w}^2+\hat{B}(t)\hat{w}+D(t)w\hat{w}, \tag{4.20}$$

where $A, B, C, \hat{A}, \hat{B}, D$ are unknown deterministic functions of time t. If (4.20) is correct, then the pointwise infimum in (4.10) is attained by p^* satisfying the relationship

$$\left(w\cdot\frac{\partial^2 V}{\partial w^2}\right)\cdot p^* = -\frac{1}{\gamma}\left(\big(\mu_1-\mu_2\big)\cdot\frac{\partial V}{\partial w}+\big(\hat{\varrho}\gamma+\theta\big)\cdot\hat{w}\cdot\frac{\partial^2 V}{\partial w\hat{w}}+\theta\cdot w\cdot\frac{\partial^2 V}{\partial w^2}\right), \tag{4.21}$$

assuming $A(t) > 0$. Here γ and θ are defined in (4.15). (4.20) implies that the relevant partial derivatives of V are of the form

$$\frac{\partial^2 V}{\partial w^2} = 2A(t), \quad \frac{\partial V}{\partial w} = 2A(t)w+B(t)+D(t)\hat{w}, \quad \frac{\partial^2 V}{\partial w\hat{w}} = D(t). \tag{4.22}$$

Substituting (4.22) into (4.21), the optimal control p^* obtained is in the form of (4.16), where h and g are given by (4.17). Then, it only remains to determine the functions A, B, D. Substituting (4.21) into PIDE (4.10), we can obtain the following ordinary differential equations (ODE) for A, B, D,

$$\begin{cases} \frac{dA(t)}{dt} = -\Big(2\mu_2 - \eta\Big)A(t) - 1, & A(T) = 0, \\ \frac{dD(t)}{dt} = -\Big(2\mu_2 - \eta\Big)D(t) + 2e^{\beta t}, & D(T) = 0, \\ \frac{dB(t)}{dt} = -(\mu_2 - \phi)B(t) - 2cA(t) - cD(t), & B(T) = 0, \end{cases} \tag{4.23}$$

Solving the ODE system gives us the A, B, D defined in (4.18) and (4.19). We also note that $A(t) > 0$, thus completing the proof.[6] □

4.3.1 Insights from CD-optimal control

The CD-optimal control (4.16) provides insights into the behavior of the optimal allocation policy. For ease of exposition, we first establish the following properties of $g(t;\beta)$ and $h(t;\beta,c)$, defined in (4.17).

Corollary 4.3.1. *(Properties of $g(t;\beta)$). The function $g(t;\beta)$ defined in (4.17) has the following properties for $t \in [t_0, T]$ and $\beta > 0$:*

(i) For fixed $t \in [t_0, T]$, $g(t;\beta)$ is strictly increasing on $\beta \in (0, \infty)$.

(ii) For fixed $\beta > 0$, $g(t;\beta)$ is strictly increasing on $t \in [t_0, T]$.

(iii) $g(t;\beta)$ admits the following bounds:

$$e^{\beta t} \leq g(t;\beta) \leq e^{\beta T}. \tag{4.24}$$

Corollary 4.3.2. *(Properties of $h(t;\beta,c)$). The function $h(t;\beta,c)$ defined in (4.17) has the following properties for $t \in [t_0, T]$, $\beta > 0$ and $c \geq 0$:*

(i) For fixed $t \in [t_0, T]$ and $c > 0$, $h(t;\beta,c)$ is strictly increasing on $\beta \in (0, \infty)$.

(ii) $h(t;\beta,c) \geq 0$, $\forall (t,\beta,c) \in [t_0, T] \times (0,\infty) \times [0,\infty)$.

(iii) For fixed $t \in [t_0, T]$ and $\beta > 0$, $h(t;\beta,c)$ is strictly increasing on $c \in [0,\infty)$. $h(t;\beta,0) \equiv 0$. Moreover, $h(t;\beta,c) \propto c$, i.e. $h(t;\beta,c)$ is proportional to c.

[6] The closed-form solution was verified using Maple software.

Proof. van Staden et al. (2023a) derive the CD-optimal control under the assumption that the stock price follows the double-exponential jump-diffusion model and the bond is risk-free with the bond price $B(t)$ following

$$\frac{dB(t)}{B(t)} = r. \tag{4.25}$$

Under such assumptions, van Staden et al. (2023a) shows that the CD-optimal control can be expressed in a similar form as in (4.16) with g and h functions. The g and h functions satisfy the same properties as in Corollary (4.3.1) and (4.3.2). Despite we assume the bond price follows a jump-diffusion model, the proof of Corollary (4.3.1) and (4.3.2) follows similar steps as the proof in van Staden et al. (2023a). □

In order to analyze the closed-form solution, we make the following assumptions.

Assumption 4.3.4. *(Drift rates of the two assets). We assume that the drift rates of the stock and the bond index μ_1 and μ_2 satisfy the following properties,*

$$\mu_1 - \mu_2 > 0, \quad \mu_1 - \mu_2 + \vartheta > 0, \tag{4.26}$$

where ϑ is defined in (4.15).

Remark 4.3.3. (Remark on drift rate assumptions). The first inequality $\mu_1 - \mu_2 > 0$ indicates that the stock index has a higher drift rate than the bond index, which is a standard assumption. The second inequality $\mu_1 - \mu_2 + \vartheta > 0$ is also practically reasonable since ϑ is a variance term that is usually on a smaller scale compared to the drift rates. In reality, it is unlikely that $\mu_1 - \mu_2 > 0$ but $\mu_1 - \mu_2 + \vartheta \leq 0$.

We now proceed to summarize the insights from the CD-optimal control (4.16). The first obvious observation is that the CD-optimal control is a contrarian strategy. This can be seen from the fact that fixing time and the wealth of the benchmark portfolio $\hat{W}(t)$, the allocation to the more risky stock index decreases when the wealth of the active portfolio $W^*(t)$ increases.

If we take a deeper look at (4.16), we can see that the CD-optimal control consists of two components: a cash injection component p^*_{cash} and a tracking component p^*_{track}. Mathematically,

$$p^*(t, w, \hat{w}, \hat{\varrho}) = p^*_{cash}(t, w, \hat{w}) + p^*_{track}(t, w, \hat{w}, \hat{\varrho}), \tag{4.27}$$

where

$$\begin{cases} p^*_{cash}(t, w, \hat{w}) = \frac{1}{W^*(t)} \left[\frac{(\mu_1 - \mu_2)}{\gamma} h(t; \beta, c) \right], \\ p^*_{track}(t, w, \hat{w}, \hat{\varrho}) = \frac{1}{W^*(t)} \left[\frac{(\mu_1 - \mu_2 + \vartheta)}{\gamma} \Big(g(t; \beta) \hat{W}(t) - W^*(t) \Big) + g(t; \beta) \hat{W}(t) \cdot \hat{\varrho} \right]. \end{cases} \tag{4.28}$$

Based on Assumption 4.3.4 and Corollary 4.3.2, the cash injection component p^*_{cash} is always non-negative. Furthermore, from Corollary 4.3.2, we know that the stock allocation from the cash injection component is proportional to the cash injection rate c. In addition, as $t \uparrow T$, $h(t;\beta,c)$ increases, and thus the stock allocation from the cash injection component also increases with time.

On the other hand, the tracking component p_{track} does not depend on the cash injection rate c, but only concerns the tracking performance of the active portfolio. One key finding is that

$$\begin{cases} p^*_{track}(t,w,\hat{w},\hat{\varrho}) \geq \hat{\varrho}, & \text{if } W^*(t) \leq g(t;\beta)\hat{W}(t), \\ p^*_{track}(t,w,\hat{w},\hat{\varrho}) < \hat{\varrho}, & \text{if } W^*(t) > g(t;\beta)\hat{W}(t). \end{cases} \tag{4.29}$$

This means that the CD-optimal control uses $g(t;\beta)\hat{W}(t)$ as the true target for the active portfolio to decide if it should take more or less risk than the benchmark portfolio. This is a key observation, since the CD objective function (4.6) measures the difference between $W(t)$ and $e^{\beta t}\hat{W}(t)$. One would naively think that the optimal strategy would be based on the deviation from $e^{\beta t}\hat{W}(t)$. In contrast, from Corollary 4.3.1, we know that the true target $g(t;\beta)\hat{W}(t)$ used for decision making is greater than $e^{\beta t}\hat{W}(t)$. The insight from this observation is that if the manager wants to track an elevated target $e^{\beta t}\hat{W}(t)$, she should aim higher than the target itself.

4.4 Approximate form under realistic constraints

The closed-form solution p^* in (4.16) is obtained under several unrealistic assumptions, namely continuous rebalancing, unlimited leverage, and continuing trading in insolvency.[7] In practice, investors are subject to constraints such as discrete rebalancing, limited leverage, and no trading when insolvent. Therefore, obtaining an approximation of the closed-form solution that satisfies realistic constraints is of practical importance.

In particular, we consider an equally-spaced discrete rebalancing schedule $\mathcal{T}_{\Delta t}$ defined as

$$\mathcal{T}_{\Delta t} = \Big\{ t_i : \ i = 0, \cdots, N \Big\}, \tag{4.30}$$

where $t_i = i\Delta t$, and $\Delta t = T/N$. Then, the *(discrete) clipped form* $\bar{p}_{\Delta t} : \mathcal{T}_{\Delta t} \times \mathbb{R}^3 \mapsto \mathbb{R}$ is defined as

$$\bar{p}_{\Delta t}(t_i, \bar{W}_{\Delta t}(t_i), \hat{W}_{\Delta t}(t_i), \hat{\varrho}) = \min\left(\max\Big(p^*(t_i, \bar{W}_{\Delta t}(t_i), \hat{W}_{\Delta t}(t_i), \hat{\varrho}), p_{min}\Big), p_{max} \right). \tag{4.31}$$

Here $[p_{min}, p_{max}]$, where $p_{min} = 0$ and $p_{max} \geq 1$, is the allowed range of stock allocation fraction, $\bar{W}_{\Delta t}(t_i)$ is the wealth of the active portfolio at t_i following $\bar{p}_{\Delta t}$ from t_0 to t_i, $\hat{W}_{\Delta t}(t_i)$

[7] Note that we consider a two-asset scenario here, thus the scalar $p^* \in \mathbb{R}$ (allocation fraction for the stock index) fully describes the allocation strategy $\boldsymbol{p}^*$, since $\boldsymbol{p}^* = (p^*, 1-p^*)^\top$.

is the wealth of the benchmark portfolio at t_i following the fixed-mix strategy described by constant allocation fraction $\hat{\varrho}$, but only rebalanced discretely according to $\mathcal{T}_{\Delta t}$. Clearly, the allocation strategy from $\bar{p}_{\Delta t}$ follows the discrete schedule of $\mathcal{T}_{\Delta t}$, and satisfies the leverage constraint that $\bar{p}_{\Delta t} \in [p_{min}, p_{max}]$. $p_{\Delta t}$ approaches the closed-form solution p^* as $\Delta t \downarrow 0, p_{min} \downarrow -\infty$ and $p_{max} \uparrow \infty$. We note that a similar clipping idea is explored in Vigna (2014) in the context of closed-form solutions for multi-period mean-variance asset allocation. However, it should be emphasized that the clipped form $\bar{p}_{\Delta t}$ with finite (p_{min}, p_{max}) is a feasible, but in general sub-optimal, control of the leverage-constrained CD problem (4.7).

We then address the assumption that trading continues when insolvent, i.e., when the wealth of the portfolio reaches zero. While necessary for the mathematical derivation of the closed-form solution, we acknowledge that this is by no means reasonable for practitioners. Under the continuous rebalancing case (assuming no jumps), if the control (allocation) is bounded, it can be shown that the wealth of the portfolio can never be negative (Wang and Forsyth, 2012). However, with discrete rebalancing, even with a bounded control, as long as the upper bound $p_{max} > 1$, it is theoretically possible that the portfolio value becomes negative. We address this assumption by applying an overlay on strategies so that in the case of insolvency, we assume the manager liquidates the long-only positions and allocates the debt (negative wealth) to a shortable asset (bond) to allow outstanding debt to accumulate until the end of the investment horizon.

In summary, the clipped form satisfies the realistic constraints and serves as a constraint-compliant alternative solution for practitioners.

4.5 Conclusions

In conclusion, in this chapter, we have discussed the following.

(i) We derived the closed-form solution to the problem of outperforming a passive benchmark portfolio under the CD objective function, under double-exponential jump-diffusion asset price models, and assumptions of continuous rebalancing, unlimited leverage, and continuation of trading when insolvent.

(ii) We showed that the optimal strategy is a contrarian strategy, which takes more risk when the strategy is underperforming the benchmark. In addition, the optimal strategy aims for a target higher than the elevated target used in the objective function.

(iii) We proposed an approximate form of the closed-form solution that satisfies realistic constraints and can be used as an alternative solution for practitioners.

Chapter 5

Neural network for portfolio optimization with leverage constraints

5.1 Introduction

In Chapter 4, we derived the closed-form solution to the multi-period portfolio optimization problem for outperforming a benchmark portfolio throughout the investment horizon. While the closed-form solution provides valuable intuition into understanding the behavior of the optimal control, it requires unrealistic assumptions such as unlimited leverage and continuation of trading when insolvent.

However, in practical investment portfolios, it is common to allow some leverage to increase exposure to specific risk factors in the financial market. Consequently, the multi-period portfolio optimization problem is subject to a bounded leverage constraint (due to risk mandates), introducing additional complexity that existing methods for multi-period portfolio optimization struggle to handle.

In this chapter, we extend the approach of approximating the optimal control with a neural network model and propose a novel methodology for solving multi-period portfolio optimization problems with bounded leverage constraints.

We begin by discussing the practical significance and motivation for considering leverage constraints in investment portfolios. Subsequently, we mathematically formulate the leverage constraints.

To tackle these complex leverage constraints and make the constrained optimization problem computationally feasible, we propose a leverage-feasible neural network (LFNN) model. The LFNN model converts the original constrained optimization problem into an unconstrained one, enabling effective calculation of the optimal controls through standard optimization methods.

Furthermore, we provide mathematical analysis demonstrating the LFNN's capability to approximate the optimal control with arbitrary precision, guaranteeing that solving the

parameterized unconstrained optimization problem can yield accurate approximations to the solution of the original complex optimization problem with leverage constraints.

Finally, to validate the LFNN model's performance, we present a numerical experiment comparing the learned neural network strategy with the clipped form of the closed-form solution. The results demonstrate that the LFNN model is capable of learning a comparable solution to the clipped form, further affirming its effectiveness in handling the leverage constraints in the portfolio optimization problem.

In summary, we make the following contributions in this chapter.

(i) We propose a novel approach to tackle the stochastic optimal control problem under realistic constraints, such as discrete rebalancing and limited leverage, by directly representing the control using a neural network. Specifically, we introduce the leverage-feasible neural network (LFNN) model, which effectively converts the original complex leverage-constrained optimization problem into an unconstrained one, allowing for a straightforward solution using standard optimization methods.

(ii) We prove that, with a suitable choice of the hyperparameter of the LFNN model, the solution of the parameterized unconstrained optimization problem can approximate the optimal control arbitrarily well. This provides a mathematical justification for the validity of the LFNN approach.

(iii) Through numerical experiments, we demonstrate that the learned neural network strategy achieves comparable performance relative to the clipped form of the closed-form solution on synthetic data. This demonstrates the LFNN model's capability to find optimal solutions efficiently and provides empirical support for the mathematical approximation theorem.

5.2 Portfolio optimization with leverage constraints

Large pension funds such as the Canada Pension Plan often have exposures to alternative assets, such as private equity (CPP Investments, 2022). Literature suggests that returns on private equity can be replicated using a leveraged small-cap stock index (Phalippou, 2014; L'Her et al., 2016). Following this line of argument, we allow managers to take leverage to invest in public stock index funds to roughly mimic the pension fund portfolios with some exposure to private equity. Typically, due to risk mandates, the total leverage in the portfolio is subject to an upper bound.

Essentially, taking leverage to invest in stocks requires borrowing additional capital. For simplicity, we consider a self-financing portfolio and assume the borrowing activity is represented by shorting some bond assets within the portfolio.

In this section, we mathematically formulate the leverage constraints. Following the notations from Section 4.2, we assume that the total N_a underlying assets are divided into

two groups. The first group of N_L assets are long-only assets, which we index by the set $\{1, \cdots, N_L\}$. The second group of $N_a - N_L$ assets are shortable assets that can be shorted to create leverage and are indexed by the set $\{N_L + 1, \cdots, N_a\}$. Recall the notation of $p_i(\boldsymbol{X}(t))$ for the allocation fraction for asset i at time t, as a function of the state variable vector $\boldsymbol{X}(t)$.

For long-only assets, the wealth fraction needs to be non-negative, hence we have

$$\text{(Long-only constraint):} \quad p_i(\boldsymbol{X}(t)) \geq 0,\ i \in \{1, \cdots, N_L\},\ t \in \mathcal{T}. \tag{5.1}$$

Furthermore, the total allocation fraction for all assets should be one. Therefore, the following summation constraint needs to be satisfied

$$\text{(Summation constraint):} \quad \sum_{i=1}^{N_a} p_i(\boldsymbol{X}(t)) = 1,\ t \in \mathcal{T}. \tag{5.2}$$

In practice, due to borrowing costs (from taking leverage) and risk management mandates, the level of leverage is often constrained. For this reason, we cap the maximum leverage by introducing a constant p_{max}, which represents the total allocation fraction for long-only assets. Therefore,

$$\text{(Maximum leverage constraint):} \quad \sum_{i=1}^{N_L} p_i(\boldsymbol{X}(t)) \leq p_{max},\ t \in \mathcal{T}. \tag{5.3}$$

Note that no leverage is permitted if $p_{max} = 1$.

Finally, we make the following assumption on the scenario of shorting multiple shortable assets.

Assumption 5.2.1. *(Simultaneous shorting). If one shortable asset has a negative weight, other shortable assets must have nonpositive weights. Mathematically, this assumption can be expressed as*

$$\text{(Simultaneous shorting):} \begin{cases} p_i(\boldsymbol{X}(t)) \leq 0,\ \forall i \in \{N_L + 1, \cdots, N_a\},\ \text{if } \sum_{i=1}^{N_L} p_i(\boldsymbol{X}(t)) > 1,\ t \in \mathcal{T} \\ p_i(\boldsymbol{X}(t)) \geq 0,\ \forall i \in \{N_L + 1, \cdots, N_a\},\ \text{if } \sum_{i=1}^{N_L} p_i(\boldsymbol{X}(t)) \leq 1,\ t \in \mathcal{T} \end{cases}. \tag{5.4}$$

Remark 5.2.1. (Remark on Assumption 5.2.1). This assumption avoids the ambiguity between long-only assets and shortable assets in scenarios that involve leverage. When leveraging occurs, all shortable assets are treated as one group to provide the needed liquidity to achieve the desired leverage level. A typical scenario would involve long-only equities and long term bonds, while allowing shorting of short term T-bills, i.e. effectively borrowing cash. This scenario falls under Assumption 5.2.1. We will discuss alternative assumptions concerning leverage and shorting in Chapter 7.

The above constraints permit scenarios with non-negative portfolio wealth. Before we proceed to the handling of the negative portfolio wealth scenarios, we first define the following partition of the state space $\mathcal{X}$,

Definition 5.2.1. *(Partition of state space). We define $\{\mathcal{X}_1, \mathcal{X}_2\}$ to be a partition of the state space $\mathcal{X}$, such that*

$$\begin{cases} \mathcal{X}_1 = \left\{ x = (t, W, \hat{W})^\top \in \mathcal{X} \middle| W \geq 0 \right\}, \\ \mathcal{X}_2 = \left\{ x = (t, W, \hat{W})^\top \in \mathcal{X} \middle| W < 0 \right\}. \end{cases} \tag{5.5}$$

Intuitively, we separate the state space $\mathcal{X}$ into two regions by the wealth of the active portfolio, one with non-negative wealth and the other with negative wealth. Since we allow shorting, and the portfolio is discretely rebalanced, it is possible for the investor to become insolvent. Then, we present the following assumption concerning the negative wealth (insolvency) scenarios.

Assumption 5.2.2. *(No trading in insolvency). If the wealth of the active portfolio is negative, then all long-only asset positions should be liquidated, and all the debt (i.e. the negative wealth) should be allocated to the least-risky shortable asset (in terms of volatility). In particular, without loss of generality, we assume all debt is allocated to the shortable asset indexed with $N_L + 1$. Let $\boldsymbol{e}_i \in \mathbb{R}^{N_a} = (0, \cdots, 0, 1, 0, \cdots, 0)^\top$ denote the standard basis vector of which the i-th entry is 1 and all other entries are 0. Then, we can formulate this assumption as follows.*

$$\text{(No trading in insolvency):} \quad p(\boldsymbol{X}(t)) = \boldsymbol{e}_{N_L+1}, \quad \text{if } \boldsymbol{X}(t) \in \mathcal{X}_2. \tag{5.6}$$

Remark 5.2.2. (Remark on Assumption 5.2.2). Essentially, when the portfolio wealth is negative, we assume the debt is allocated to a short-term bond asset and accumulates over time.

Summarizing the constraints, we can define two sets $\mathcal{Z}_1, \mathcal{Z}_2$:

$$\begin{cases} \mathcal{Z}_1 = \left\{ \boldsymbol{z} \in \mathbb{R}^{N_a} \middle| \begin{cases} z_i \geq 0, \forall i \in \{1, \cdots, N_L\}, \\ \sum_{i=1}^{N_a} z_i = 1, \\ \sum_{i=1}^{N_L} z_i \leq p_{max}, \\ z_i \leq 0, \ \forall i \in \{N_L + 1, \cdots, N_a\}, \text{ if } \sum_{i=1}^{N_L} z_i > 1, \\ z_i \geq 0, \ \forall i \in \{N_L + 1, \cdots, N_a\}, \text{ if } \sum_{i=1}^{N_L} z_i \leq 1 \end{cases} \right\}, & (5.7) \\ \mathcal{Z}_2 = \left\{ \boldsymbol{e}_{N_L+1} \right\}, & (5.8) \end{cases}$$

Then, the corresponding space of feasible control vector values $\mathcal{Z}$ and the admissible strategy set $\mathcal{A}$ can be expressed as follows.

$$\text{(Admissible set)}: \quad \begin{cases} \mathcal{Z} = \mathcal{Z}_1 \cup \mathcal{Z}_2, & (5.9) \\ \mathcal{A} = \left\{ p \,\middle|\, \begin{cases} p(\boldsymbol{X}(t)) \in \mathcal{Z}_1, \text{ if } \boldsymbol{X}(t) \in \mathcal{X}_1, \\ p(\boldsymbol{X}(t)) \in \mathcal{Z}_2, \text{ if } \boldsymbol{X}(t) \in \mathcal{X}_2, \end{cases} \forall t \in \mathcal{T} \right\}. & (5.10) \end{cases}$$

Consequently, we need to solve the following challenging constrained optimization problem.

$$\text{(Constrained optimization problem)}: \quad \inf_{p \in \mathcal{A}} \mathbb{E}_p^{(t_0, w_0)} \big[F(\mathcal{W}_p, \hat{\mathcal{W}}_{\hat{p}}) \big]. \tag{5.11}$$

5.3 Leverage-feasible neural network

In Section 4.3, we derived the closed-form solution under the jump-diffusion model, which requires several unrealistic assumptions such as continuous rebalancing, unlimited leverage, and trading in insolvency. Furthermore, the closed-form solution is specific to the investment objective defined in the CD problem (4.6). It is thus critically beneficial to develop a technique to discover optimal solutions to the constrained investment problem (5.11) for different objectives and under realistic constraints, such as discrete rebalancing and the leverage constraints discussed in Section 5.2.

Specifically, the numerical solution to the general problem (5.11) requires solving for the feedback control p. In this chapter, we follow the high-level idea discussed in Chapter 2, and approximate the control function $p(\cdot)$ with a neural network function $f_{\boldsymbol{\theta}} : \mathcal{X} \mapsto \mathbb{R}^{N_a}$, where $\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}$ represents the parameters of the neural network (i.e., weights and biases). In other words,

$$p(\boldsymbol{X}(t)) \simeq f_{\boldsymbol{\theta}}(\boldsymbol{X}(t)). \tag{5.12}$$

Then, the optimization problem (5.11) can be converted to solving the following parameterized optimization problem.

$$\text{(Parameterized optimization problem)}: \quad \inf_{\boldsymbol{\theta} \in \mathcal{Z}_{\boldsymbol{\theta}}} \mathbb{E}_{f_{\boldsymbol{\theta}}}^{(t_0, w_0)} \big[F(\mathcal{W}_{\boldsymbol{\theta}}, \hat{\mathcal{W}}_{\hat{p}}) \big]. \tag{5.13}$$

Here $\mathcal{W}_{\boldsymbol{\theta}}$ is the wealth trajectory of the active portfolio with control following the neural network approximation function parameterized by θ. $\mathcal{Z}_{\boldsymbol{\theta}} \subseteq \mathbb{R}^{N_{\boldsymbol{\theta}}}$ is the feasibility domain of the parameter $\boldsymbol{\theta}$, which is translated from the constraints of the original problem, i.e., (5.9) and (5.10). Mathematically,

$$\mathcal{Z}_{\boldsymbol{\theta}} = \left\{ \boldsymbol{\theta} : \begin{cases} f_{\boldsymbol{\theta}}(\boldsymbol{X}) \in \mathcal{Z}_1, \text{ if } \boldsymbol{X} \in \mathcal{X}_1, \\ f_{\boldsymbol{\theta}}(\boldsymbol{X}) \in \mathcal{Z}_2, \text{ if } \boldsymbol{X} \in \mathcal{X}_2. \end{cases} \right\}. \tag{5.14}$$

Here $\mathcal{Z}_1, \mathcal{Z}_2$ are defined in (5.7), (5.8) and $\mathcal{X}_1, \mathcal{X}_2$ are partitions of the state space $\mathcal{X}$ defined in Definition 5.2.1.

Note here that $\mathcal{Z}_{\boldsymbol{\theta}}$ depends on the structure of the neural network function $f_{\boldsymbol{\theta}}$. Intuitively, $\mathcal{Z}_{\boldsymbol{\theta}}$ can be thought of the preimage of $\mathcal{Z}$ under the neural network function. We are interested in the neural network model design that results in $\mathcal{Z}_{\boldsymbol{\theta}} = \mathbb{R}^{N_{\boldsymbol{\theta}}}$, which means problem (5.13) becomes an unconstrained optimization problem. For long-only investment problems, the only constraints are the long-only constraint (5.1) and the summation constraint (5.2). Previous work has proposed a neural network architecture with a softmax activation function at the last layer so that the output (vector of allocation fractions) automatically satisfies the two constraints, and thus $\mathcal{Z}_{\boldsymbol{\theta}} = \mathbb{R}^{N_{\boldsymbol{\theta}}}$ and problem (5.13) becomes an unconstrained optimization problem (see, e.g., Li and Forsyth (2019); Ni et al. (2022)). However, as discussed in Section 5.2, we consider the more complicated case where leverage and shorting are allowed. The problem thus involves more constraints than the long-only case and therefore requires us to design a new model architecture to convert the constrained optimization problem to an unconstrained problem. We will discuss the design of the *leverage-feasible neural network* (LFNN) model in the next section, and how the LFNN model achieves this goal.

5.3.1 LFNN model

In this section, we propose the leverage-feasible neural network (LFNN) model, which yields $\mathcal{Z}_{\boldsymbol{\theta}} = \mathbb{R}^{N_{\boldsymbol{\theta}}}$ for leverage constraints defined in equation (5.9), and therefore converts a constrained optimization problem (5.13) to an unconstrained problem.

We first define the commonly used fully connected feedforward neural network (Lu and Lu, 2020) as follows.

Definition 5.3.1. (Fully connected feedforward neural network $\tilde{f}_{\boldsymbol{\theta}}$). *A fully connected feedforward neural network (FNN) maps an input vector $\boldsymbol{x} \in \mathbb{R}^{d_0}$ to an output vector $\boldsymbol{h} \in \mathbb{R}^{d_{K+1}}$, and contains K hidden layers of sizes $d_1, \cdots, d_K$. The neural network is parameterized by the weight matrices $\boldsymbol{\theta}^{(k)} \in \mathbb{R}^{d_{k-1} \times d_k}$ and bias vectors $\boldsymbol{\theta}_b^{(k)} \in \mathbb{R}^{d_k}$, for $k = 1, \cdots, K+1$. Then, the output $\boldsymbol{h}$ is derived from the input $\boldsymbol{x}$ iteratively as follows.*

$$\begin{cases} \boldsymbol{x}^{(0)} = \boldsymbol{x}, \\ \boldsymbol{x}^{(k)} = \sigma\Big(\big(\boldsymbol{\theta}^{(k)}\big)^\top \cdot \boldsymbol{x}^{(k-1)} + \boldsymbol{\theta}_b^{(k)}\Big), 1 \le k \le K, \\ \boldsymbol{h} = \big(\boldsymbol{\theta}^{(K+1)}\big)^\top \cdot \boldsymbol{x}^{(K)} + \boldsymbol{\theta}_b^{(K+1)}. \end{cases} \tag{5.15}$$

Here σ is the pointwise sigmoid activation function, i.e. for any vector $\boldsymbol{z}$, $[\sigma(\boldsymbol{z})]_i = \sigma(\boldsymbol{z}_i)$. For notational simplicity, we flatten and assembly all weight matrices and bias vectors into a single parameter vector $\boldsymbol{\theta} = (\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}_b^{(1)}, \cdots, \boldsymbol{\theta}^{(K+1)}, \boldsymbol{\theta}_b^{(K+1)})^\top \in \mathbb{R}^{N_{\boldsymbol{\theta}}}$, where $N_{\boldsymbol{\theta}} = \sum_{k=1}^{K+1}(d_{k-1} \cdot d_k + d_k)$. Furthermore, we use the 2-tuple $\Big(K, (d_1, \cdots, d_K)^\top\Big)$ to denote the hyperparameters, i.e. the number of hidden layers and the sizes of each hidden layer.

The function defined by the above fully connected feedforward neural network parameterized by $\boldsymbol{\theta}$ is denoted by $\tilde{f}_{\boldsymbol{\theta}}$.

Note that the size of $\boldsymbol{\theta}$ depends on hyperparameters $\Big(K, (d_1, \cdots, d_K)^\top\Big)$. However, for notational simplicity, we omit the 2-tuple in $\tilde{f}_{\boldsymbol{\theta}}$.

Building on $\tilde{f}_{\boldsymbol{\theta}}$, we propose the following *leverage-feasible neural network* (LFNN) model $f_{\boldsymbol{\theta}} : \mathcal{X} \mapsto \mathcal{Z}$:

$$\text{(LFNN)}: \quad f_{\boldsymbol{\theta}}(\boldsymbol{x}) := \psi\Big(\tilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{x}\Big) \in \mathcal{Z}. \tag{5.16}$$

Here, $\psi(\cdot)$ is the *leverage-feasible activation function*. For $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{o} = (o_1, \cdots, o_{N_a+1})^\top \in \mathbb{R}^{N_a+1}$, let $\boldsymbol{p} = (p_1, \cdots, p_{N_a}) = \psi(\boldsymbol{o}, \boldsymbol{x})$, where $\psi(\cdot) : \mathbb{R}^{N_a+1} \times \mathcal{X} \mapsto \mathcal{Z}$ is defined as follows.

$$\boldsymbol{p} = \psi(\boldsymbol{o}, \boldsymbol{x}) = \begin{cases} \begin{cases} l = p_{max} \cdot \sigma(o_{N_a+1}), \\ p_i = l \cdot \frac{e^{o_i}}{\sum_{k=1}^{N_L} e^{o_k}},\ i \in \{1, \cdots, N_L\}, & \text{if } \boldsymbol{x} \in \mathcal{X}_1, \\ p_i = (1-l) \cdot \frac{e^{o_i}}{\sum_{k=N_L+1}^{N_a} e^{o_k}},\ i \in \{N_L+1, \cdots, N_a\}, \end{cases} \\ \boldsymbol{e}_{N_L+1}, & \text{if } \boldsymbol{x} \in \mathcal{X}_2. \end{cases} \tag{5.17}$$

Here σ is the sigmoid function, N_L is the number of long-only assets, $\boldsymbol{e}_{N_L+1}$ is the standard basis vector with the (N_L+1)-th entry being 1 and other entries being 0, and p_{max} is the maximum leverage allowed. We show that the leverage-feasible activation function ψ has the following property.

Lemma 5.3.1. *(Decomposition of ψ). The leverage-feasible function ψ defined in (5.17) has the function decomposition that*

$$\psi(\boldsymbol{o}, \boldsymbol{x}) = \varphi(\zeta(\boldsymbol{o}), \boldsymbol{x}), \tag{5.18}$$

where

$$\begin{cases} \zeta : \mathbb{R}^{N_a+1} \mapsto \tilde{\mathcal{Z}}, \zeta(\boldsymbol{o}) = \Bigg(Softmax\Big((o_1, \cdots, o_{N_L})\Big), Softmax\Big((o_{N_L+1}, \cdots, o_{N_a})\Big), p_{max} \cdot \sigma(o_{N_a+1})\Bigg)^\top, \\ \varphi : \tilde{\mathcal{Z}} \times \mathcal{X} \mapsto \mathcal{Z}, \varphi(\boldsymbol{z}, \boldsymbol{x}) = \Big(z_{N_a+1} \cdot (z_1, \cdots, z_{N_L}), (1 - z_{N_a+1}) \cdot (z_{N_L+1}, \cdots, z_{N_a})\Big)^\top \cdot \boldsymbol{1}_{\boldsymbol{x} \in \mathcal{X}_1} \\ \qquad\qquad + \boldsymbol{e}_{N_L+1} \cdot \boldsymbol{1}_{\boldsymbol{x} \in \mathcal{X}_2}, \end{cases} \tag{5.19}$$

and

$$\tilde{\mathcal{Z}} = \Bigg\{ \boldsymbol{z} = (z_1, \cdots, z_{N_a+1})^\top \in \mathbb{R}^{N_a+1}, \sum_{i=1}^{N_L} z_i = 1, \sum_{i=N_L+1}^{N_a} z_i = 1, z_{N_a+1} \le p_{max}, z_i \ge 0, \forall i \Bigg\}. \tag{5.20}$$

Proof. This is easily verifiable by the definition of ψ in (5.17). □

Remark 5.3.1. (Remark on Lemma 5.3.1). The leverage-feasible activation function ψ corresponds to a two-step decision process described by ζ and φ. Intuitively, ζ first determines the internal allocations within long-only assets and shortable assets, as well as the total leverage. Then, φ converts the internal allocations and total leverage into final allocation fractions, which depend on the wealth of the active portfolio.

With the LFNN model outlined above, we can show that the parameterized optimization problem (5.13) becomes an unconstrained optimization problem. Specifically, we present the following theorem regarding the feasibility domain $\mathcal{Z}_{\boldsymbol{\theta}}$ associated with the LFNN model (5.16).

Theorem 5.3.1. *(Unconstrained feasibility domain). The feasibility domain $\mathcal{Z}_{\boldsymbol{\theta}}$ defined in (5.14) associated with the LFNN model (5.16) is $\mathbb{R}^{N_{\boldsymbol{\theta}}}$.*

Proof. First, it is obvious that $\mathcal{Z}_{\boldsymbol{\theta}} \subseteq \mathbb{R}^{N_{\boldsymbol{\theta}}}$ by definition of (5.14). Next, we show that $\mathbb{R}^{N_{\boldsymbol{\theta}}} \subseteq \mathcal{Z}_{\boldsymbol{\theta}}$. To prove this, we need to show that for any $\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}$,

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{p} \in \begin{cases} \mathcal{Z}_1, \text{ if } \boldsymbol{x} \in \mathcal{X}_1, \\ \mathcal{Z}_2, \text{ if } \boldsymbol{x} \in \mathcal{X}_2, \end{cases} \quad \forall \boldsymbol{x} \in \mathcal{X}. \tag{5.21}$$

Here, $f_{\boldsymbol{\theta}}$ is the LFNN function defined in (5.16), $\boldsymbol{p} = (p_1, \cdots, p_{N_a})^\top \in \mathbb{R}^{N_a}$ is the output of the LFNN model that represents the wealth allocation to the assets, $\mathcal{Z}_1$ and $\mathcal{Z}_2$ are the feasibility sets defined in (5.7) and (5.8), and $\boldsymbol{x} \in \mathcal{X}$ is a feature vector. To prove (5.21), we verify the two scenarios for $\boldsymbol{x} \in \mathcal{X}_1$ and $\boldsymbol{x} \in \mathcal{X}_2$ respectively.

When $\boldsymbol{x} \in \mathcal{X}_2$, it is easily verifiable that $\boldsymbol{p} = \boldsymbol{e}_{N_L+1}$ via the definition of the leverage-feasible activation function (5.17).

Next, we verify that when $\boldsymbol{x} \in \mathcal{X}_1$, $\boldsymbol{p} \in \mathcal{Z}_1$. To prove this, we need to show that constraints of (5.1)-(5.4) are satisfied when $\boldsymbol{x} \in \mathcal{X}_1$.

We borrow the variables l and o_{N_a+1} in equation (5.17).

By definition of (5.17), it is obvious that the long-only constraint (5.1) holds for long-only assets.

It is also easy to verify that the summation constraint (5.2) is satisfied. This can be observed after the fact that

$$\sum_{i=1}^{N_L} p_i = l, \quad \text{and} \quad \sum_{i=N_L+1}^{N_a} p_i = 1 - l. \tag{5.22}$$

The maximum leverage constraint (5.3) is also satisfied, as

$$\sum_{i=1}^{N_L} p_i = l = p_{max} \cdot \sigma(-o_{N_a+1}) \leq p_{max}. \tag{5.23}$$

Finally, the simultaneous shorting constraint (5.2.1) is satisfied. To see this, we examine the scenario when leverage occurs, i.e., $\sum_{i=1}^{N_L} p_i = l > 1$. Then, by definition from (5.17), we know

$$p_i = (1-l) \cdot \frac{e^{o_i}}{\sum_{k=N+1}^{N_a} e^{o_k}} \leq 0, \ \forall i \in \{N_L + 1, \cdots, N_a\} \tag{5.24}$$

From (5.24) it is clear that if $l \leq 1$, then $p_i \geq 0, \forall i$.

Therefore, for any $\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}$, (5.21) is satisfied. This implies $\mathbb{R}^{N_{\boldsymbol{\theta}}} \subseteq \mathcal{Z}_{\boldsymbol{\theta}}$.

□

Following Theorem 5.3.1, the constrained optimization problem (5.11) can be transformed into the following unconstrained optimization problem

$$\text{(Unconstrained parameterized problem):} \quad \inf_{\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}} \mathbb{E}_{f_{\boldsymbol{\theta}}}^{(t_0, w_0)} \big[F(\mathcal{W}_{\boldsymbol{\theta}}, \hat{\mathcal{W}}_{\hat{p}}) \big]. \tag{5.25}$$

5.3.2 Mathematical justification for LFNN approach

By approximating the feasible control with a parameterized LFNN model, we have shown that the original constrained optimization problem is transformed into an unconstrained optimization problem, which is computationally more feasible.

However, an important question remains: is the solution to the parameterized unconstrained optimization problem (5.25) capable of yielding the optimal control of the original problem (5.11)? In other words, suppose $\boldsymbol{\theta}^*$ is the solution to (5.25), can $f_{\boldsymbol{\theta}^*}$ approximate solution to the original problem (5.11) with high accuracy?

In this section, we prove that under benign assumptions about the optimal control and appropriate choices of the hyperparameters of the LFNN model (5.16), solving the unconstrained problem (5.25) provides an arbitrarily close approximation to solving the original problem (5.11). We start by establishing the following lemma.

Lemma 5.3.2. *(Structure of feasible control). Any feasible control function $p : \mathcal{X} \mapsto \mathcal{Z}$, where $\mathcal{Z}$ is defined in* (5.10)*, has the following function decomposition*

$$p(\boldsymbol{x}) = \varphi(\omega(\boldsymbol{x}), \boldsymbol{x}), \forall \boldsymbol{x} \in \mathcal{X}, \tag{5.26}$$

where $\varphi : \tilde{\mathcal{Z}} \times \mathcal{X} \mapsto \mathcal{Z}$ is defined in (5.19) and $\omega : \mathcal{X} \mapsto \tilde{\mathcal{Z}}$.

Proof. We prove the lemma by existence.

Define ω as

$$\omega(\boldsymbol{x}) = \begin{cases} \phi\big(p(\boldsymbol{x})\big), & \text{if } \boldsymbol{x} \in \mathcal{X}_1, \\ \big(\frac{1}{N_L}, \cdots, \frac{1}{N_L}, \frac{1}{N_a - N_L}, \cdots, \frac{1}{N_a - N_L}, 0\big)^\top, & \text{if } \boldsymbol{x} \in \mathcal{X}_2, \end{cases} \tag{5.27}$$

where for any $\boldsymbol{z} = (z_1, \cdots, z_{N_a})^\top \in \mathcal{Z}_1$ (defined in 5.7), $\boldsymbol{y} = (y_1, \cdots, y_{N_a+1})^\top = \phi(\boldsymbol{z}) \in \mathbb{R}^{N_a+1}$ is defined as

$$\phi(\boldsymbol{z}) \equiv \boldsymbol{y} = \begin{cases} \begin{cases} y_i = \frac{z_i}{\sum_{j=1}^{N_L} z_j},\ i \in \{1, \cdots, N_L\}, \\ y_i = \frac{z_i}{1-\sum_{j=1}^{N_L} z_j},\ i \in \{N_L, \cdots, N_a\}, \\ y_{N_a+1} = \sum_{j=1}^{N_L} z_j, \end{cases} & \text{if } \sum_{i=1}^{N_L} z_i \in (0,1) \cup (1, p_{max}], \\ \begin{cases} y_i = z_i,\ i \in \{1, \cdots, N_L\}, \\ y_i = 1/(N_a - N_L),\ i \in \{N_L, \cdots, N_a\}, \\ y_{N_a+1} = 1, \end{cases} & \text{if } \sum_{i=1}^{N_L} z_i = 1, \\ \begin{cases} y_i = 0,\ i \in \{1, \cdots, N_L\}, \\ y_i = z_i,\ i \in \{N_L, \cdots, N_a\}, \\ y_{N_a+1} = 0, \end{cases} & \text{if } \sum_{i=1}^{N_L} z_i = 0, \end{cases} \tag{5.28}$$

It can then be easily verified that $\omega : \mathcal{X} \mapsto \tilde{\mathcal{Z}}$, and that $p(\boldsymbol{x}) = \varphi(\omega(\boldsymbol{x}), \boldsymbol{x})$.

□

Next, we propose the following benign assumptions on the state space and the optimal control.

Assumption 5.3.1. (Assumption on state space and optimal control).

(i) The space $\mathcal{X}$ of state variables is a compact set.

(ii) Following Lemma 5.3.2, for any $\boldsymbol{x} \in \mathcal{X}$, the optimal control $p^* : \mathcal{X} \mapsto \mathcal{Z}$ has the decomposition $p^*(\boldsymbol{x}) = \varphi(\omega^*(\boldsymbol{x}), \boldsymbol{x})$ for some $\omega^* : \mathcal{X} \mapsto \tilde{\mathcal{Z}}$. We assume $\omega^* \in C(\mathcal{X}, \tilde{\mathcal{Z}})$, where $C(\mathcal{X}, \tilde{\mathcal{Z}})$ denotes the set of continuous mappings from $\mathcal{X}$ to $\tilde{\mathcal{Z}}$.

Remark 5.3.2. (Remark on Assumption 5.3.1). In our particular problem of outperforming a benchmark portfolio, the state variable vector is $(t, W(t), \hat{W}(t))^\top \in \mathcal{X}$ where $t \in [0, T]$. In this case, assumption (i) is equivalent to the assumption that the wealth of the active portfolio and benchmark portfolio is bounded, i.e. $\mathcal{X} = [0, T] \times [w_{min}, w_{max}] \times [w_{min}, \hat{w}_{max}]$, where w_{min}, w_{max} and $\hat{w}_{min}, \hat{w}_{max}$ are the respective wealth bounds for the portfolios. Intuitively, assumption (ii) states that the decision process for the optimal control to obtain the allocation fractions within the long-only assets and the shortable assets, and the total leverage, is a continuous function of state variables. This is a natural extension of the long-only case, in which it is commonly assumed that the allocation within long-only assets is a continuous function of state variables.

Before presenting the approximation theorem, we first briefly review the results of Kratsios and Bilokopytov (2020).

Lemma 5.3.3. *Let $\mathcal{X} \subset \mathbb{R}^l$ be a compact set, and $\mathcal{Y} \subset \mathbb{R}^m$. Let $\rho : \mathbb{R}^n \mapsto \mathcal{Y}$ satisfy the following:*

(i) ρ is continuous and has a right inverse on $Im(\rho)$, i.e. $\exists \overrightarrow{\rho} : Im(\rho) \mapsto \mathbb{R}^n$, s.t. $\rho(\overrightarrow{\rho}(z)) = z$, $\forall z \in Im(\rho)$.

(ii) $Im(\rho)$ is dense in $\mathcal{Y}$.

Then, for any continuous $g : \mathcal{X} \mapsto \mathcal{Y}$, and any $\epsilon > 0$, there exists a choice of hyperparameters $\left(K, (d_1, \cdots, d_K)^\top\right)$ and parameter $\boldsymbol{\theta}$, such that the corresponding FNN $\tilde{f}_{\boldsymbol{\theta}} : \mathcal{X} \mapsto \mathbb{R}^n$ described in Definition 5.3.1 satisfies

$$\sup_{\boldsymbol{x} \in \mathcal{X}} \|\rho(\tilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) - g(\boldsymbol{x})\| < \epsilon, \forall \boldsymbol{x} \in \mathcal{X}. \tag{5.29}$$

Here $\|\cdot\|$ denotes the vector norm.

Proof. This is a direct application of Theorem 3.3 of Kratsios and Bilokopytov (2020) (for general topological spaces) in the metric space. □

Intuitively, the second assumption of Lemma 5.3.3 allows the use of an activation function (such as the softmax function) which output an open set, as long as this open set is dense in $\mathcal{Y}$ (which can be a closed set). The two assumptions ensures the existence of a continuous mapping of which the image almost covers $\mathcal{Y}$.

We then propose the following lemma, which is a generalization of the main theorem in this chapter.

Lemma 5.3.4. (Approximation of controls with a specific structure). *Assume a control function $p : \mathcal{X} \mapsto \mathcal{Z}$ has the structure*

$$p(\boldsymbol{x}) = \Phi(\Omega(\boldsymbol{x}), \boldsymbol{x}), \forall \boldsymbol{x} \in \mathcal{X}, \tag{5.30}$$

where $\mathcal{X}$ is compact, $\Omega \in C(\mathcal{X}, \mathcal{Y})$, i.e. Ω is a continuous mapping from $\mathcal{X}$ to $\mathcal{Y}$, and $\Phi : \mathcal{Y} \times \mathcal{X} \mapsto \mathcal{Z}$ is Lipschitz continuous on $\mathcal{Y} \times \mathcal{X}_i$, $\forall i = 1, \cdots, n$, where $\{\mathcal{X}_i, i = 1, \cdots, n\}$ is a partition of $\mathcal{X}$, i.e.

$$\begin{cases} \bigcup_{i=1}^n \mathcal{X}_i = \mathcal{X}, \\ \mathcal{X}_i \bigcap \mathcal{X}_j = \varnothing, \forall 1 \leq i, j \leq n. \end{cases} \tag{5.31}$$

If there exists $m \in \mathbb{N}$ and $\Upsilon : \mathbb{R}^m \mapsto \mathcal{Y}$ such that

(i) Υ is continuous and has a right inverse on $Im(\Upsilon)$.

(ii) $Im(\Upsilon)$ is dense in $\mathcal{Y}$.

Then for any $\epsilon > 0$, there exists a choice of hyperparameters $\Big(K, (d_1, \cdots, d_K)^\top\Big)$ and model parameters $\boldsymbol{\theta}$ such that the fully connected feedforward neural network function $\tilde{f}_{\boldsymbol{\theta}}$ defined in Definition 5.3.1 satisfies

$$\sup_{\boldsymbol{x}\in\mathcal{X}} \|\Phi\Big(\Upsilon\big(\tilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x})\big), \boldsymbol{x}\Big) - p(\boldsymbol{x})\| < \epsilon. \tag{5.32}$$

Proof. Let

$$L_\Phi = \max_{1\le i\le n} L_i, \tag{5.33}$$

where L_i is the Lipschitz constant for Φ on $\mathcal{Y} \times \mathcal{X}_i$.

Since $\Omega \in C(\mathcal{X})$ for compact $\mathcal{X}$, following Lemma 5.3.3, we know that $\forall\epsilon > 0$, there exist $\Big(K, (d_1, \cdots, d_K)^\top\Big)$ (the number of hidden layers and hidden nodes for each layer) and $\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}$ such that the corresponding FNN $\tilde{f}_{\boldsymbol{\theta}} : \mathcal{X} \mapsto \mathbb{R}^m$ described in Definition 5.3.1 satisfies

$$\sup_{\boldsymbol{x}\in\mathcal{X}} \|\Upsilon\big(\tilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x})\big) - \Omega(\boldsymbol{x})\| < \epsilon/L_\Phi, \tag{5.34}$$

Then

$$\begin{aligned}
\sup_{\boldsymbol{x}\in\mathcal{X}} \|\Phi\Big(\Upsilon\big(\tilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x})\big), \boldsymbol{x}\Big) - p(\boldsymbol{x})\| &= \sup_{1\le i\le n} \sup_{\boldsymbol{x}\in\mathcal{X}_i} \|\Phi\Big(\Upsilon\big(\tilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x})\big), \boldsymbol{x}\Big) - \Phi\Big(\Omega(\boldsymbol{x}), \boldsymbol{x}\Big)\| && (5.35)\\
&\le \sup_{1\le i\le n} \sup_{\boldsymbol{x}\in\mathcal{X}_i} L_i \cdot \Big(\|\Upsilon\big(\tilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x})\big) - \Omega(\boldsymbol{x})\|\Big) && (5.36)\\
&< \sup_{1\le i\le n} \frac{L_i}{L_\Phi}\epsilon && (5.37)\\
&\le \epsilon. && (5.38)
\end{aligned}$$

□

Remark 5.3.3. (Application of Lemma 5.3.4 to portfolios with stochastic allocation constraints). Normally, the universal approximation theorem only applies to the approximation of continuous functions defined on a compact set (Hornik, 1991). Lemma 5.3.4 extends the universal approximation theorem to a broader class of functions that have the structure of (5.30). Furthermore, Lemma 5.3.4 provides guidance on constructing neural network functions that handle stochastic constraints on controls which are usually difficult to address in stochastic optimal control problems. Consider the following example: the control $p : \mathcal{X} \mapsto \mathbb{R}^{N_a}$ has stochastic constraints such that $p(\boldsymbol{x}) \in [a(\boldsymbol{x}), b(\boldsymbol{x})]$ where $a, b : \mathcal{X} \mapsto \mathbb{R}^{N_a}$ are deterministic functions. This is a common setting in portfolio optimization problems in which allocation fractions to specific assets are subject to thresholds tied to the performance of the portfolio (which is stochastic). With Lemma 5.3.4, by parameterizing the control $p(x) \approx f_{\boldsymbol{\theta}}(\boldsymbol{x}) = a(\boldsymbol{x}) + \sigma\big(\tilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x})\big) \cdot (b(\boldsymbol{x}) - a(\boldsymbol{x}))$, where $\sigma(\cdot)$ is the sigmoid function and $\tilde{f}$ is a standard FNN defined in Definition 5.3.1, one can easily construct the corresponding neural network that satisfies the constraints naturally and be guaranteed that such a neural network can approximate the control well.

Finally, we present the following approximation theorem for the LFNN.

Theorem 5.3.2. *(Approximation of optimal control). Given the optimal control p^* of problem (5.11) and Assumption 5.3.1, $\forall \epsilon > 0$, there exist hyperparameters $\Big(K, (d_1, \cdots, d_K)^\top\Big)$, and $\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}$ such that the corresponding LFNN model $f_{\boldsymbol{\theta}}$ described in (5.16) satisfies the following:*

$$\sup_{\boldsymbol{x} \in \mathcal{X}} \|f_{\boldsymbol{\theta}}(\boldsymbol{x}) - p^*(\boldsymbol{x})\| < \epsilon. \tag{5.39}$$

Proof. From (5.16) and Lemma 5.3.1, we know that

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \psi\big(\tilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{x}\big) = \varphi\Big(\zeta\big(\tilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x})\big), \boldsymbol{x}\Big), \tag{5.40}$$

where $\tilde{f}$ is the FNN defined in Definition 5.3.1 and $\varphi : \tilde{\mathcal{Z}} \times \mathcal{X} \mapsto \mathbb{R}^{N_a}, \zeta : \mathbb{R}^{N_a+1} \mapsto \tilde{\mathcal{Z}}$ are defined in (5.19).

It can be easily verified that ζ satisfies the following:

(i) ζ is continuous and has a right inverse, e.g.

$$\zeta^{-1}(z) : Im(\zeta) \mapsto \mathbb{R}^{N_a+1}, \zeta^{-1}(z) = \begin{pmatrix} \log(z_1), \cdots, \log(z_{N_a}), \sigma^{-1}(z_{N_a+1}/p_{max}) \end{pmatrix}^\top, \tag{5.41}$$

where σ^{-1} is the inverse function of the sigmoid function.

(ii) $Im(\zeta)$ is dense in $\tilde{\mathcal{Z}}$. This is because $\overline{Im(\zeta)}$, the closure of $Im(\zeta)$, is $\tilde{\mathcal{Z}}$.

Furthermore, consider the partition of $\mathcal{X}$, $\{\mathcal{X}_1, \mathcal{X}_2\}$, which is defined in Definition 5.2.1. It is easily verifiable that φ is Lipschitz continuous on $\tilde{\mathcal{Z}} \times \mathcal{X}_1$ and $\tilde{\mathcal{Z}} \times \mathcal{X}_2$ respectively.

Finally, according to Assumption 5.3.1, $p^*(\boldsymbol{x}) = \varphi\big(\omega^*(\boldsymbol{x}), \boldsymbol{x}\big)$, where $\omega^* \in C(\mathcal{X}, \tilde{\mathcal{Z}})$.

Applying Lemma 5.3.4 with $\mathcal{Y} = \tilde{\mathcal{Z}}, \Omega(\cdot) = \omega^*(\cdot)$, $\Upsilon(\cdot) = \zeta(\cdot)$, and $\Phi(\cdot, \cdot) = \varphi(\cdot, \cdot)$, we know that there exist hyperparameters $\Big(K, (d_1, \cdots, d_K)^\top\Big)$, and model parameters $\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}$ such that the corresponding LFNN model $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \varphi\Big(\zeta\big(\tilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x})\big), \boldsymbol{x}\Big)$ satisfies the following:

$$\sup_{\boldsymbol{x} \in \mathcal{X}} \|f_{\boldsymbol{\theta}}(\boldsymbol{x}) - p^*(\boldsymbol{x})\| < \epsilon. \tag{5.42}$$

□

Theorem 5.3.2 shows that given any arbitrarily small tolerance $\epsilon > 0$, there exists a suitable choice of the hyperparameter of the LFNN model (e.g. the number of hidden layers and nodes) and a parameter vector $\boldsymbol{\theta}$, such that the corresponding parameterized LFNN

function is within this tolerance of the optimal control function.[1] In other words, with a large enough LFNN model (in terms of the number of hidden nodes), solving the unconstrained parameterized problem (5.25) approximately solves the original optimization problem (5.11) with any required precision.

5.3.3 Training LFNN

Since the numerical experiments involve the solution and evaluation of the optimal parameters $\boldsymbol{\theta}^*$ of the LFNN model (5.16) in problem (5.25), we briefly review how the parameters are computed in experiments.

In numerical experiments, the expectation in (5.25) is approximated by using a finite set of samples of the set $\boldsymbol{Y} = \{Y^{(j)} : j = 1, \cdots, N_d\}$, where N_d is the number of samples, and $Y^{(j)}$ represents a time series sample of *joint* asset return observations $R_i(t),\ i \in \{1, \cdots, N_a\}$, observed at $t \in \mathcal{T}$, where $\mathcal{T}$ is the rebalancing schedule.[2] Mathematically, problem (5.25) is approximated by

$$\inf_{\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}} \left\{ \frac{1}{N_d} \sum_{j=1}^{N_d} F\left(\mathcal{W}_{\boldsymbol{\theta}}^{(j)}, \hat{\mathcal{W}}_{\hat{p}}^{(j)} \right) \right\}. \tag{5.43}$$

Here $\mathcal{W}_{\boldsymbol{\theta}}^{(j)}$ is the wealth trajectory of the active portfolio following the LFNN parameterized by $\boldsymbol{\theta}$, and $\hat{\mathcal{W}}_{\hat{p}}^{(j)}$ is the wealth trajectory of the benchmark portfolio following the benchmark strategy $\hat{p}$, both evaluated on $Y^{(j)}$, the j-th time series sample.

We use a shallow neural network model, specifically, an LFNN model with one single hidden layer with 10 hidden nodes, i.e., $K = 1$ and $d_1 = 10$. We use the 3-tuple vector $(t, W_{\boldsymbol{\theta}}(t), \hat{W}(t))^\top$ as the input (feature) to the LFNN network. At $t \in [t_0, T]$, $W_{\boldsymbol{\theta}}(t)$ is the wealth of the active portfolio of the strategy that follows the LFNN model parameterized by $\boldsymbol{\theta}$, and $\hat{W}(t)$ is the wealth of the benchmark portfolio.

Then, the optimal parameter $\boldsymbol{\theta}^*$ can be numerically obtained by numerically solving problem (5.43) using standard optimization algorithms such as ADAM (Kingma and Ba, 2014). This process is commonly referred to as "training" of the neural network model, and $\boldsymbol{Y}$ is often referred to as the training data set (Goodfellow et al., 2016).

Note that the definition of φ in Equation (5.19) depends on non-differentiable step functions $\mathbf{1}_{\boldsymbol{x} \in \mathcal{X}_1}$ and $\mathbf{1}_{\boldsymbol{x} \in \mathcal{X}_2}$. It can be easily verified that φ is piecewise analytic under analytic partition (in short, PAP), which has the property of being differentiable almost everywhere (Lee et al., 2020), i.e. the set of inputs making the function non-differentiable is contained in a measure-zero set. Due to the recurrent nature of the neural network model, the objective function is essentially a chaining composition of φ and smooth functions (i.e. the

[1] The distance is defined in (5.39), i.e. the supremum of the pointwise distance over the extended state space $\mathcal{X}$.

[2] Note that the corresponding set of asset prices can be easily inferred from the set of asset returns, or vice versa.

wealth evolution function, sigmoid function, and affine function). According to Lee et al. (2020), the objective function is also PAP, thus is differentiable almost everywhere. In rare cases when the function is not differentiable, PyTorch computes the gradient either by using sub-gradients or continuity to allow optimization to proceed.[3]

In our specific case of portfolio optimization with leverage constraints, the objective function is only non-differentiable when the wealth of the portfolio reaches zero at any time during the investment horizon. However, in numerical experiments, the wealth has not reached zero in any of the paths, most likely due to the mildly capped total leverage. Therefore, concerns about the non-differentiability of the objective function should be further alleviated.

Once $\boldsymbol{\theta}^*$ is numerically obtained, the resulting optimal strategy $f_{\boldsymbol{\theta}^*}$ is evaluated on a separate "testing" data set $\boldsymbol{Y}^{test}$, which contains a different set of samples generated from either the same distribution of the training process or a different process (depending on experiment purposes) so that the "out-of-sample" performance of $f_{\boldsymbol{\theta}^*}$ is assessed.

5.4 LFNN model vs clipped-form solution

In this section, we compare the performance of the strategy following the learned shallow LFNN model (which we refer to as the "neural network strategy") with the clipped form (4.31) of the closed-form solution, and provide empirical validation of the LFNN approach. In particular, we assess and compare the performance of the neural network strategy and the clipped form, under the following investment scenario described in Table 5.1.

Investment horizon T (years)	10
Assets	CRSP cap-weighted index (real) 30-day T-bill (U.S.) (real)
Data	Double-exponential jump-diffusion model calibrated to concatenated 1940:8-1951:7 & 1968:9-1985:10
Initial portfolio wealth/annual cash injection	100/10
Rebalancing frequency	Monthly, quarterly, semi-annually, annually
Maximum leverage	1.3
Benchmark equity percentage	0.7
Outperformance target rate β	1% (100 bps)

Table 5.1: Investment scenario.

Note that the asset price model is calibrated to the concatenated returns of 1940:8-1951:7 & 1968:9-1985:10, the historical high inflation periods. Since investing in high inflation

[3]See Pytorch Documentation.

periods is currently of much interest, we use this data as an interesting example. Further discussion of high inflation regimes is delayed to Chapter 6. However, it is also worth noting that the exact historical period for model calibration does not matter for this experiment, and only serves the purpose of obtaining some model parameters.

We assume the stock index and the bond index prices follow a double exponential jump model (4.8), see e.g., (Kou, 2002; Kou and Wang, 2004), i.e., for the jump variable ξ_i, $y_i = \log(\xi_i)$ follows the double exponential distribution with density functions $g_i(y_i)$ defined as follows

$$g_i(y_i) = \nu_i \iota_i e^{-\iota_i y_i} \mathbf{1}_{y_i \geq 0} + (1 - \nu_i) \varsigma_i e^{-\varsigma_i y_i} \mathbf{1}_{y_i < 0},\ i = 1, 2. \tag{5.44}$$

where ν_i is the probability for an upward jump, and ι_i and ς_i are parameters that describe the upward jump and downward jump respectively. The double exponential jump-diffusion model allows the flexibility of modeling asymmetric upward and downward jumps in asset prices, which are suitable for market conditions such as the current high inflation scenario.[4]

Using the threshold technique (Mancini, 2009; Cont et al., 2011; Dang and Forsyth, 2016), we calibrate the double exponential jump-diffusion models to historical returns. The calibrated parameters can be found in Table 5.2. Then, we construct a training data set $\boldsymbol{Y}$ and a testing data set $\boldsymbol{Y}^{test}$ by sampling from the calibrated model, each with 10,000 samples.

μ_1	σ_1	λ_1	ν_1	ι_1	ς_1	μ_2	σ_2	λ_2	ν_2	ι_2	ς_2	ρ
0.051	0.146	0.178	0.2	7.13	7.33	-0.014	0.017	0.321	0	N/A	44.48	0.14

Table 5.2: Estimated annualized parameters for double exponential jump-diffusion model (5.44) based on scenario described in Table 5.1.

We consider a shallow LFNN model which contains 1 hidden layer with 10 hidden nodes. The neural network strategy follows the LFNN model learned from $\boldsymbol{Y}$. We then evaluate the performance of the neural network strategy and the approximate form (4.31) on the testing data set $\boldsymbol{Y}^{test}$. Specifically, we compare the value of the CD objective function (4.7) for the neural network strategy and the clipped form on $\boldsymbol{Y}^{test}$. In particular, this training/testing process is repeated for various rebalancing frequencies from monthly to annually, as described in Table 5.1.

In Table 5.3, we can see that the neural network strategy consistently outperforms the clipped form in terms of the objective function value for all rebalancing frequencies. From Table 4.2 we can see that the objective function values of both the neural network strategy and the clipped form converge at roughly a first-order rate as $\Delta t \downarrow 0$. Assuming this to be true, we extrapolate the solution to $\Delta t = 0$ using Richardson extrapolation. These extrapolated values are estimates of the exact value of the continuous-time CD objective function (4.6) for the clipped form and the neural network strategy. We can see that the

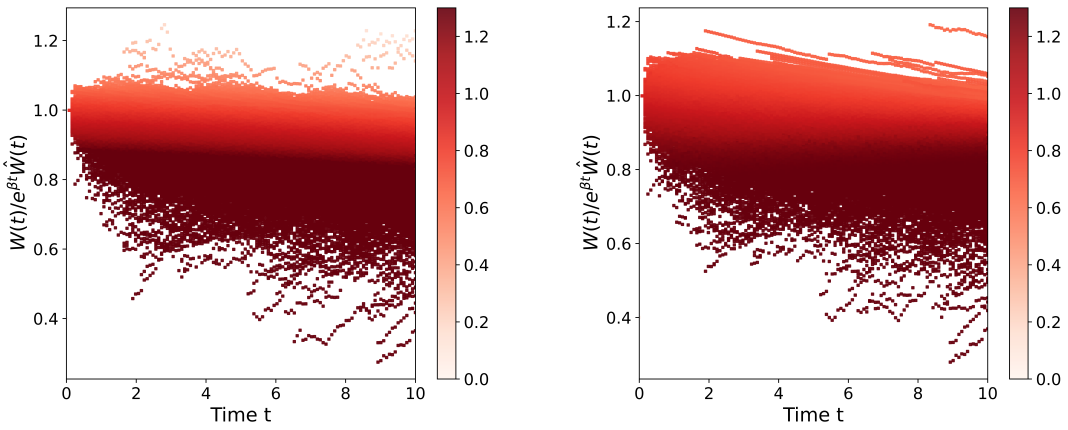
[4]We remind the reader that the closed-form solution is derived under the jump-diffusion model.

Closed-form solution objective function value: 418					
Strategy	$\Delta t = 1$	$\Delta t = 1/2$	$\Delta t = 1/4$	$\Delta t = 1/12$	$\Delta t = 0$
Clipped form	545	504	479	467	461 (extrapolated)
Neural network	537	498	476	464	458 (extrapolated)

Table 5.3: CD objective function values. Results shown are evaluated on $\boldsymbol{Y}^{test}$, the lower the better.

neural network strategy still outperforms the clipped form in terms of the extrapolated objective function value. We can also see that the extrapolated neural network objective function value is lower than the (suboptimal) clipped form extrapolated value. Observe from Table 5.3 that the clipped control and the NN control (in the extrapolated limit as $\Delta t \to 0$) result in objective function values larger than the unconstrained (closed form) control, which is to be expected.

It is also worth noting that, with monthly rebalancing, the numerical results for both the clipped form and the neural network solution are very close to the extrapolated value. In fact, Forsyth and Vetzal (2022) point out that the empirical performance of strategies only has marginal gains with a higher rebalancing frequency than monthly rebalancing.



(a) Clipped form

(b) Neural network strategy

Figure 5.1: Stock allocation fraction w.r.t. tracking ratio $W(t)/\left(e^{\beta t}\hat{W}(t)\right)$ and time t. Results are based on the evaluation on the testing data set $\boldsymbol{Y}^{test}$, and monthly rebalancing (i.e. $\Delta t = 1/12$).

Finally, we compare the neural network allocation strategy with the clipped form strategy. Specifically, in Figure 5.1, we consider the case of monthly rebalancing and present the scatter plots of the allocation fraction in the stock index with respect to time t and the ratio between the wealth of the active portfolio $W(t)$ and the elevated target $e^{\beta t}\hat{W}(t)$. For simplicity,

we call this ratio the "tracking ratio". We plot the 3-tuple $\left(\frac{W(t)}{e^{\beta t}\hat{W}(t)}, t, p_1\big(W(t), \hat{W}(t), t\big)\right)$ (obtained from the evaluation of the strategies on samples from $\boldsymbol{Y}^{test}$) by using time t as the horizontal axis, the tracking ratio $\frac{W(t)}{e^{\beta t}\hat{W}(t)}$ as the vertical axis, and the values of the corresponding allocation fraction to the cap-weighted index $p_1(W(t), \hat{W}(t), t)$ to color the scattered dots on the plot. A darker shade of the color indicates a higher allocation fraction.

As we can see from Figure 5.1, the stock allocation fraction of the neural network strategy behaves similarly to the stock allocation fraction from the clipped form. Both strategies invest more wealth in the stock when the tracking ratio is lower, which is consistent with the insights we obtained in Section 4.3.1. In addition, the transition patterns of the allocation fraction of the two strategies are also highly similar. One can almost draw an imaginary horizontal dividing line around $\frac{W(t)}{e^{\beta t}\hat{W}(t)} = 0.9$ that separates high stock allocation and low stock allocation for both strategies.

We remark that a common criticism towards the use of neural networks is about the lack of interpretability compared to more interpretable counterparts such as the regression models (Rudin, 2019). In this section, we see that the neural network strategy closely resembles (an approximate form of) the closed-form solution for the CD objective. The closed-form solution, in turn, complements the neural network model and offers an alternative way of interpreting results obtained from the neural network.

In summary, we find that a small neural network structure with one single hidden layer and only 10 hidden nodes achieves good performance compared to the approximate form of the closed-form solution, which provides further empirical evidence to support Theorem 5.3.2.

5.5 Conclusions

In this chapter, we discussed the multi-period portfolio optimization problem with bounded leverage constraints. The leverage constraints introduce additional complexity to the multi-period portfolio optimization problem, resulting in a challenging constrained optimization problem.

By proposing a novel leverage-feasible neural network (LFNN), we transformed the original constrained optimization problem into an unconstrained optimization problem that is computationally feasible.

Furthermore, we mathematically proved that the solution to the parameterized unconstrained optimization problem can approximate the optimal control of the original constrained optimization problem arbitrarily well when appropriate hyperparameters are chosen.

Finally, through numerical experiments, we showed that the neural network strategy attains comparable performances relative to the clipped form of the closed-form solution on synthetic data sets, providing empirical evidence of the efficiency of the proposed model.

Chapter 6

Case study: optimal leverage portfolio in high inflation

6.1 Introduction

Since the global outbreak of COVID-19 in March 2020, we have observed significant worldwide inflation. Before the pandemic, the U.S. economy had seen almost four decades of low inflation. The sudden shift from a long-term low-inflation environment to a high-inflation environment causes large uncertainty and volatility in the financial market. In 2022, the technology-heavy NASDAQ stock index recorded a yearly return of -33.10% (NASDAQ, 2023).

Equally concerning is the uncertainty around the duration of this round of high inflation. Some believe that the geopolitical tensions and the COVID-19 pandemic will overturn the trend of globalization and lead to global supply chain restructuring (Javorcik, 2020) which may cause a higher cost of production in the foreseeable future. Moreover, Ball et al. (2022) suggests that the future inflation rate may remain high if the unemployment rate remains higher than expected.

Going forward, investors have to be cognizant of the risk of inflation. If we enter into a long period of even moderate inflation, will the traditional passive approach using a mix of stock indexes and moderate-term bonds still work? Furthermore, how should an active manager invest in order to achieve consistent outperformance over the passive benchmark? In this chapter, we aim to answer these questions.

To gain some initial insights into investing during high inflation regimes, we first study the performances of fixed-mix constant weight strategies on bootstrap resampled data from filtered historical high inflation periods. Fixed-mix strategies allocate a fixed fraction of the portfolio wealth to the underlying assets and are common choices of benchmark strategies for large sovereign wealth funds (CPP Investments, 2022; Norges Bank, 2022). Our preliminary findings show that the fixed-mix strategies with the equal-weighted stock index achieves

partial first-order stochastic dominance over fixed-mix strategies that use the cap-weighted stock index during high inflation regimes.

We then consider an active portfolio with leverage constraints and apply the LFNN model proposed in Chapter 5 to an investment case with four underlying assets, which include the U.S. equal-weighted stock index, the U.S. cap-weighted stock index, the 30-day U.S. treasury bill index, and the 10-year U.S. treasury bond index. We conduct numerical experiments on bootstrap resampled data under the cumulative quadratic tracking shortfall (CS) objective and obtain a leverage-constrained strategy that outperforms the benchmark with more than 2% higher median (annualized) internal rate of return (IRR), and more than 90% probability of achieving a higher terminal wealth.

Furthermore, we evaluate the strategy learned from high-inflation periods in low-inflation environments and show that the optimal high-inflation strategy still outperforms the benchmark.

In summary, we make the following contributions in this chapter.

(i) We assess the performance of the passive investment strategies on resampled data from filtered historical high inflation periods. We find that the fixed-mix strategy with the equal-weighted stock index partially stochastically dominates the fixed-mix strategy which uses the cap-weighted stock index in resampled high inflation regimes. This suggests that the equal-weighted stock index is preferred over the cap-weighted stock index during high inflation.

(ii) We apply the neural network method to bootstrap resampled asset returns with four underlying assets, including the equal-weighted and cap-weighted stock index, and the 30-day/10-year treasury bond indexes. The dynamic strategy that follows the learned LFNN model outperforms the fixed-mix benchmark strategy with a high probability throughout the investment horizon, with a 2% higher median (annualized) internal rate of return (IRR), and more than 90% probability of achieving a higher terminal wealth. Furthermore, the learned allocation strategy suggests that the equal-weighted stock index and short-term bonds are preferable investment assets during high inflation regimes.

(iii) We show that the neural network strategy learned on high inflation periods still outperforms the benchmark when tested in a low-inflation environment, demonstrating the robustness of the strategy's outperformance with respect to changes in market conditions.

6.2 Passive investing in historical high inflation periods

6.2.1 Filtering historical high inflation regimes

To understand the performance of passive strategies in high inflation periods, we should first isolate the historical high inflation periods. In particular, we deploy a filtering algorithm to uncover historical periods of high, sustained inflation from the historical time series data (1926-2022). We use monthly data from the Center for Research in Security Prices (CRSP) over the 1926:1-2022:1 period[1][2] , which also includes the U.S. CPI index.

Our objective is to select high inflation periods as determined by the CPI index. Monthly data is quite volatile, so we use the following filtering procedure. We use a moving window of k months, and we determine the cumulative CPI index log return (annualized) in this window. If the cumulative annualized CPI index log return is greater than a cutoff, then all the months in the window are flagged as part of a high inflation regime. Note that some months may appear in more than one moving window. Any months which do not meet this criterion are considered to be in low-inflation regimes. The pseudo-code can be found in Algorithm 6.2.1.

Algorithm 6.2.1: Pseudocode window inflation filter

Data:
  CPI[i]; $i = 1, \ldots, N$ /* CPI Index */
  Cutoff /* High inflation cutoff: annualized */
  Δt /* CPI index time interval */
  K /* smoothing window size */
Result: Flag[i]; $i = 1, \ldots, N$ /* = 1 high inflation month; = 0 otherwise */
/* initialization */
Flag[i] =0; $i = 1, \ldots, N$;
for $i = 1, \ldots, N - K$ **do**
  if $\log(CPI[i + K]/CPI[i])/(K * \Delta t) > Cutoff$ **then**
    for $j = 0, \ldots, K$ **do**
      Flag[i+j] = 1 ;
    end
  end
end

[1]The date convention is that, for example, 1926:1 refers to January 1, 1926.

[2]More specifically, results presented here were calculated based on data from Historical Indexes, ©2022 Center for Research in Security Prices (CRSP), The University of Chicago Booth School of Business. Wharton Research Data Services (WRDS) was used in preparing this article. This service and the data available thereon constitute valuable intellectual property and trade secrets of WRDS and/or its third-party suppliers.

This approach requires the specification of the cutoff and the window size. The average annual inflation over the period 1926:1-2022:1 was 2.9%. In other words, inflation of about 3% was normal in the past century. In fact, Federal Reserve policymakers have been targeting the inflation rate of 2% over the long run to achieve maximum employment and price stability (The Federal Reserve, 2011). After some experimentation, we use a cutoff of 5% as the threshold for high inflation, which is more than double the Fed's target inflation rate. Figure 6.1 shows the filtering results for windows of size 12, 60, and 120 months. We can see that the five-year window produces two obvious inflation regimes: 1940:8-1951:7 and 1968:9-1985:10, which correspond to well-known market shocks (i.e. the second world war, and price controls; the oil price shocks and stagflation of the seventies). Increasing the window size to 10 years results in similar-looking plots as the five-year window size, but the number of months in each window increases, and the average inflation rate is lower. Since our objective is to determine the effect of high inflation periods on allocation strategies, we choose the five-year window size.

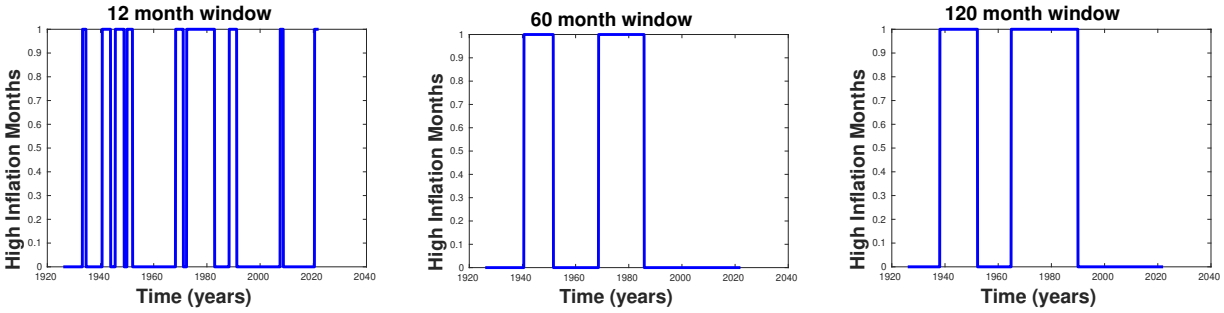


Figure 6.1: High inflation regimes, using the moving-window method, with the window size shown. The cutoff for high inflation regimes was 0.05. High-inflation months have a label value of one, and low-inflation months have a label value of zero. CPI data identified from the historical period 1926:1-2022:1.

Table 6.1 shows the average annual inflation over the two regimes identified from the moving-window filter.

Time Period	Average Annualized Inflation
1940:8-1951:7	.0564
1968:9-1985:10	.0661

Table 6.1: Inflation regimes determined using a five-year moving window with a cutoff inflation rate of 0.05.

For possible investment assets, we consider the 30-day U.S. T-bill index (CRSP designa-

tion "t30ind"). In addition, we construct a constant maturity 10-year U.S. treasury index.[3] We also study the cap-weighted stock index (CapWt) and the equal-weighted stock index (EqWt), also from CRSP.[4] The CRSP indexes are total return indexes, which include all distributions for all domestic stocks trading on major U.S. exchanges. All of these various indexes are in nominal terms, so we adjust them for inflation by using the CPI index.

To gain some intuition on the behavior of asset returns during the inflation periods, we assume that each real (adjusted by CPI) index follows geometric Brownian motion (GBM). For example, given an index with value S, then

$$dS = \mu S\, dt + \sigma S\, dZ \tag{6.1}$$

where dZ is the increment of a Wiener process. We use maximum likelihood estimation to fit the drift rate μ (expected arithmetic return) and volatility σ in each regime, for each index, as shown in Table 6.2. We also show a series constructed by: converting the indexes in each regime to returns, concatenating the two return series, and converting the concatenated return series back to an index. This concatenated index does not correspond to an actual historical index but is a pseudo-index constructed from high inflation regimes. This amounts to a worst-case sequence of returns in terms of the duration of historical high inflation periods, that could plausibly be expected during a long period of high inflation.

It is striking that in each historical high inflation regime (i.e., 1940:8-1951:7 and 1968:9-1985:10) in Table 6.2, the drift rate μ for the equal-weighted index is much larger than the drift rate for the cap-weighted index. We can observe that the mean geometric return (which is the median return) for the cap-weighted index, in the period 1968:9-1985:10, was only about one percent per year.

It is also noticeable that bonds performed very poorly in the period 1940:8-1951:7. As well, during the period 1968:9-1985:10, there was essentially no term premium for 10-year treasuries, compared with 30-day T-bills. In addition, the 10-year treasury index had much higher volatility compared to the 30-day T-bill index. Looking at the concatenated series, it appears that 30-day T-bills are arguably the better defensive asset here since the volatility of this index is quite low (but with a negative (real) drift rate).

Consequently, in the following, we will focus attention on 30-day T-bills, the cap-weighted index, and the equal-weighted index.

[3]The 10-year treasury index was generated from monthly returns from CRSP back to 1941 (CRSP designation "b10ind"). The data for 1926-1941 are interpolated from annual returns in Homer and Sylla (1996). The 10-year treasury index is constructed by (a) buying a 10-year treasury at the start of each month, (b) collecting interest during the month, and then (c) selling the treasury at the end of the month. We repeat the process at the start of the next month. The gains in the index then reflect both interest and capital gains and losses.

[4]The capitalization-weighted total returns have the CRSP designation "vwretd", and the equal-weighted total returns have the CRSP designation "ewretd".

Index	μ	σ	$\mu - \sigma^2/2$
		1940:8-1951:7	
CapWt	0.079	0.140	.069
EqWt	0.145	0.190	.127
10 Year Treasury	-0.035	0.036	-.036
30-day T-bill	-0.050	0.029	-.050
		1968:9-1985:10	
CapWt	0.026	0.164	.013
EqWt	0.065	0.220	.041
10 Year Treasury	0.011	0.093	.007
30-day T-bill	0.009	0.012	.009
	Concatenated: 1940:8-1951:7 and 1968:9 - 1985:10		
CapWt	0.049	0.156	.038
EqWt	0.098	0.209	.076
10 Year Treasury	-0.008	0.076	-.011
30-day T-bill	-0.014	0.022	-.014

Table 6.2: GBM parameters for the indexes shown. All indexes are real (deflated). μ is the expected annualized arithmetic return. σ is the annualized volatility. $(\mu - \sigma^2/2)$ is the annualized mean geometric return, which is the median return.

6.2.2 Non-contiguous bootstrap resampling

Once we have obtained the filtered historical high inflation data series from Section 6.2.1, it becomes necessary to generate training and testing data sets from the original time series data. Same as in Section 2.3, we use stationary block bootstrap resampling method (Politis and Romano, 1994) to generate the datasets.

Recall that the stationary block bootstrap uses random blocksizes which preserves the stationarity of the original time series data. Briefly, each bootstrap resample consists of (i) selecting a random starting date in the historical return series, (ii) selecting a block (of random size) of consecutive returns from this start date, and (iii) repeating this process until a sample of the total desired length is obtained. The pseudo-code can be found in Algorithm 2.3.1.

Typically, the bootstrap technique resamples from data sourced from one contiguous segment of historical periods. However, the moving-window filtering algorithm has identified two non-contiguous historical high inflation regimes. To apply the bootstrap method, there are two intuitive methods: 1) concatenate the two historical high inflation regimes first, then bootstrap from the concatenated combined series following the steps in the previous paragraph, or 2) bootstrap within each regime (i.e., using circular block bootstrap resampling within each regime), then combine the bootstrapped resampled data points.

More specifically, the second bootstrap method comprises the following steps. Each bootstrap resample consists of (i) selecting a random segment (probability proportional to the length of the segment), (ii) selecting a random starting date in the selected segment, (iii) then selecting a block (of random size) of consecutive returns from this start date, (iv) in the event that the end of the data set in a segment is reached, use circular block bootstrap resampling within that segment, and (v) repeating this process until a sample of the total desired length is obtained.

We compare the bootstrapped data from concatenated segments and separate segments, by evaluating the performance of the 70%/30% equal-weighted index/T-bill fixed-mix portfolio, using the investment scenario described in Table 6.3.

Investment horizon T (years)	10
Equity market indexes	CRSP equal-weighted index (real)
Bond index	30-day T-bill (U.S.) (real)
Index Samples	Concatenated 1940:8-1951:7, 1968:9-1985:10
Initial portfolio wealth	100
Rebalancing frequency	Monthly
Cash injection	0

Table 6.3: Investment scenario.

We can observe from Table 6.4 that the strategy performance on bootstrap resampled data using two methods only varies slightly. This indicates that the two methods do not yield much difference for practical purposes. This is indeed expected - after all, the difference between the two methods only occurs when a random block crosses the edge of each of the segments. However, such a situation only occurs with a very low probability. Except for this low-probability situation, the two bootstrap methods are identical.

	Median$[W_T]$	E$[W_T]$	std$[W_T]$	5th Percentile
Bootstrap from concatenated segments	174.2	204.2	125.9	66.8
Bootstrap from separate segments	176.9	208.0	132.4	65.4

Table 6.4: Effect of bootstrap method - bootstrap from concatenated segments vs bootstrap from separate segments, on the statistics of the final wealth $W(T)$ at $T = 10$ years. Constant weight, scenario in Table 6.3. Equity weight: 0.7, rebalanced monthly. Bond index: 30-day T-bill. Equity index: equal-weighted. Concatenated series: 1940:8-1951:7, 1968:9-1985:10 (high inflation regimes). All quantities are real (inflation-adjusted). Initial wealth 100. Bootstrap resampling, $10,000$ resamples).

In this article, we adopt the first method, i.e., we concatenate the historical regimes first, then bootstrap from the combined series. This method is also adopted by Anarkulova et al.

(2022), where stock returns from different countries are concatenated and the bootstrap is applied to the combined data.

6.2.3 Passive strategies in high inflation

In this section, we compare the performances of two fixed-mix strategies. The first strategy, the "EqWt" strategy, maintains a 70% allocation to the equal-weighted index, and 30% allocation to the 30-day T-bill index. The second strategy, the "CapWt" strategy, maintains a 70% allocation to the cap-weighted index, and a 30% allocation to the 30-day T-bill index.

In our numerical experiment, we consider the investment scenario described in Table 6.3. Briefly, we begin with an initial wealth of 100 for both strategies, with no further cash injections and withdrawals. The investment horizon is 10 years, with monthly rebalancing to maintain the constant weights in the portfolio. We evaluate the investment results by examining the distribution of the final wealth $W(T)$ at $T = 10$ years. Using $10,000$ block bootstrap resampled data samples (Politis and Romano, 1991; Dichtl et al., 2016; Anarkulova et al., 2022) from the concatenated CRSP combined time series from 1940:8-1951:7 and 1968:9-1985:10, with an expected blocksize of six months. We then compare the CDF (cumulative distribution functions) of the terminal wealth of the EqWt strategy and the CapWt strategy.

In Figure 6.2, we compare the CDFs of the terminal wealth of the EqWt strategy and the CapWt strategy. Recall the concept of partial stochastic dominance in Definition (2.4.1). Remarkably, the EqWt strategy appears to *partially* stochastically dominate the CapWt strategy, as the CDF curve of the EqWt strategy appears to be entirely on the right side of the CDF curve of the CapWt strategy, except at very low probability values. In fact, close examination shows that the curves cross at the point $CDF_{EqWt}(W_{lo}) = CDF_{CapWt}(W_{lo}) \simeq .02$, with a slight underperformance of the EqWt strategy compared to the CapWt strategy in this extreme left tail.

Note that both strategies perform very poorly in this extreme left tail and the fact that the CapWt strategy is marginally better here but is probably not practically meaningful.

It is in fact rare to find two strategies such that one strategy stochastically dominates another strategy, with the usual definition of the first order stochastic dominance where $(W_{lo}, W_{hi}) = (-\infty, +\infty)$ in Definition 2.4.1. as this would mean that one strategy is strictly better than another strategy in terms of the distribution of terminal wealth. In other words, any investor who prefers more rather than less would choose the dominating strategy. The fact that the EqWt strategy partially dominates the CapWt strategy seems to suggest that the equal-weighted stock index is almost definitely a better choice for the stock index than the cap-weight stock index during high inflation times.

However, if we examine more recent data[5] the situation is not so clear (Taljaard and

[5]Since about 2010. Of course, this is outside a period of sustained high inflation.

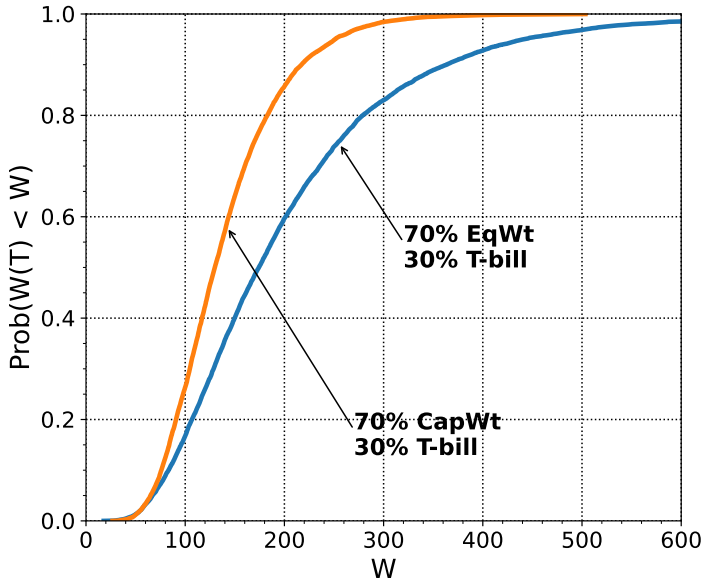


Figure 6.2: Cumulative distribution function of final real wealth W at $T = 10$ years, bootstrap resampling expected blocksize one year, $10,000$ resamples (Appendix 6.2.1). $T = 10$ years. Data: concatenated returns, 1940:8-1951:7, 1968:9-1985:10. Scenario described in Table 6.3.

Mare, 2021), since the equal-weighted index appears to underperform. However, Taljaard and Mare (2021) suggest that this is due to the recent market concentration in tech stocks.[6]

In fact, a plausible explanation for the historical outperformance of an equal-weighted index is that it is simply due to the small-cap effect, which was not widely known until about 1981 (Banz, 1981). Plyakha et al. (2021) acknowledge that the equal-weighted index has significant exposure to the size factor. However, Plyakha et al. (2021) argue that the equal-weighted index also has a larger exposure to the value factor. In addition, there is a significant *alpha* effect due to the contrarian strategy of frequent rebalancing to equal weights. It would appear to be simplistic to dismiss an equal weight strategy on the grounds that this is simply a small cap effect that has become less effective.

[6] As of February 2023, Apple, Microsoft, Amazon and Alphabet (A and C) in total comprised 17% of the market capitalization of the S&P 500.

6.3 Portfolio optimization for outperforming the benchmark

As discussed in Chapter 1, large sovereign wealth funds such as the CPP or the Oil Fund were founded a decade after the last inflation period ended in the 1980s. Considering their recent meager outperformance over passive benchmarks and the lack of track record in inflation regimes, how can such funds outperform the benchmarks with a high probability in a potential persistent inflation environment?

In this section, we attempt to suggest a solution to this question based on the LFNN technique discussed in Chapter 5. We conduct a case study of a multi-period portfolio optimization problem, with the goal of outperforming a passive fixed-mix 70/30 benchmark in high inflation regimes. We allow the active portfolio to take a bounded amount of leverage, to mimic alternative asset holdings of the sovereign wealth funds. We then apply the LFNN model to find the optimal portfolio under the cumulative quadratic tracking shortfall (CS) objective function.

6.3.1 Choice of objective function

When studying a portfolio optimization problem, the choice of the objective function is critical, since it determines the desired characteristics of the optimal portfolio.

As discussed in Chapter 4, the CD objective function (4.7) allows tracking of an elevated target throughout the investment horizon. However, one caveat of the CD objective function is that it not only penalizes the underperformance relative to the elevated target but also penalizes the outperformance over the elevated target. In practice, the outperformance of the elevated target is favorable, and managers may not want to penalize the strategy when it occurs. Therefore, in such cases, it may be more appropriate to consider a one-sided objective function, such as the following cumulative quadratic shortfall (CS) objective.

$$(CS(\beta)): \quad \inf_{\boldsymbol{\theta}\in\mathbb{R}^{N_{\boldsymbol{\theta}}}} \mathbb{E}_{\boldsymbol{\theta}}^{(t_0,w_0)}\left[\sum_{t\in\mathcal{T}}\Big(\min\big(W_{\boldsymbol{\theta}}(t)-e^{\beta t}\hat{W}(t),0\big)\Big)^2+\epsilon W_{\boldsymbol{\theta}}(T)\right], \tag{6.2}$$

The CS objective function in (6.2) only penalizes the underperformance against the elevated target. Here $\epsilon W(T)$ is a regularization term. We remark that problem (6.2) without the regularization term can be ill-posed. To see this, consider a case where $W_{\boldsymbol{\theta}}(t) \gg e^{\beta t}\hat{W}(t)$, for some $t \in [t_0, T]$. In this case, the future cumulative quadratic shortfall (on $[t, T]$) will almost surely be zero without the regularization term, so the control from thereon has no effect on the objective function under that scenario. We choose ϵ to be a small positive scalar. As William Bernstein once said, "if you have won the game, stop playing." If one has accumulated as much wealth as Warren Buffet, then it does not matter what assets she

invests in. The positive regularization factor of ϵ forces the strategy to put all wealth into less risky assets when the portfolio has already performed extremely well.

To illustrate the superiority of the CS objective in practice, we design a numerical experiment to compare the CS objective with the symmetric CD objective in the following problem (6.3) with the same LFNN parameterization.

$$\text{(Parameterized } CD(\beta)\text{)}: \quad \inf_{\boldsymbol{\theta}\in\mathbb{R}^{N_{\boldsymbol{\theta}}}} \mathbb{E}_{\boldsymbol{\theta}}^{(t_0,w_0)}\left[\sum_{t\in\mathcal{T}}\left(W_{\boldsymbol{\theta}}(t)-e^{\beta t}\hat{W}(t)\right)^2\right]. \tag{6.3}$$

Specifically, we adopt the investment scenario in Table 5.1 with $\beta = 0.02$ and reuse the training and testing data sets simulated from the calibrated double exponential jump-diffusion model, as discussed in Section 5.4.[7] The neural network strategies follow the trained LFNN models on the training data set $\boldsymbol{Y}$ for both the CS and CD objectives. We then evaluated both strategies on the same testing data set $\boldsymbol{Y}^{test}$.

We compare the *wealth ratio*, i.e., the wealth of the managed active portfolio divided by the wealth of the benchmark portfolio, over time, for both strategies. The wealth ratio metric reflects how well the active strategy performs against the benchmark strategy along the investment horizon; a higher wealth ratio metric is better. Below we show the percentiles of the wealth ratio for both strategies evaluated on $\boldsymbol{Y}^{test}$.

We can see from Figure 6.3 that the CS strategy (neural network strategy trained under the CS objective) yields a more favorable wealth ratio than the CD strategy (neural network strategy trained under the CD objective). On average, the CS strategy achieves a higher terminal wealth ratio than the CD strategy. Even in the 20th percentile case, the CS strategy lags initially but recovers over time.[8] The result indicates that the CS objective might be a wiser choice for managers in practice. In the following numerical experiments with bootstrap resampled data, we use the CS objective (6.2) instead of the CD objective (6.3). Therefore, we choose the CS objective function in the following numerical experiments in this chapter.

[7] We choose the synthetic data set because we know the neural network strategy under the CD objective shows comparable performance to the clipped form of the closed-form solution, as discussed in Section 5.4, hence it should be near optimal.

[8] The CS strategy starts with a higher allocation to the stock, and thus encounters more volatility early on.

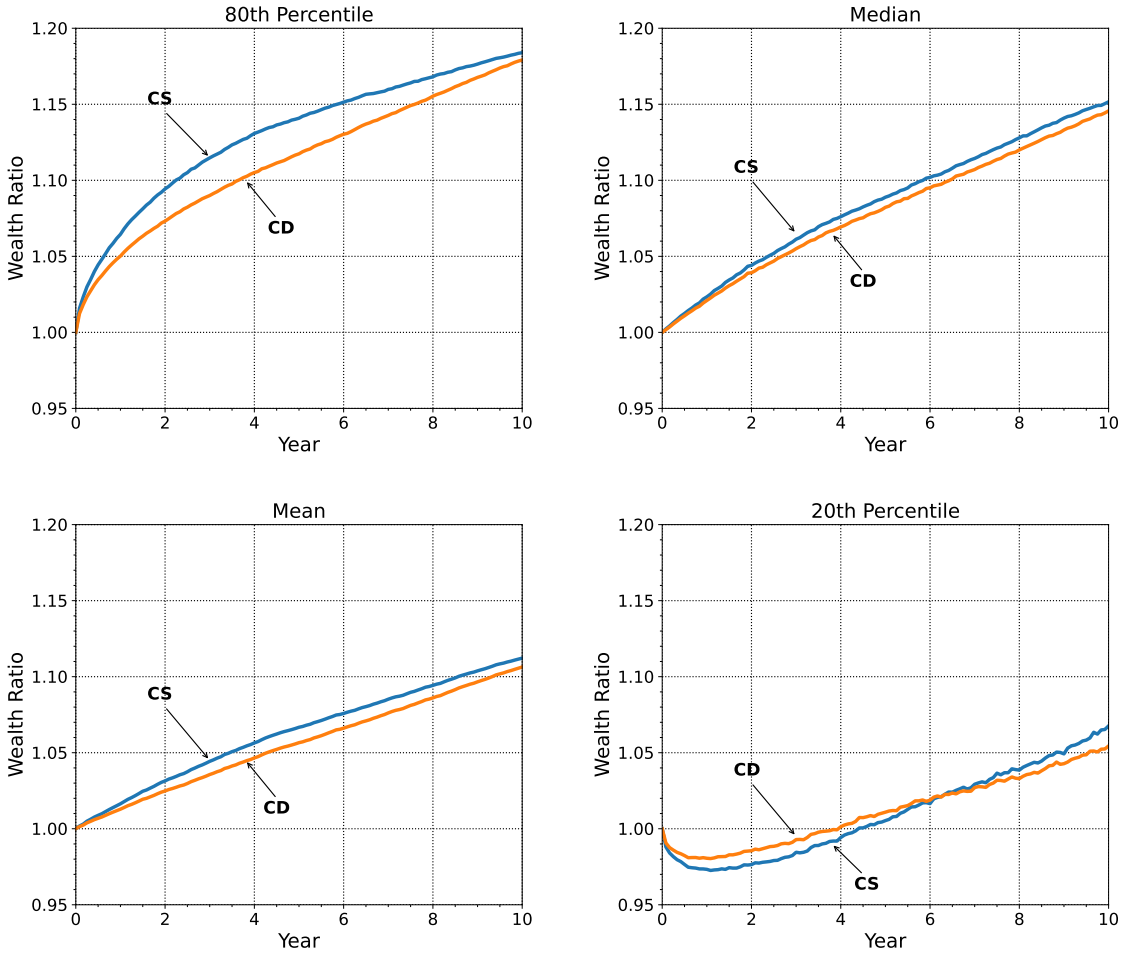


Figure 6.3: Percentiles of wealth ratio of the neural network strategy (i.e., the neural network model) learned under the cumulative quadratic tracking difference (CD) objective and the neural network strategy learned under the cumulative quadratic shortfall (CS) objective. The results shown are based on evaluations on the testing data set $\boldsymbol{Y}^{test}$.

6.3.2 High-inflation case study

The details of the investment specification are given in Table 6.5. Briefly, the active portfolio and benchmark portfolio begin with the same initial wealth of 100 at $t_0 = 0$. Both portfolios are rebalanced monthly. The investment horizon is 10 years, and there is an annual cash injection of 10 for both portfolios, evenly divided over 12 months.

We consider an empirical case in which we allow the manager to allocate between four investment assets: the equal-weighted stock index, the cap-weighted stock index, the 30-day U.S. T-bill index, and the 10-year U.S. T-bond index. We assume that the stock indexes and the 10-year T-bond index are long-only assets. The manager can short the T-bill index to

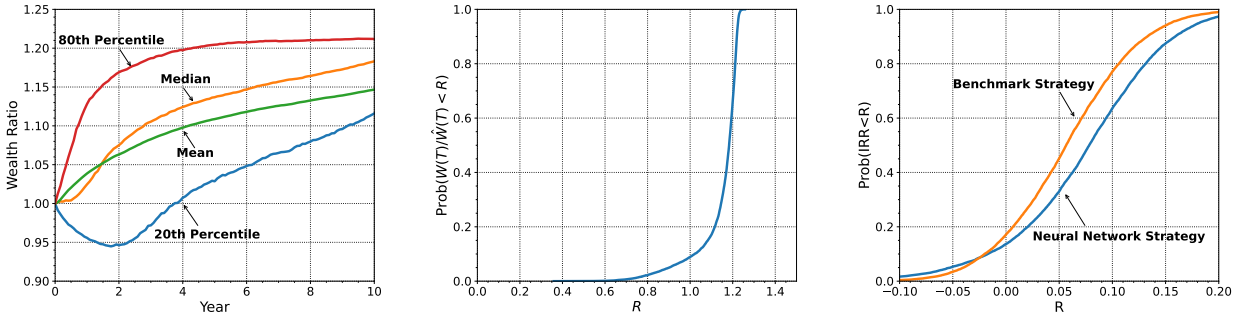
take leverage and invest in the long-only assets (with a maximum total leverage of 1.3, i.e. $p_{max} = 1.3$). The annual outperformance target β is set to be 2% (i.e. 200 bps per year). In this experiment, we assume the borrowing premium rate is zero. Essentially, we assume that the manager can borrow short-term funding to take leverage at the same cost as the treasury bill. This may be a reasonable assumption for sovereign wealth funds, as they are state-owned and enjoy a high credit rating.

It is worth noting that we choose the benchmark portfolio to be a fixed-mix portfolio that maintains a 70% weight in the equal-weighted stock index and 30% in the 30-day U.S. T-bill index. We select this fixed-mix portfolio as the benchmark based on our observation that the equal-weighted stock index shows superior performance compared to the cap-weighted stock index during high inflation environments. As discussed in Section 6.2.3, when analyzing bootstrap resampled data from the historical high inflation regimes, we find that the fixed-mix portfolio consisting of 70% equal-weighted stock index and 30% 30-day U.S. T-bill index partially stochastically dominates the fixed-mix portfolio consisting of 70% cap-weighted stock index and 30% 30-day U.S. T-bill index.

Investment horizon T (years)	10
Equity market indexes	CRSP cap-weighted/equal-weighted index (real)
Bond index	CRSP 30-day/10-year U.S. treasury index (real)
Index samples for bootstrap	Concatenated 1940:8-1951:7, 1968:9-1985:10
Initial portfolio wealth/annual cash injection	100/10
Rebalancing frequency	Monthly
Maximum leverage	1.3
Outperformance target rate β	2%

Table 6.5: Investment scenario.

As discussed in Section 6.2.2, we use the stationary bootstrap resampling algorithm to generate a training data set $\boldsymbol{Y}$ and a testing data set $\boldsymbol{Y}^{test}$ (both with 10,000 resampled paths) from the concatenated index samples from two historical high inflation regimes: 1940:8-1951:7 and 1968:9-1985:10, using an expected blocksize of 6 months. The testing data set $\boldsymbol{Y}^{test}$ is generated using a different random seed as the training data set $\boldsymbol{Y}$, and thus the probability of seeing the same sample in $\boldsymbol{Y}$ and $\boldsymbol{Y}^{test}$ is near zero (the proof has been discussed in Section 2.3). In this chapter, unless stated otherwise, all the results presented are testing results.



(a) Percentiles of wealth ratio $\frac{W(t)}{\hat{W}(t)}$ over time (b) Terminal wealth ratio CDF (c) CDF of IRR

Figure 6.4: Percentiles of wealth ratio $\frac{W(t)}{\hat{W}(t)}$ and CDF of terminal wealth ratio $\frac{W(T)}{\hat{W}(T)}$ and internal rate of return (IRR). Results are based on the evaluation of the learned neural network model on $\boldsymbol{Y}^{test}$.

Strategy	Median$[W_T]$	E$[W_T]$	std$[W_T]$	5th Percentile	Median IRR (annual)
Neural network	364.2	403.4	211.8	136.3	0.078
Benchmark	308.5	342.9	165.0	149.0	0.056

Table 6.6: Statistics of strategies. Results are based on the evaluation results on the testing data set.

The analysis of Figure 6.4a reveals that the neural network strategy (the strategy following the training LFNN model) outperforms the benchmark strategy in terms of the wealth ratio $W(t)/\hat{W}(t)$ with a high probability. Over time, both the mean and median wealth ratios demonstrate a smooth increase. Regarding tail performance (20th percentile), the neural network strategy initially falls behind the benchmark but gradually recovers and ultimately achieves 10% greater wealth at the terminal time. This observation indicates that the neural network strategy effectively manages tail risk.

An additional metric that holds significant interest for managers is the distribution of the terminal wealth ratio $\frac{W(T)}{\hat{W}(T)}$. This metric examines the relative performance of the strategies at the end of the investment period. Figure 6.4b illustrates that there is a greater than 90% chance that the neural network strategy outperforms the benchmark strategy in terms of terminal wealth. This outcome is particularly noteworthy as the objective function (6.2) does not directly target the terminal wealth ratio.

Given the constant cash injections in the portfolios, it is appropriate to employ the internal rate of return (IRR) as a measure of the portfolio's annualized performance. Figure 6.4c demonstrates that the neural network strategy has a more than 90% chance of producing

a higher IRR. Furthermore, the median IRR of the neural network strategy exceeds that of the benchmark strategy by slightly over 2%, aligning with the chosen target outperformance rate of $\beta = 0.02$. This indicates that the neural network model achieves the desired target performance across most outcomes. To put this in perspective, Table 6.6 shows a median annual outperformance of 200 bps.

The results from Table 6.6 indicate that the 5th percentile of the terminal wealth for the neural network strategy is lower than that of the benchmark strategy. This suggests that in some scenarios, particularly during persistent bear markets when stocks perform poorly, the neural network strategy may experience lower terminal wealth compared to the benchmark strategy. The neural network strategy takes on more risk by allocating a higher fraction of wealth to the equal-weighted stock index, which is considered a riskier asset, in comparison to the benchmark portfolio.

It is important to note, however, that these scenarios occur with low probability. As depicted in Figure 6.4b, the neural network strategy exhibits a significantly high probability of outperforming the benchmark in terms of terminal wealth, exceeding 90%. This implies that while there might be instances where the neural network strategy suffers relative to the benchmark, the overall performance is consistently strong, resulting in a high likelihood of achieving superior terminal value.

To gain insight into the strong performance of the neural network strategy, we further examine its allocation profile.

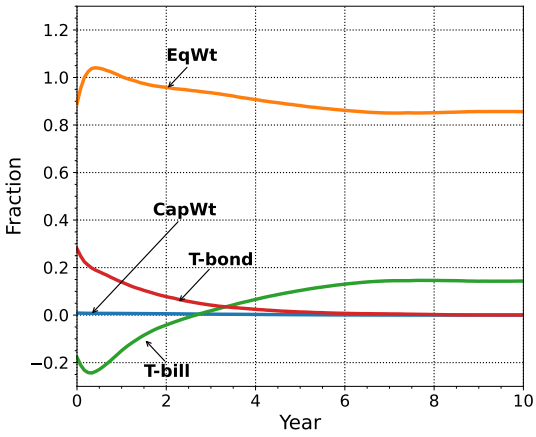


Figure 6.5: Mean allocation fraction over time, evaluated on $\boldsymbol{Y}^{test}$

We begin by examining the mean allocation fraction for the four assets over time, as depicted in Figure 6.5. The first noteworthy observation from Figure 6.5 is that, on average, the neural network strategy does not allocate wealth to the cap-weighted stock index. Initially, this might appear surprising; however, it aligns with historical data indicating significantly higher real returns for the equal-weighted stock index during periods of high inflation. Given

that the objective is to outperform a benchmark heavily invested in the equal-weighted index (70%), it is logical to avoid allocating wealth to a comparatively weaker index in the active portfolio.

The second observation derived from Figure 6.5 pertains to the evolution of mean bond allocation fractions. Initially, the neural network strategy shorts the 30-day T-bill index and assumes some leverage while heavily investing in the equal-weighted stock index and also a small fraction to the T-bond during the first two years. This indicates a deliberate risk-taking approach early on to establish an advantage over the benchmark strategy. Subsequently, the allocation to the 10-year T-bond decreases, coinciding with the reduction in the allocation to the equal-weighted index. This suggests that the initial allocation to the T-bond was primarily for leveraging purposes, with the 10-year bond being the only defensive asset available. As leverage is no longer used in later years, the neural network strategy favors the T-bill over the 10-year bond.

Overall, despite the gradual decrease in stock allocation over time, the neural network strategy maintains an average allocation of more than 80% to the equal-weighted stock index. This is expected, as outperforming an aggressive benchmark with a 70% allocation to the equal-weighted stock index necessitates assuming higher levels of risk. Despite the higher allocation to riskier assets, the neural network strategy delivers better investment results compared to the benchmark strategy with a high probability, as illustrated in Figure 6.4.

6.3.3 Strategy performance in low inflation regimes

Until now, the discussion has been centered around the imaginary scenario of a long, persistent inflation environment. We have shown that the neural network strategy outperforms the benchmark strategy with a high probability under a high inflation regime. However, what if we are wrong? More dramatically, if the high-inflation situation ends immediately and we enter a low-inflation environment, how will the neural network strategy learned under high-inflation regimes perform?

To answer these questions, we evaluate the neural network strategy learned under high-inflation regimes on a testing data set bootstrapped from low-inflation historical regimes. Specifically, we exclude the two inflation regimes (1940:8-1951:7 and 1968:9-1985:10) from the full historical data of 1926:1-2022:1 and obtain several low-inflation data segments. We concatenate the low-inflation data segments and use the stationary bootstrap resampling method to generate a testing data set. We adopt the investment scenario described in Table 6.5 and evaluate the performance of the neural network strategy obtained in Section 6.3.2 on this low-inflation data set.

Note that we continue to use the equal-weighted stock index/30-day T-bill fixed-mix portfolio as the benchmark. This is validated by Figure 6.6, which plots the CDF of the terminal wealth of the fixed-mix portfolios using 70% equal-weighted stock index vs 70% cap-weighted stock index (both with 30% 30-day U.S. T-bill as the bond component). As we

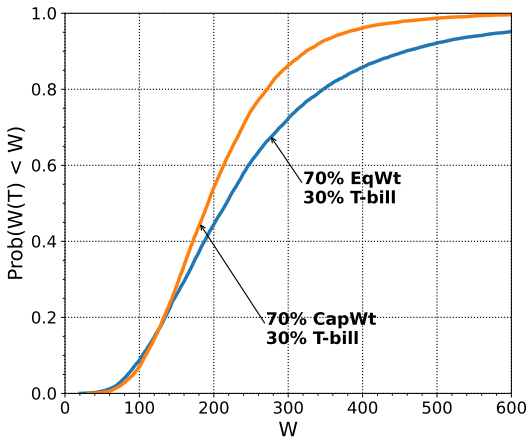
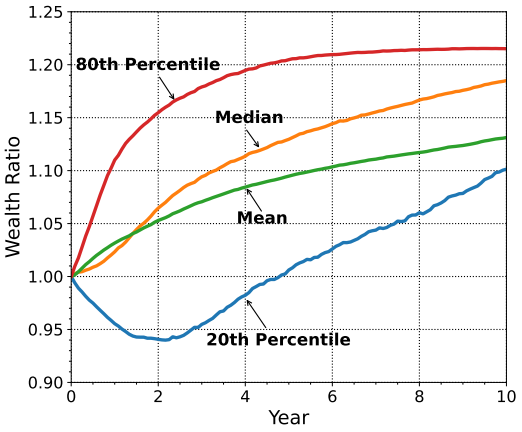
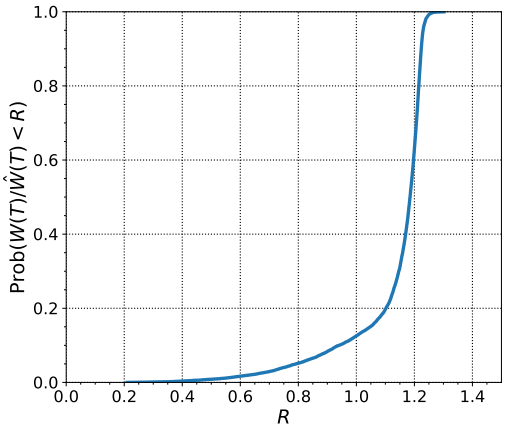


Figure 6.6: Cumulative distribution functions (CDFs) for cap-weighted and equal-weighted indexes, as a function of final real wealth W at $T = 10$ years. Initial stake $W_0 = 100$, no cash injections or withdrawals. Block bootstrap resampling, expected blocksize 6 months. 70% stocks, 30% bonds, rebalanced monthly. Bond index: 30-day U.S. T-bills. Stock index: CRSP capitalization-weighted or CRSP equal-weighted index. Underlying data excludes high inflation regimes. All indexes are deflated by the CPI. $10,000$ resamples. Data set 1926:1-2022:1, excluding high inflation regimes (1940:8-1951:7 and 1968:9-1985:10).

can see from Figure 6.6, the fixed-mix portfolio with an equal-weighted stock index clearly has a more right-skewed distribution than the portfolio with a cap-weighted stock index. This seems to suggest that the equal-weighted index is the superior choice to use in the benchmark portfolio, even in low inflation regimes.



(a) Wealth ratio $\frac{W(t)}{\hat{W}(t)}$



(b) Terminal wealth ratio $\frac{W(T)}{\hat{W}(T)}$

Figure 6.7: Percentiles of wealth ratio over the investment horizon, and CDF of terminal wealth ratio. Results are based on the evaluation of the learned neural network model (from high-inflation data) on the low-inflation testing data set).

Strategy	Median$[W_T]$	E$[W_T]$	std$[W_T]$	5th Percentile	Median IRR (annual)
Neural network	429.7	489.6	301.9	151.6	0.100
Benchmark	368.3	420.8	238.2	175.7	0.079

Table 6.7: Statistic of strategies. Results are based on the evaluation of the learned neural network model (from high-inflation data) on the low-inflation testing data set.

We then present the performance of the neural network strategy learned on high inflation data on the testing data set bootstrapped from low-inflation historical returns. Surprisingly, as we can see from Figure 6.7a, the performance of the neural network strategy learned under high-inflation regimes performs quite well in low-inflation environments. Compared to the testing results on the high inflation data set, there is a noticeable performance degradation; for example, the probability of outperforming the benchmark strategy in terminal wealth is now slightly less than 90%. However, the degradation is quite minimal. The neural network strategy still has more than an 85% chance of outperforming the benchmark strategy at the end of the investment horizon. As shown in Table 6.7, the median IRR of the neural network strategy is still 2% higher than the median IRR of the benchmark strategy, meeting the investment target.

The above results indicate that the neural network strategy is surprisingly robust. Despite being specifically trained under a high-inflation scenario, the strategy performs admirably well in a low-inflation environment.

6.3.4 Experiments with non-zero borrowing premium

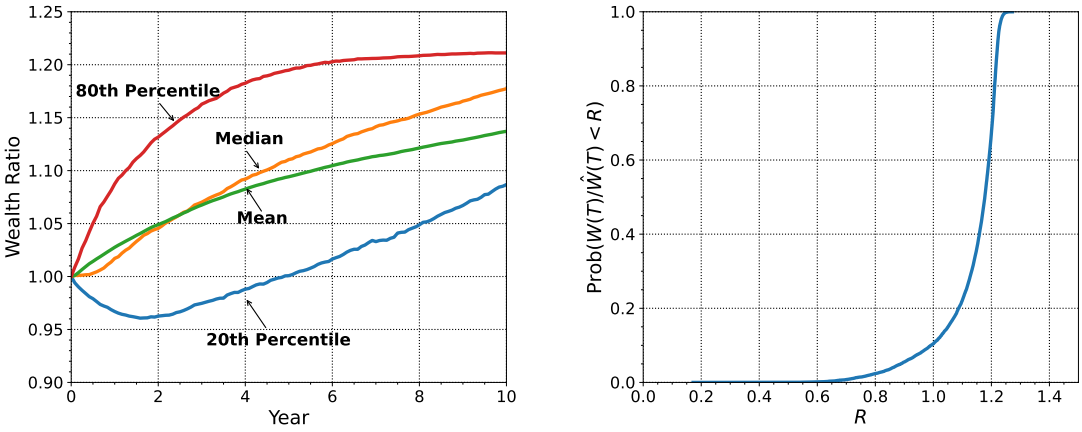
In Section 6.3.2, we conducted the numerical experiments, assuming the borrowing premium is zero. This assumption is based on the fact that large sovereign wealth funds are often considered to have almost risk-free credit ratings, due to their state-backed nature. In other words, we assume that sovereign wealth funds can borrow funding at the same rate as risk-free treasury bills.

This assumption may be too benign for general public funds. In general, it is unlikely that a non-sovereign wealth fund can borrow at a risk-free rate. However, the actual borrowing cost within large public funds is often unavailable to the general public. For this reason, we use the corporate bond yields issued by corporations with similar credit ratings as these large public funds as an approximation to the borrowing cost. Currently, large public funds such as the Blackstone Group or Apollo Global Management are rated between Aaa and Baa rating by Moody's. We obtain the nominal corporate bond yields with Moody's Aaa (Moody's, 2023a) and Baa (Moody's, 2023b) ratings and adjust them with CPI returns. During the two high inflation regimes we have identified, Aaa-rated corporate bonds have an average real yield of 0.7%, while Baa-rated bonds have 1.8%. Taking an average of the two, we use 1.25% as an estimate for the real yield of corporate bonds as well as the borrowing cost of large public funds.[9] As shown in Table 6.2, the average real return for T-bill index is -1.4% during high inflation regimes. This gives us an average borrowing premium rate of 2.65%. In this section, as a stress test, we conduct the same experiment as in Section 6.3.2, except that we use a fixed borrowing premium of 3% (we round 2.65% up) instead of zero. We note that the historical corporate bond yields are based on bonds with a long maturity. Typically, long-term yields are higher than short-term yields, which accounts for the term risk. Therefore, the assumption of a 3% borrowing premium should be a fairly aggressive stress test. We then train and test the neural network strategy with the 3% borrowing premium.

In Figure 6.8, We plot the percentiles of the wealth ratio and the CDF of the terminal wealth ratio. From the percentiles of the wealth ratio, we can observe that the neural network strategy still comfortably outperforms the benchmark strategy throughout the entire investment horizon. At the terminal date, the neural network strategy has a higher than 90% probability of achieving a higher terminal wealth than the benchmark.

In Table 6.8, we list some statistics for the neural network strategy and the benchmark strategy. We can see that the neural network strategy still maintains more than a 200 bps advantage in terms of median IRR compared to the benchmark. In fact, when we compare the results with the neural network strategy obtained under zero borrowing premium (and evaluated without borrowing premium) on the same testing data set, there is little difference in terms of the terminal wealth statistics. The impact of the borrowing premium is so marginal that the median IRR does not change.

[9]Note that the corporate bonds from Moody's yield data have maturities of more than 20 years. Usually, long-term bonds have higher yields than short-term bonds. Thus, using corporate yields likely overestimates the borrowing cost, since we assume the manager is only borrowing short-term funding.



(a) Wealth ratio $\frac{W(t)}{\hat{W}(t)}$ (b) Terminal wealth ratio $\frac{W(T)}{\hat{W}(T)}$

Figure 6.8: Percentiles of wealth ratio over the investment horizon, and CDF of terminal wealth ratio. Annualized borrowing premium is 3%. Results are based on the evaluation of the learned neural network model on the testing data set.

Strategy	Median$[W_T]$	E$[W_T]$	std$[W_T]$	5th Percentile	Annual Median IRR
Neural network	362.7	401.3	212.6	133.9	0.078
Benchmark	308.5	342.9	165.0	149.0	0.056
Neural network (0 premium)	364.2	403.4	211.8	136.3	0.078

Table 6.8: Statistic of strategies. Annualized borrowing premium is 3%. Results are based on the evaluation of the learned neural network model on the testing data set.

The most noticeable impact of the borrowing premium is on the allocation fraction, as shown in Figure 6.9. With a significantly higher borrowing cost, the neural network strategy does not leverage as much in the first two years, resulting in a less negative allocation to the T-bill and a lower allocation to the equal-weighted stock index. However, as we have seen in Figure 6.8 and Table 6.8, this only results in minimal impact on the performance of the strategy.

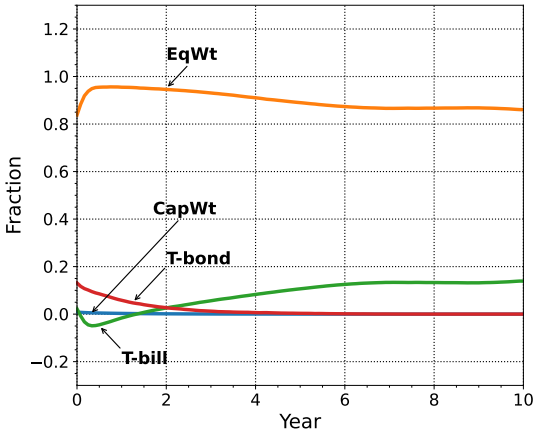


Figure 6.9: Mean of allocation fraction of the learned neural network strategy over time, evaluated on the testing data set. Annualized borrowing premium is 3%. The neural network strategy is learned from bootstrap resampled data based on concatenated series: 1940:8-1951:7, 1968:9-1985:10 (high inflation regimes). The investment scenario is described in Table 6.5.

6.4 Conclusions

In this chapter, our primary objective is to apply the LFNN model to a high-inflation setting and study the optimal portfolio in such scenarios.

We first used a filtering algorithm to discover the two high-inflation segments from the historical data. Furthermore, we compare the passive strategies on bootstrap resampled data from the filtered high inflation segments and find that the equal-weighted stock index is the more favorable stock index in high inflation scenarios.

We then applied the LFNN model to a case study that includes four assets. We find that the cumulative quadratic tracking shortfall (CS) objective function yields more favorable portfolio performance than the cumulative quadratic tracking difference (CD) objective function. Under the CD objective, the neural network strategy achieves the desired outperformance target and outperforms the benchmark portfolio throughout the investment period.

Furthermore, we evaluated the optimal high-inflation strategy in low-inflation environments and found that it still robustly outperforms the benchmark.

Finally, we showed that the borrowing premium only has a minimal impact on strategy performances, validating the assumption of zero borrowing premium in the main case study.

Chapter 7

A neural network model for solving optimal 130/30 portfolios

7.1 Introduction

In Chapter 5, we studied the multi-period portfolio optimization problem with leverage constraints. Specifically, we worked with the assumption that in order to take leverage, the investment manager is only allowed to short within a designated subset of funding securities.

In this chapter, our focus shifts to studying the portfolio optimization problem with a more general constraint. Particularly, we examine long/short strategies that allow the investment manager the flexibility to either long or short any securities across the entire spectrum of underlying securities, with the constraint that the total long position of the portfolio does not exceed a certain limit. Specifically, a strategy that limits its total long position by 130% of the portfolio value, known as the 130/30 strategy, belongs to a class of strategies commonly referred to as 1X0/X0 strategies (X is a digit), where "1X0" means the maximum allowed total long position is 1X0% of the portfolio value.

7.1.1 Rise and fall of 130/30 strategy

1X0/X0 strategies, also known as relaxed constraint (RC) or active extension strategies (Ang et al., 2017), constitute a class of investment strategies that allow portfolio managers to simultaneously hold long and short positions in a diverse universe of securities. Among these strategies, the 130/30 strategy has gained significant popularity and recognition. In the case of the 130/30 strategy, the portfolio exhibits an aggregate long position of up to 130% and a total short position of up to 30%. Notably, alternative variants of 1X0/X0 strategies encompass distinct structures spanning from 120/20 to 150/50.[1] Although the primary focus

[1]The structure is capped at 150/50 due to Federal Reserve Board Regulation T (Federal Reserve Board, 1974).

of this chapter revolves around the 130/30 strategy, the methodological framework proposed herein can readily be applied to other variants within the 1X0/X0 framework.

The 130/30 investment strategy represents a substantial advancement over traditional long-only portfolios by permitting short positions, enabling a more dynamic and sophisticated approach to portfolio management. This strategy is particularly appealing to investors amidst the current environment of global economic uncertainties and a potential regime switch in the financial market. By embracing short positions, the 130/30 strategy offers investors the flexibility to adapt their investment approach and capitalize on both positive and negative market trends, enhancing their ability to navigate uncertainties and potentially enhance investment outcomes.

Shortly before the financial crisis of 2007-2009, the 130/30 strategy gained much attention from the investment community. Numerous studies surrounding the 130/30 idea emerged amidst the increasing popularity (e.g. Johnson et al. (2007); Gastineau (2008); Lo and Patel (2008); Krusen et al. (2008)). In 2007, Tabb Group, a consultancy firm, projected that the size of 130/30 funds would reach $2 trillion by 2010 (Tabb and Johnson, 2007). However, the actual performance of 130/30 funds since their inception was disappointing (Gay, 2012). By the end of 2012, the total identifiable assets of 130/30 funds across Europe and the United States amounted to a mere $9 billion, significantly below the optimistic expectations set in 2007 (Johnson, 2013). Since then, the hype in 130/30 funds has never regained the level of popularity witnessed in 2007. In a recent report by Morningstar, the concept of 130/30 funds is regarded as a "flopped" invention (Rekenthaler, 2022).

Some people argue that the underperformance of 130/30 funds can be attributed to poor timing, as many of these funds were launched just before the onset of the financial crisis of 2007-2009. Consequently, they suffered alongside other active funds that had a net positive exposure to the market. However, reports indicate that even when compared to long-only funds, which the 130/30 funds were designed to replace, the 130/30 funds did not demonstrate superior returns (Johnson, 2013). This is perplexing since 130/30 portfolios theoretically offer a larger solution space than long-only portfolios due to the relaxed constraint. However, empirical evidence fails to support the anticipated performance advantage. Practitioners attribute the poor performance of 130/30 funds to the managers' inability to effectively apply their knowledge of long-only strategies to shorting stocks.

Upon closer examination, it becomes evident that many 130/30 funds, particularly those managed quantitatively, adopted a heuristic stock ranking system as a means to manage their 130/30 portfolios. This ranking system involved assessing all stocks within the investment universe and constructing the 130/30 portfolio by taking long positions in the highest-ranked stocks and short positions in the lowest-ranked stocks (Leibowitz et al., 2009; Korhonen and Kunz, 2010). In comparison to the implementation of a rigorous portfolio optimization framework aimed at maximizing specific utility functions, the use of a stock ranking system appears simplistic and lacking in rigor. Fundamentally, this crude approach may be the underlying cause of the poor performance observed in 130/30 funds.

However, it is important to acknowledge that criticizing 130/30 funds for not adopting

a portfolio optimization approach may be overly harsh. The portfolio optimization problem associated with 130/30 portfolios is inherently more complex than that of long-only portfolios. This complexity arises from the fact that the 130/30 portfolio constraint offers the flexibility to short any asset while also imposing a limit on the total long position. In fact, the literature on the topic of 130/30 portfolio optimization, particularly in the context of multi-period settings, remains scarce even to this day. Consequently, many 130/30 funds had to rely on less rigorous approaches, such as ranking systems, to construct their portfolios. Given the limited knowledge and tools available at the time, it is not surprising that these crude approaches failed to deliver satisfactory results.

7.1.2 Portfolio optimization with 130/30 constraints

In this chapter, we provide a portfolio optimization framework to address the problem of constructing an optimal 130/30 portfolio. Particularly, we formulate a multi-period stochastic optimal control problem with 130/30 portfolio constraints, in which security prices are assumed to be stochastic.

Existing literature related to 130/30 strategies primarily focuses on empirical studies and is not concerned with portfolio optimization. On the other hand, literature in the domain of multi-period portfolio optimization either disregards allocation constraints at all (Zhou and Li, 2000; Li and Ng, 2000) or considers simple constraints such as long-only stock positions with unbounded leverage (Li et al., 2002), with minimal attention given to the unique characteristics of 130/30 strategies. This limited research can be attributed to the non-conventional nature of the total long position constraint imposed by the 130/30 rule, which existing methodologies are ill-equipped to handle.

Given the scarcity of literature on the optimization of 130/30 portfolios, we aim to bridge this gap by providing a novel portfolio optimization framework that addresses the specific challenges posed by the 130/30 constraints. Particularly, following the lines of our work in previous chapters, we propose to use a neural network model to approximate the optimal 130/30 strategy. On a high level, the idea of approximating the optimal control (allocation strategy) in a multi-period portfolio optimization problem is also considered in Han et al. (2016); Tsang and Wong (2020); Reppen et al. (2023); Li and Forsyth (2019); van Staden et al. (2023b); Ni et al. (2022, 2023). However, Han et al. (2016); Tsang and Wong (2020) consider a stacked neural network approach which uses a different subnetwork at every rebalancing time, which is computationally inefficient. On the other hand, Li and Forsyth (2019); Reppen et al. (2023); van Staden et al. (2023b) use a single recurrent neural network for all timesteps, in which time is considered a feature for the network model.

In neural network models for portfolio optimization, it is common to incorporate constraints by designing the model in a way that satisfies the constraints automatically. One common approach is to use the softmax activation function in the last layer of the network, ensuring that the output allocation fractions are non-negative and sum up to one. This technique is widely used in both portfolio optimization with long-only constraints and other

fields such as classification and probabilistic modeling. By formulating the problem as an unconstrained optimization with appropriate activation functions, gradient-based optimization algorithms like SGD can be applied effectively (Buehler et al., 2019).

However, for the 130/30 constraints, which limit the total long position of the portfolio to 130%, designing such a neural network model is more challenging. The closest work is the proposed methodology in Chapter 5, in which we consider the multi-period portfolio optimization problem where the portfolio allows bounded leverage. However, In Chapter 5, we assume that the manager can only short a specific pre-determined subset of the universe of securities, which does not pertain to the 130/30 portfolio setting.

To address this, we propose a novel *relaxed-constraint neural network* (RCNN) model that specifically satisfies the 130/30 constraints. By formulating the problem as an unconstrained optimization with appropriate activation functions, we convert the complex constrained stochastic optimization problem into a more computationally feasible unconstrained problem. Furthermore, we mathematically prove that the RCNN can approximate any optimal 130/30 strategy arbitrarily well, implying that solving the unconstrained optimization problem almost yields the optimal 130/30 strategy.

In practice, 130/30 funds are considered part of the long-only fund family and are evaluated based on their relative performance to a passive benchmark portfolio. We employ a cumulative quadratic shortfall (CS) objective function that measures the tracking performance of the active portfolio against the benchmark. However, the RCNN is flexible and can accommodate various investment objective functions. As long as standard optimization methods can backpropagate through the chosen objective function, our methodology can be applied to a wide range of investment problems with ease.

In our numerical experiments, we utilize the RCNN to investigate the performance of the 130/30 strategy compared to long-only strategies in a four-asset scenario. Our results demonstrate clear advantages of the 130/30 strategy, showcasing superior returns and improved risk management outcomes. This empirical evidence not only validates the effectiveness of our proposed RCNN approach but also highlights the untapped potential of 130/30 portfolios that is yet to be fully realized.

7.1.3 Contributions

In this chapter, we make the following contributions:

(i) We propose a novel relaxed-constraint neural network (RCNN) model for solving the otherwise difficult constrained portfolio optimization problem with 130/30 constraints in an unconstrained optimization fashion, which further illustrates the computational advantage of the neural network methodology proposed in this thesis and its ability to solve complicated practical investment problems.

(ii) We mathematically prove that the RCNN is capable of approximating an optimal 130/30 strategy arbitrarily well. This proof serves as a theoretical foundation, validating the efficacy of our proposed methodology.

(iii) Through numerical experiments, we provide evidence of the advantages of 130/30 portfolios over traditional long-only portfolios. Our findings debunk the misconception prevalent among practitioners that 130/30 portfolios yield few benefits for investors over long-only portfolios.

7.2 Mathematical formulation

In this section, we mathematically formalize the multi-period portfolio optimization problem with 130/30 constraints.

As mentioned in Section 7.1.1, 130/30 funds are considered part of the extended family of long-only funds due to the net long exposure to the market, and are thus assessed against a passive benchmark. Therefore, we consider two portfolios: an active portfolio and a benchmark portfolio. We consider a fixed-horizon investment period of $[t_0, T]$. At time $t \in [t_0, T]$, let $W(t), \hat{W}(t)$ denote the values of the active portfolio and the benchmark portfolio respectively. To ensure a fair assessment of the relative performance of the two portfolios, we assume both portfolios start with an equal initial value $w_0 > 0$, i.e. $W(t_0) = \hat{W}(t_0) = w_0 > 0$.

For simplicity, we assume that both the active portfolio and the benchmark portfolio can allocate among the same set of N_a assets. Let vector $\boldsymbol{S}(t) = (S_i(t) : i = 1, \cdots, N_a)^\top \in \mathbb{R}^{N_a}$ denote the asset prices of the N_a underlying assets at time $t \in [t_0, T]$. In addition, let vectors $p^{(t)} = (p_i^{(t)} : i = 1, \cdots, N_a)^\top \in \mathbb{R}^{N_a}$ and $\hat{p}^{(t)} = (\hat{p}_i^{(t)} : i = 1, \cdots, N_a)^\top \in \mathbb{R}^{N_a}$ denote the allocation fractions to the N_a underlying assets at time $t \in [t_0, T]$, respectively, for the active portfolio and the benchmark portfolio. In this chapter, we consider a passive benchmark portfolio, i.e. $\hat{p}^{(t)} \equiv \hat{\boldsymbol{p}}$, $\forall t \in [0, T]$, where $\hat{\boldsymbol{p}}$ is a constant vector that represents the pre-defined allocation fractions to each asset.

Under an optimal control perspective, the allocation vector $p^{(t)}$ is regarded as the stochastic process denoting the control value at time t (which affects the outcome of the system, i.e. the value evolution of the portfolio), and that p is assumed to be a function of the state variables which are the portfolio values and time. Mathematically, $p^{(t)} = p(\boldsymbol{X}(t)) = (p_i(\boldsymbol{X}(t)) : i = 1, \cdots, N_a)^\top \in \mathbb{R}^{N_a}$, where $\boldsymbol{X}(t) = \left(t, W(t), \hat{W}(t)\right)^\top \in \mathcal{X} \subseteq \mathbb{R}^3$. Our goal is to find the optimal control function p so that the relative performance of the active portfolio over the benchmark portfolio is maximized.

We assume that the active portfolio and the benchmark portfolio follow the same discrete rebalancing schedule denoted by $\mathcal{T} \subseteq [t_0, T]$. Specifically, we consider an equally spaced discrete schedule with N rebalancing events, i.e.

$$\mathcal{T} = \Big\{ t_i : \ i = 0, \cdots, N-1 \Big\}, \tag{7.1}$$

where $t_i = i\Delta t$, and $\Delta t = T/N$.

7.2.1 Feasible strategies under 130/30 constraints

Here we define the feasible strategies under the 130/30 constraints.

Definition 7.2.1. (Feasible strategies under 130/30 constraints). *A strategy $p : \mathcal{X} \mapsto \mathbb{R}^{N_a}$ is a feasible strategy under the 130/30 constraints if and only if*

$$Im(p) \subseteq \mathcal{Z}, \tag{7.2}$$

where $\mathcal{Z} \subset \mathbb{R}^{N_a}$ encodes the 130/30 constraints, i.e. the summation to one constraint and the maximum total long position constraint, as follows.

$$\mathcal{Z} = \left\{ \boldsymbol{z} \in \mathbb{R}^{N_a} \middle| \sum_{i=1}^{N_a} z_i = 1, \sum_{i=1}^{N_a} (z_i)^+ \leq p_{max}, \right\}, \tag{7.3}$$

where $(z_i)^+ = \max(z_i, 0)$ is the positive part of z_i, and $p_{max} = 1.3$ is a constant.

Furthermore, the set of all feasible strategies is denoted by $\mathcal{A}$.

Remark 7.2.1. (Choice of p_{max}). As we focus on the 130/30 strategies in this chapter, we choose $p_{max} = 1.3$ as the maximum total long position of the portfolio. However, the methodology can be readily extended to other 1X0/X0 strategies to accommodate different values of p_{max}. For example, for portfolios with the 150/50 constraints, p_{max} will be 1.5.

7.2.2 Stochastic optimal control problem

In this chapter, we focus on a registered investment fund operating as a limited-liability legal entity (Carney, 1998). This structure is commonly found among investment funds in the United States (Fung and Hsieh, 1999; McCrary, 2004). Limited liability is a crucial characteristic of these funds that restricts investors' liability to the amount they have invested in the fund (Easterbrook and Fischel, 1985). Consequently, investors are protected from personal liability for the fund's debts or obligations beyond their initial investment.

Due to the nature of an active portfolio, which allows for both long and short positions, there is a theoretical possibility for the value of the portfolio to become negative. In such circumstances, the fund would initiate a bankruptcy process, resulting in the settlement of outstanding liabilities and the cessation of future trading activities. From a mathematical perspective, the portfolio value remains at zero throughout the remainder of the investment horizon. Note that this is a different assumption from that used in Chapter 5, where we assumed unlimited liability and debt accumulation. In addition, for simplicity, we do not consider subsequent cash injections after the initial investment. Consequently, the evolution

of the portfolio values can be described as follows from the perspective of an investor in the limited-liability fund:

$$\begin{cases} W(t_{j+1}) = \begin{cases} \Big(\sum\limits_{i=1}^{N_a} p_i(\boldsymbol{X}(t_j)) \cdot \frac{S_i(t_{j+1})-S_i(t_j)}{S_i(t_j)}\Big) W(t_j), & \text{if } W(t_j) \geq 0, \\ 0, & \text{if } W(t_j) < 0, \end{cases} \\ \hat{W}(t_{j+1}) = \Big(\sum\limits_{i=1}^{N_a} \hat{p}_i \cdot \frac{S_i(t_{j+1})-S_i(t_j)}{S_i(t_j)}\Big) \hat{W}(t_j), \end{cases} \qquad \forall j \in \{0, \cdots, N-1\}, \tag{7.4}$$

Let sets $\mathcal{W}_p = \{W(t), t \in \mathcal{T}\}$ and $\hat{\mathcal{W}}_{\hat{p}} = \{\hat{W}(t), t \in \mathcal{T}\}$ represent the trajectories of the portfolio values for the active portfolio and the benchmark portfolio, respectively, following the dynamics specified in equation (7.4). We introduce an investment metric denoted by $F(\mathcal{W}_p, \hat{\mathcal{W}}_{\hat{p}}) \in \mathbb{R}$, which quantifies the relative performance of the active portfolio in relation to the benchmark portfolio based on their respective value trajectories.

In this chapter, we assume the asset prices $\boldsymbol{S}(t) \in \mathbb{R}^{N_a}$ are stochastic. Consequently, the value trajectories $\mathcal{W}_p$, $\hat{\mathcal{W}}_{\hat{p}}$, and the performance metric $F(\mathcal{W}_p, \hat{\mathcal{W}}_{\hat{p}})$ are also stochastic. Therefore, when investment managers aim to optimize an investment metric, the evaluation commonly involves evaluating the expectation of the random metric.

Let $\mathbb{E}_p^{(t_0,w_0)}[F(\mathcal{W}_p, \hat{\mathcal{W}}_{\hat{p}})]$ denote the expectation of the performance metric F, given a specific initial value $w_0 = W(0) = \hat{W}(0)$ at time $t_0 = 0$. The expectation is evaluated on the wealth trajectories following the admissible investment strategies $p \in \mathcal{A}$ and the benchmark investment strategy $\hat{p}$. As we assume the benchmark strategy to be predetermined and known, we keep the benchmark strategy $\hat{p}$ implicit in this notation for simplicity. Subsequently, we try to solve the following stochastic optimization (SO) problem:

$$\text{(Stochastic optimization problem):} \quad \inf_{p \in \mathcal{A}} \mathbb{E}_p^{(t_0,w_0)}\big[F(\mathcal{W}_p, \hat{\mathcal{W}}_{\hat{p}})\big]. \tag{7.5}$$

Solving the constrained stochastic optimal control problem (7.5) is challenging due to the presence of the intricate feasibility constraint (7.2) represented by the set $\mathcal{A}$. In the subsequent section, we propose a neural network methodology that circumvents the complexity of handling this constraint through the introduction of a specially designed relaxed-constraint neural network (RCNN) model.

7.3 Relaxed-constraint neural network (RCNN)

In this section, we propose a neural network methodology for solving the stochastic optimization problem (7.5) with 130/30 constraints. Given the unconventional and complex nature of these constraints, the key concept is to approximate the optimal control function p^* using a neural network function that automatically satisfies the feasibility constraint (7.2).

We first define the following relaxed-constraint activation function.

Definition 7.3.1. (Relaxed-constraint activation function). *Given a constant $\alpha \in \mathbb{R}$, $\alpha \neq \frac{1}{2}$, for any $\boldsymbol{h} = (h_1, \cdots, h_{N_a-1})^\top \in \mathbb{R}^{N_a-1}$, define the "bounded mapping", $\phi_1 : \mathbb{R}^{N_a-1} \mapsto \mathbb{R}^{N_a-1}$, as*

$$\phi_1(\boldsymbol{h}) = (1-\alpha) + (2\alpha - 1) \cdot \sigma(\boldsymbol{h}). \tag{7.6}$$

Here $\sigma : \mathbb{R}^{N_a-1} \mapsto \mathbb{R}^{N_a-1}$ is the pointwise sigmoid function, i.e. $[\sigma(\boldsymbol{h})]_i = \sigma(h_i)$. ϕ_1 maps the unbounded real vector space $\mathbb{R}^{N_a-1}$ into the bounded open set of $(1-\alpha, \alpha)^{N_a-1}$ (if $\alpha > \frac{1}{2}$) or $(\alpha, 1-\alpha)^{N_a-1}$ (if $\alpha < \frac{1}{2}$).[2]

Next, for any $\boldsymbol{u} = (u_1, \cdots, u_{N_a-1})^\top \in \mathbb{R}^{N_a-1}$, define the "extension mapping", $\phi_2 : \mathbb{R}^{N_a-1} \mapsto \mathbb{R}^{N_a}$, as

$$\phi_2(\boldsymbol{u}) = \big(\boldsymbol{u}, 1 - \sum_{i=1}^{N_a-1} u_i\big)^\top. \tag{7.7}$$

ϕ_2 extends a vector from $\mathbb{R}^{N_a-1}$ into a vector in $\mathbb{R}^{N_a}$, which has the property that all entries of this vector sum up to one.

Furthermore, for any $\boldsymbol{v} = (v_1, \cdots, v_{N_a})^\top \in \mathbb{R}^{N_a}$, and a constant $p_{max} > 1$, define the "scaling mapping", $\phi_3 : \mathbb{R}^{N_a} \mapsto \mathbb{R}^{N_a}$, as

$$\phi_3(\boldsymbol{v}) = \begin{cases} \boldsymbol{v}, & \text{if } \sum_{i=1}^{N_a} (v_i)^+ \leq p_{max}, \\ \boldsymbol{v}^+ \cdot \frac{p_{max}}{\sum_{i=1}^{N_a}(v_i)^+} + \boldsymbol{v}^- \cdot \frac{1-p_{max}}{1-\sum_{i=1}^{N_a}(v_i)^+}, & \text{if } \sum_{i=1}^{N_a} (v_i)^+ > p_{max}. \end{cases} \tag{7.8}$$

Here $(v_i)^+ = \max(v_i, 0), \forall i \in \{1, \cdots, N_a\}$. $\boldsymbol{v}^+ = \big(\max(v_1, 0), \cdots, \max(v_{N_a}, 0)\big)^\top \in \mathbb{R}^{N_a}$ and $\boldsymbol{v}^- = \big(\min(v_1, 0), \cdots, \min(v_{N_a}, 0)\big)^\top \in \mathbb{R}^{N_a}$ are the positive and negative parts of vector $\boldsymbol{v}$. ϕ_3 scales any vector in $\mathbb{R}^{N_a}$ so that the sum of all positive entries of the scaled vector is less than or equal to the constant p_{max}.

Finally, define the "relaxed-constraint activation function", $\phi : \mathbb{R}^{N_a-1} \mapsto \mathbb{R}^{N_a}$, as

$$\phi = \phi_3 \circ \phi_2 \circ \phi_1. \tag{7.9}$$

In other words, the relaxed-constraint activation function ϕ is a composition of ϕ_3, ϕ_2 and ϕ_1.

Then, we define the relaxed-constraint neural network (RCNN) function as follows.

Definition 7.3.2. (Relaxed-constraint neural network). *Let $\mathcal{X} \subset \mathbb{R}^3$ be the state space. Given hyperparameters $\big(K, (d_1, \cdots, d_K)^\top\big)$ (i.e. number of hidden layers and their sizes), and parameter $\boldsymbol{\theta}$, let $\tilde{f}_{\boldsymbol{\theta}} : \mathcal{X} \mapsto \mathbb{R}^{N_a-1}$ be the fully connected feedforward neural network (FNN) function parameterized by $\boldsymbol{\theta}$ as defined in Definition 5.3.1. Let $\phi : \mathbb{R}^{N_a-1} \mapsto \mathbb{R}^{N_a}$ be the relaxed-constraint activation function defined in Definition 7.3.1. Then, we define the relaxed-constraint neural network (RCNN) function, $f_{\boldsymbol{\theta}} : \mathcal{X} \mapsto \mathbb{R}^{N_a}$, as*

$$f_{\boldsymbol{\theta}} = \phi \circ \tilde{f}_{\boldsymbol{\theta}}. \tag{7.10}$$

[2]Obviously, if $\alpha = \frac{1}{2}$, then ϕ_1 becomes a trivial constant mapping.

We first establish the following lemma to show that the RCNN function defined in Definition 7.3.2 is a feasible strategy that satisfies the 130/30 constraints.

Lemma 7.3.1. *(Feasibility of RCNN function). For any hyperparameters* $\big(K, (d_1, \cdots, d_K)^\top\big)$ *(i.e. number of hidden layers and their sizes), and parameter* $\boldsymbol{\theta}$*, let* $f_{\boldsymbol{\theta}}$ *be the corresponding RCNN function defined in Definition 7.3.2. Then,* $f_{\boldsymbol{\theta}}$ *is a feasible strategy under the 130/30 constraints, as described in Definition 7.2.1.*

Proof. According to the 130/30 constraints in Definition 7.2.1, it is sufficient to show that

$$Im(f_{\boldsymbol{\theta}}) \subseteq \mathcal{Z}, \tag{7.11}$$

where $\mathcal{Z}$ is the feasibility region defined in (7.3).

Let $\boldsymbol{h} = \tilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $\boldsymbol{y} = (y_1, \cdots, y_{N_a})^\top = f_{\boldsymbol{\theta}}(\boldsymbol{x})$, $\forall \boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^3$, where $\tilde{f}_{\boldsymbol{\theta}}$ is the FNN in Definition 5.3.1. To show (7.11), it is sufficient to show that

$$\boldsymbol{y} \in \mathcal{Z}, \tag{7.12}$$

which is equivalent to

$$\begin{cases} \sum_{i=1}^{N_a} y_i = 1, \\ \sum_{i=1}^{N_a} (y_i)^+ \leq p_{max}. \end{cases} \tag{7.13}$$

Let ϕ_1, ϕ_2 and ϕ_3 be the bounded mapping, extension mapping, and scaling mapping in Definition 7.3.1. Let $\boldsymbol{v} = \phi_2(\phi_1(\boldsymbol{h})) \in \mathbb{R}^{N_a}$. Note that $\sum_{i=1}^{N_a} v_i = 1$ (due to ϕ_2), and $\boldsymbol{y} = \phi_3(\boldsymbol{v})$.

If $\sum_{i=1}^{N_a} (v_i)^+ \leq p_{max}$, then $\boldsymbol{y} = \phi_3(\boldsymbol{v}) = \boldsymbol{v} \in \mathcal{Z}$.

On the other hand, if $\sum_{i=1}^{N_a} (v_i)^+ > p_{max}$, then

$$\boldsymbol{y} = \phi_3(\boldsymbol{v}) = (\boldsymbol{v})^+ \cdot \frac{p_{max}}{\sum_{i=1}^{N_a} (v_i)^+} + (\boldsymbol{v})^- \cdot \frac{1 - p_{max}}{1 - \sum_{i=1}^{N_a} (v_i)^+}. \tag{7.14}$$

Note that $\frac{p_{max}}{\sum_{i=1}^{N_a} (v_i)^+} > 0$, $\frac{1-p_{max}}{1-\sum_{i=1}^{N_a} (v_i)^+} > 0$. Thus,

$$\begin{cases} (\boldsymbol{y})^+ = (\boldsymbol{v})^+ \cdot \frac{p_{max}}{\sum_{i=1}^{N_a} (v_i)^+}, \\ (\boldsymbol{y})^- = (\boldsymbol{v})^- \cdot \frac{1-p_{max}}{1-\sum_{i=1}^{N_a} (v_i)^+}. \end{cases} \tag{7.15}$$

Then, we have

$$\begin{cases} \sum_{i=1}^{N_a} (y_i)^+ = \frac{p_{max}}{\sum_{i=1}^{N_a} (v_i)^+} \cdot \big(\sum_{i=1}^{N_a} (v_i)^+ \big) = p_{max} \leq p_{max}, \\ \sum_{i=1}^{N_a} (y_i)^- = \frac{1-p_{max}}{1-\sum_{i=1}^{N_a} (v_i)^+} \cdot \big(\sum_{i=1}^{N_a} (v_i)^- \big) = \frac{1-p_{max}}{\sum_{i=1}^{N_a} (v_i)^-} \cdot (\sum_{i=1}^{N_a} (v_i)^-) = 1 - p_{max}, \end{cases} \tag{7.16}$$

and

$$\sum_{i=1}^{N_a} y_i = \sum_{i=1}^{N_a} (y_i)^+ + \sum_{i=1}^{N_a} (y_i)^- = 1. \tag{7.17}$$

Therefore, both conditions in (7.13) are satisfied, thus concluding the proof. □

Remark 7.3.1. (Intuition behind the RCNN design). As shown in Definition 7.3.2, the RCNN function is constructed by applying the relaxed-constraint activation function ϕ (Definition 7.3.1) on top of a FNN (Definition 5.3.1). The FNN provides the approximation power by connecting several hidden layers via the sigmoid activation functions. The relaxed-constraint activation function ϕ, on the other hand, guarantees that the RCNN function satisfies the 130/30 constraints. Particularly, recall the three mappings ϕ_1, ϕ_2 and ϕ_3 in Definition 7.3.1. ϕ_1 converts the output of the FNN (which is scattered all over $\mathbb{R}^{N_a-1}$) into a bounded region of $(1-\alpha, \alpha)^{N_a-1}$ (if $\alpha > \frac{1}{2}$) or $(\alpha, 1-\alpha)^{N_a-1}$ (if $\alpha < \frac{1}{2}$). Intuitively, the output of ϕ_1 represents an initial estimate of the allocation fraction for the first $N_a - 1$ assets. Due to the maximum total long position of $p_{max} > 1$, the allocation fraction for each asset falls into $[1 - p_{max}, p_{max}]$. Therefore, we choose α to be slightly larger than p_{max} (in numerical experiments, we use $\alpha = p_{max} + \epsilon$ where $\epsilon = 10^{-6}$ is a small constant), so that $(1-\alpha, \alpha)^{N_a-1}$ tightly covers $[1 - p_{max}, p_{max}]^{N_a-1}$. As we will show in the following lemma, choosing $\alpha > p_{max}$ guarantees the existence of a right inverse of ϕ, which is critical to ensuring the RCNN function can approximate the optimal 130/30 strategy accurately. Subsequently, ϕ_2 guarantees that the summation to one constraint is satisfied, and ϕ_3 guarantees that the maximum total long position constraint is satisfied while preserving the summation to one property obtained from ϕ_2. It is worth noting that without ϕ_1, the RCNN function is still a feasible 130/30 strategy. However, we find that applying ϕ_1 allows faster convergence in the training of the neural network.

As discussed in Remark 7.3.1, we show that ϕ has a right inverse if $\alpha > p_{max}$.

Lemma 7.3.2. *(Existence of right-inverse of ϕ). Given a relaxed constraint activation function $\phi : \mathbb{R}^{N_a-1} \mapsto \mathbb{R}^{N_a}$ as defined in Definition 7.3.1. Let p_{max} be the maximum total long position defined in (7.3). If $\alpha > p_{max}$, then there exists a function $\overrightarrow{\phi} : Im(\phi) \mapsto \mathbb{R}^{N_a-1}$, such that $\overrightarrow{\phi}$ is the right-inverse of ϕ, i.e. $\phi(\overrightarrow{\phi}(\boldsymbol{y})) = \boldsymbol{y}, \forall \boldsymbol{y} \in Im(\phi)$.*

Proof. Let $\boldsymbol{y} = (y_1, \cdots, y_{N_a})^\top \in Im(\phi) \subset \mathbb{R}^{N_a}$. According to Lemma 7.3.1,

$$Im(\phi) \subseteq \mathcal{Z}. \tag{7.18}$$

Therefore, $y_i \in [1 - p_{max}, p_{max}], \forall i \in \{1, \cdots, N_a\}$. Then,

$$\frac{y_i - 1 + \alpha}{2\alpha - 1} \in \Big[\frac{\alpha - p_{max}}{2\alpha - 1}, \frac{\alpha + p_{max} - 1}{2\alpha - 1}\Big] \subset \Big(\frac{0}{2\alpha - 1}, \frac{2\alpha - 1}{2\alpha - 1}\Big) = (0, 1). \tag{7.19}$$

We can then define $\overrightarrow{\phi} : Im(\phi) \mapsto \mathbb{R}^{N_a-1}$ as

$$\overrightarrow{\phi}(\boldsymbol{y}) = \Big(\sigma^{-1}\big(\frac{y_1 - 1 + \alpha}{2\alpha - 1}\big), \cdots, \sigma^{-1}\big(\frac{y_{N_a-1} - 1 + \alpha}{2\alpha - 1}\big)\Big)^\top, \tag{7.20}$$

where $\sigma^{-1}(\cdot)$ is the inverse function of the sigmoid function.

Then, it can be easily verified that $\overrightarrow{\phi}$ is a right-inverse of ϕ, i.e. for all $\boldsymbol{y} \in Im(\phi)$

$$\phi\big(\overrightarrow{\phi}(\boldsymbol{y})\big) = \boldsymbol{y}. \tag{7.21}$$

□

Denote the wealth trajectory following $f_{\boldsymbol{\theta}}$ by $\mathcal{W}_{\boldsymbol{\theta}}$. Then, the original optimization problem (7.5) constrained by the 130/30 rule is converted into the following unconstrained optimization problem:

$$\text{(Unconstrained parameterized problem):} \quad \inf_{\boldsymbol{\theta}\in\mathbb{R}^{N_{\boldsymbol{\theta}}}} \mathbb{E}_{f_{\boldsymbol{\theta}}}^{(t_0,w_0)}\big[F(\mathcal{W}_{\boldsymbol{\theta}}, \hat{\mathcal{W}}_{\hat{p}})\big]. \tag{7.22}$$

However, an essential question remains unanswered: for the optimal 130/30 strategy p^*, can we find a selection of the hyperparameters $\Big(K, (d_1, \cdots, d_K)^\top\Big)$ and parameter $\boldsymbol{\theta}$ so that the corresponding RCNN function $f_{\boldsymbol{\theta}}$ can be arbitrarily close to p^*? If the answer is affirmative, it assures that solving the unconstrained problem (7.22) yields a good approximation of the optimal 130/30 strategy. To address this crucial question, we establish an approximation theorem in the following section, providing a formal proof of the existence of such approximations. This theorem will justify the theoretical effectiveness of the neural network methodology for approximating the optimal 130/30 strategy.

7.4 Universal approximation theorem for RCNN

Before we prove the approximation theorem for the RCNN, we first present some reasonable assumptions necessary for the proceeding of the proof.

Assumption 7.4.1. *(Assumption on state space and optimal control).*

(i) The state space $\mathcal{X}$ is a compact set.

(ii) The optimal control $p^* : \mathcal{X} \mapsto \mathcal{Z}$ is continuous.

Remark 7.4.1. (Remark on Assumption 7.4.1). In our particular problem of outperforming a benchmark portfolio, the state variable vector is $X(t) = (t, W(t), \hat{W}(t))^\top \in \mathcal{X}$ where $t \in [0, T]$. In this case, assumption (i) is equivalent to the assumption that the wealth of the active portfolio and benchmark portfolio is bounded, i.e. $\mathcal{X} = [0, T] \times [0, w_{max}] \times [0, \hat{w}_{max}]$, where w_{max} and $\hat{w}_{max}$ are the respective upper bounds for the portfolio values. Assumption (ii) assumes that the optimal control is a continuous function, which is common and intuitive.

We then proceed to present the approximation theorem for the RCNN.

Theorem 7.4.1. *(Approximation of optimal 130/30 strategy). Assume the constant α in Definition 7.3.1 satisfies $\alpha > p_{max}$. Given the optimal control p^* and following Assumption 7.4.1, $\forall \epsilon > 0$, there exists $\left(K, (d_1, \cdots, d_K)^\top\right)$, and $\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}$ such that the corresponding RCNN $f_{\boldsymbol{\theta}}$ described in Definition (7.3.2) satisfies the following:*

$$\sup_{x \in \mathcal{X}} \|f_{\boldsymbol{\theta}}(x) - p^*(x)\| < \epsilon. \tag{7.23}$$

Proof. Let ϕ be the relaxed constraint activation function in Definition 7.3.1. According to Lemma 7.3.1,

$$Im(\phi) \subseteq \mathcal{Z}. \tag{7.24}$$

Furthermore, $\forall \boldsymbol{z} = (z_1, \cdots, z_{N_a})^\top \in \mathcal{Z}$, we know that $z_i \in [1 - p_{max}, p_{max}], \forall i \in \{1, \cdots, N_a\}$. For the same reason as (7.19), $\overrightarrow{\phi}(\boldsymbol{z})$ is well-defined and $\phi(\overrightarrow{\phi}(\boldsymbol{z})) = \boldsymbol{z}$. Therefore,

$$\mathcal{Z} \subseteq Im(\phi). \tag{7.25}$$

Combine (7.25) with (7.24), we know that $Im(\phi) = \mathcal{Z}$, and thus $Im(\phi)$ is dense in $\mathcal{Z}$.

In addition, ϕ is continuous and has a right-inverse, according to Lemma 7.3.2.

Then, following Lemma 5.3.3, we know that there exists $\left(K, (d_1, \cdots, d_K)^\top\right)$, and $\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}$, such that the corresponding FNN $\tilde{f}_{\boldsymbol{\theta}}$ (Definition 5.3.1) and RCNN $f_{\boldsymbol{\theta}} = \phi \circ \tilde{f}_{\boldsymbol{\theta}}$ satisfy

$$\sup_{x \in \mathcal{X}} \|f_{\boldsymbol{\theta}}(x) - p^*(x)\| = \sup_{x \in \mathcal{X}} \|\phi\big(\tilde{f}_{\boldsymbol{\theta}}(x)\big) - p^*(x)\| < \epsilon. \tag{7.26}$$

□

Remark 7.4.2. (Implication of Theorem 7.4.1). Theorem 7.4.1 provides valuable insight that, for any feasible control that satisfies the 130/30 constraints, there exists a selection of hyperparameters and parameter values that enables the corresponding RCNN to approximate the control arbitrarily well. Intuitively, when the RCNN is sufficiently large in terms of the number of hidden nodes, solving the unconstrained optimization problem (7.22) results in an approximate solution that closely approximates the optimal control p^*.

7.5 Numerical experiments

In this section, we perform numerical experiments based on the RCNN method. We adopt a shallow neural network structure with a single hidden layer consisting of 10 hidden nodes ($K = 1$ and $d_1 = 10$). As input (feature) to the RCNN network, we utilize a 3-tuple vector $(t, W_{\boldsymbol{\theta}}(t), \hat{W}(t))^\top$. Here, at time $t \in [t_0, T]$, $W_{\boldsymbol{\theta}}(t)$ represents the wealth of the active portfolio governed by the RCNN parameterized by $\boldsymbol{\theta}$, while $\hat{W}(t)$ represents the wealth of

the benchmark portfolio. The procedure for training the neural network model is described in Chapter 5.3.3.

We use the U.S. monthly data from the Center for Research in Security Prices (CRSP)[3] from January 1926 to January 2023 as the original dataset for experiments. Particularly, we obtain the real historical returns of the equal-weighted/cap-weighted U.S. stock indexes and 10-year/30-day treasury indexes by adjusting for the CPI index.

Similar to the numerical experiments in the previous chapters, we use the stationary block bootstrap resampling method (Politis and Romano, 1994) to generate the training data set $\boldsymbol{Y}$ and testing data set $\boldsymbol{Y}^{test}$. Both data sets consist of 20,000 resampled paths. The pseudo-code for the bootstrap algorithm can be found in Algorithm 2.3.1.

Same as in Chapter 6, in the objective function, we use the cumulative quadratic shortfall (CS) metric:

$$F(\mathcal{W}_p, \hat{\mathcal{W}}_{\hat{p}}) = \sum_{t\in\mathcal{T}} \Big(\min\big(W_{\boldsymbol{\theta}}(t) - e^{\beta t}\hat{W}(t), 0\big)\Big)^2 + \epsilon W_{\boldsymbol{\theta}}(T). \tag{7.27}$$

In the CS metric, the parameter β represents a predefined outperformance target, and the regularization term ϵ is a small positive constant.[4] The CS metric quantifies the underperformance of the active portfolio relative to an elevated target $e^{\beta t}\hat{W}(t)$. Notably, the CS objective captures the intermediate tracking performance of the portfolio, and the parameter β provides a clear performance goal. For a more detailed discussion on comparisons of objective functions, see Section 6.3.1.

$$\inf_{\boldsymbol{\theta}\in\mathbb{R}^{N_{\boldsymbol{\theta}}}} \mathbb{E}_{f_{\boldsymbol{\theta}}}^{(t_0,w_0)}\left[\sum_{t\in\mathcal{T}} \Big(\min\big(W_{\boldsymbol{\theta}}(t) - e^{\beta t}\hat{W}(t), 0\big)\Big)^2 + \epsilon W_{\boldsymbol{\theta}}(T)\right]. \tag{7.28}$$

Finally, it is worth mentioning that the proposed neural network methodology is agnostic to the choice of the objective function and can be applied to a broad range of performance metrics.

7.5.1 Experiment results

In this section, we present a case study focusing on optimal asset allocation with 130/30 constraints. The objective of the study is to assess the potential advantages offered by a 130/30 portfolio compared to long-only portfolios. Specifically, we analyze and compare the

[3] ©2023 Center for Research in Security Prices (CRSP), The University of Chicago Booth School of Business. Wharton Research Data Services (WRDS) was used in preparing this chapter. This service and the data available thereon constitute valuable intellectual property and trade secrets of WRDS and/or its third-party suppliers.

[4] Same as in Chapter 6, we use $\epsilon = 10^{-6}$ in experiments.

performance of the optimal 130/30 portfolio, as determined by the RCNN, with the optimal long-only portfolio generated by the state-of-the-art model for long-only constraints (Li and Forsyth, 2019; Ni et al., 2022), using the same investment scenario.

The investment scenario is outlined in Table 7.1. In summary, both the active portfolio and the benchmark portfolio commence with an initial wealth of 100 at time $t_0 = 0$. Monthly rebalancing is implemented for both portfolios over a 10-year investment horizon. Our empirical case allows the investment manager to allocate among four distinct investment assets: the equal-weighted stock index, the cap-weighted stock index, the 30-day U.S. T-bill index, and the 10-year U.S. T-bond index.

Same as in Chapter 6, we choose the benchmark portfolio to be a constant weight portfolio with 70% in the equal-weighted stock and 30% in the 30-day T-bill index. By varying the outperformance target rates (β) across the range of 0.5% to 5% (incremental with a 0.5% step size), we obtain the corresponding optimal portfolios through the cumulative quadratic shortfall (CS) objective.

Investment horizon T (years)	10
Underlying assets	CRSP cap-weighted/equal-weighted index (real) CRSP 30-day/10-year U.S. treasury index (real)
Index samples for bootstrap	1926/01 to 2023/01
Initial portfolio wealth	100
Rebalancing frequency	Monthly
Cash injections	0
Benchmark portfolio	70% equal-weighted index/30% 30-day T-bill
Performance metric	Cumulative quadratic shortfall (CS)
Outperformance target rate β	0.5% - 5%, incremental by 0.5%

Table 7.1: Investment scenario.

In Table 7.2, we present the capabilities of the 130/30 portfolio and the long-only portfolio for tracking elevated targets. Particularly, the tracking performance is reflected in the value of the CS objective, which measures the cumulative quadratic shortfall with respect to elevated targets defined by the target rate β.

β	0.5%	1%	1.5%	2%	2.5%	3%	3.5%	4%	4.5%	5%
130/30	275	402	622	985	1491	2434	3427	4782	6518	8709
Long-only	280	432	708	1150	1841	2855	4547	6430	9041	12594

Table 7.2: CS objective function values for various β, lower is better. The results are based on the performance of trained models evaluated on $\boldsymbol{Y}_{test}$.

As we can see from Table 7.2, even though the optimal long-only portfolio is obtained

under the same investment scenario and optimized under the same objective function, it achieves significantly worse tracking performance than the optimal 130/30 portfolio.

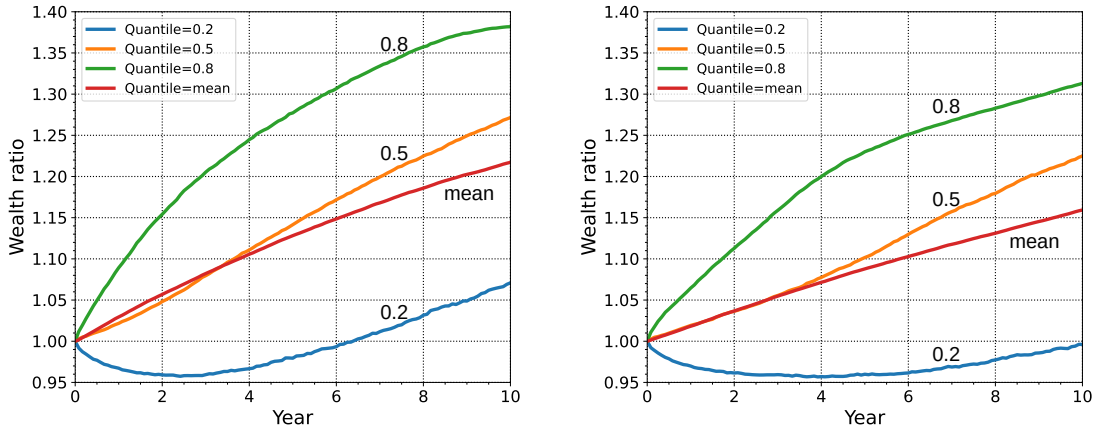
Particularly, the gap between the CS objective function values widens as the target outperformance rate β increases, indicating that the long-only portfolio is further restricted by the long-only constraints as the outperformance target becomes more ambitious. This phenomenon is further demonstrated in Table 7.3, in which we list the median annual return of both portfolios.

β	0.5%	1%	1.5%	2%	2.5%	3%	3.5%	4%	4.5%	5%
130/30	7.2%	7.6%	8.1%	8.5%	8.9%	9.4%	9.7%	10.0%	10.2%	10.5%
Long-only	7.2%	7.6%	8.1%	8.5%	8.8%	8.9%	8.9%	8.9%	8.9%	8.9%

Table 7.3: Median annualized returns. The benchmark portfolio has a median annualized return of 6.7%. The results are based on the performance of trained models evaluated on $\boldsymbol{Y}_{test}$.

As we can see from Table 7.3, when β is modest ($< 3\%$), the long-only portfolio shows similar median returns as the 130/30 portfolio (despite that the objective function value is slightly worse). However, as β becomes more ambitious ($\geq 3\%$), the long-only portfolio has a harder time keeping up with the 130/30 portfolio. Specifically, we can see that the median return of the long-only portfolio stagnates for $\beta \geq 3\%$. As we will discuss shortly, at $\beta = 3\%$, the optimal long-only portfolio is already allocating almost 100% allocation to the equal-weighted stock index, the riskiest asset with the highest expected return. There is little room for the long-only portfolio to take more risks due to the long-only constraint to meet the more aggressive β values. On the other hand, we can see that the median return of the optimal 130/30 portfolio increases alongside β.

In the following, for $\beta = 3\%$, we plot the quantiles of the wealth ratio $\frac{W(t)}{\tilde{W}(t)}$, which measures the relative performance of the active portfolio with respect to the benchmark portfolio throughout the investment horizon.



(a) 130/30 portfolio

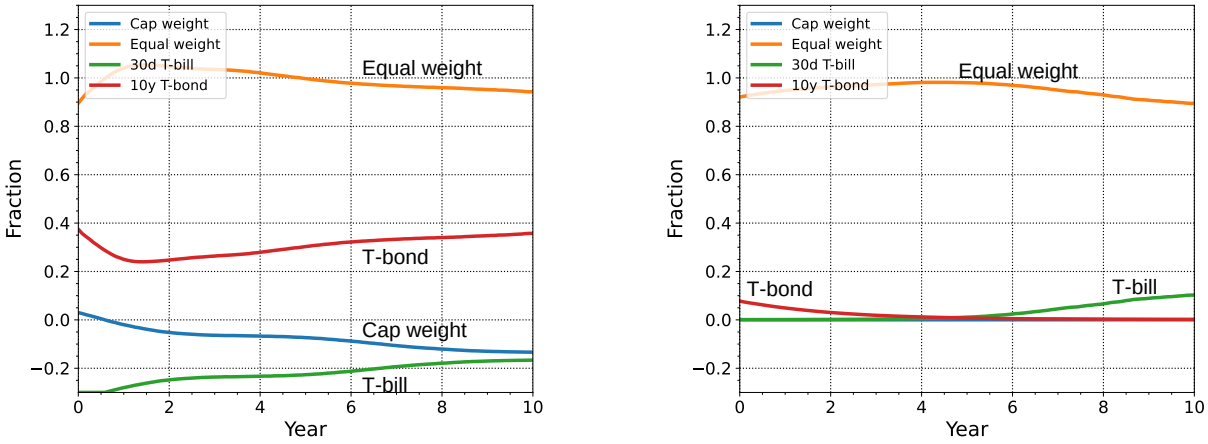
(b) Long-only portfolio

Figure 7.1: Quantiles of wealth ratio over the investment horizon $[0, T]$. $\beta = 3\%$. The 130/30 portfolio follows the RCNN trained on $\boldsymbol{Y}$. The long-only portfolio follows the neural network model from (Li and Forsyth, 2019; Ni et al., 2022) trained on $\boldsymbol{Y}$. Results in the plots are testing results evaluated on $\boldsymbol{Y}_{test}$.

Based on the results and analysis presented in Figure 7.1, it is evident that the 130/30 portfolio outperforms the long-only portfolio across various quantiles. The wealth ratios of the 130/30 portfolio consistently exceed those of the long-only portfolio, indicating superior performance.

Unsurprisingly, the superior performance of the 130/30 portfolio can be attributed to its relaxed portfolio constraints. We plot the median allocation fractions of the 130/30 portfolio and long-only portfolio in Figure 7.2. We can see from Figure 7.2a that the optimal 130/30 portfolio strategically leverages its position by exceeding 100% exposure to the equal-weighted stock index in the first half of the investment period. Interestingly, the 130/30 portfolio longs the equal-weighted stock and the long-term bond, and shorts the cap-weighted stock and the short-term bond, creating long/short patterns within both asset classes (i.e. stock and bond). On the other hand, as observed from Figure 7.2b, the long-only portfolio is obviously restricted by the long-only constraint. It yields an almost trivial strategy that has a close to 100% allocation to the equal-weighted stock index throughout the investment horizon.

Of course, we remark that there is no free lunch, and the 130/30 strategy achieves superior results with some compromises. Particularly, if we examine the extreme tail statistics such as the 1% CVaR of the terminal wealth (i.e. the average of the lowest 1% of the terminal wealth), we can see that the 130/30 portfolios have slightly worse results than the long-only portfolios, as shown in Table 7.4. This is because the 130/30 portfolios are leveraged and exposed to greater market risk and thus perform worse in consistent bear market scenarios.



(a) 130/30 portfolio

(b) Long-only portfolio

Figure 7.2: Median allocation fractions over the investment horizon $[0, T]$. $\beta = 3\%$. The 130/30 portfolio follows the RCNN trained on $\boldsymbol{Y}$. The long-only portfolio follows the neural network model from (Li and Forsyth, 2019; Ni et al., 2022) trained on $\boldsymbol{Y}$. Results in the plots are testing results evaluated on $\boldsymbol{Y}_{test}$.

β	0.5%	1%	1.5%	2%	2.5%	3%	3.5%	4%	4.5%	5%
130/30	44	38	32	29	27	25	24	23	22	22
Long-only	44	39	36	35	34	33	32	32	32	32

Table 7.4: 1% CVaR of terminal wealth, higher is better. The results are based on the performance of trained models evaluated on $\boldsymbol{Y}_{test}$.

However, one cannot simply conclude that 130/30 portfolios are riskier than long-only portfolios. If we look at the 20th quantile of the wealth ratio in Figure 7.1, we can observe that the optimal 130/30 portfolio exhibits better wealth ratios compared to the optimal long-only portfolio. This suggests that the 130/30 portfolio is capable of mitigating downside risks well in most scenarios.

Overall, the numerical experiments illustrate the superiority of the 130/30 portfolio under the tracking performance-based investment objective. The 130/30 portfolio not only achieves more ambitious returns but also demonstrates good risk management. This can be attributed to the broader range of portfolio strategies available within the 130/30 structure, which allows for more flexibility and potential for generating excess return.

In summary, this example also illustrates the effectiveness of the RCNN methodology. It is not necessary to determine a priori of which assets can be shorted. The optimal control solution will find the most effective strategy.

7.6 Conclusions

In this chapter, we introduced a neural network-based solution for the portfolio optimization problem with 130/30 constraints. By formulating the problem as a multi-period stochastic optimal control problem, we proposed a novel relaxed-constraint neural network (RCNN) model to approximate the optimal control.

The RCNN addresses the complexity of the original constrained optimization problem by converting it into an unconstrained problem that can be solved efficiently. We provided mathematical proof demonstrating that the RCNN can accurately approximate any leverage-constrained strategy. In particular, in contrast to the method in Chapter 5, we do not need to specify a subset of assets that can be shorted.

In numerical experiments, we compared the optimal 130/30 portfolio following the RCNN with the optimal long-only portfolio under the same investment scenarios and the investment objective. The results consistently indicate that the optimal 130/30 portfolio outperforms the long-only portfolio in terms of optimizing the investment objective.

We believe the methodology developed in this chapter can be applied to investment problems of widespread interest, such as finding optimal portfolios of factor ETFs (Glushkov, 2015). In addition, it is worth considering other types of securities such as options in the portfolio (Andersson and Oosterlee, 2023), which may yield better practical results.

Chapter 8

Conclusions

This thesis introduced a neural network-based approach for solving multi-period portfolio optimization problems with diverse constraints. By formulating the portfolio optimization problem as a stochastic optimal control problem, the proposed method utilizes neural network models to approximate optimal control. This enables efficient computation of optimal portfolios under realistic constraints such as discrete rebalancing and no shorting, accommodating various objective functions that existing methods often struggle to handle.

In Chapters 2 and 3, we applied the methodology to obtain the optimal portfolios that outperform stochastic benchmark targets. Specifically, we proposed a novel asymmetric objective function based on the concept of the elevated target, guiding the neural network model to achieve the desired relative performance over the benchmark. In numerical experiments, we adopted the stationary block bootstrap resampling algorithm to generate the training and testing data sets from the underlying historical data path. To justify the validity of using bootstrap resampling for generating the data sets, we established theorems demonstrating that the probability of observing identical paths in the training and testing data sets is exceedingly low. In numerical experiments, we showed that the adaptive strategy following a single shallow neural network model achieves desirable outperformance over the benchmark target, illustrating the effectiveness and efficiency of the proposed methodology.

In Chapter 4, we derived the closed-form solution for the multi-period portfolio optimization problem under the cumulative quadratic tracking difference (CD) objective function. While the solution relies on certain assumptions, such as continuous rebalancing, infinite leverage, and continuation of trading in bankruptcy, it yields key insights into optimal control behavior. In particular, we found that the optimal control is contrarian, and aims higher than the elevated target in allocation decision-making. To accommodate realistic constraints, we proposed an approximate form of the closed-form solutions.

In Chapter 5 and 6, we studied the multi-period portfolio optimization problem in which the portfolio is subject to a capped-leverage constraint, which is complex and unorthodox. We proposed a novel leverage-feasible neural network (LFNN) that converts the otherwise complex optimization problem into an unconstrained optimization problem. Furthermore,

we established an approximation theorem to ensure that the proposed neural network model is capable of approximating the optimal control arbitrarily well. In numerical experiments, we considered a case study of high-inflation investment and demonstrated that the LFNN model can yield a strategy that achieves the desired outperformance over the benchmark throughout the entire investment horizon.

In Chapter 7, we extended the methodology to multi-period portfolio optimization problems with more general constraints. In particular, we considered a leveraged portfolio that is not restricted to specific subsets of securities for shorting. There is little literature on this topic, despite its empirical importance. By proposing a novel relaxed-constraint neural network (RCNN) model, we showed that the complex portfolio optimization problem can be transformed into a computationally feasible unconstrained optimization problem. Furthermore, we mathematically proved that the RCNN model can approximate the optimal control arbitrarily well. Finally, in numerical experiments, we compared the relative performance over the market portfolio for the optimal 130/30 strategy and the optimal long-only strategy obtained under the same investment objective function and found that the optimal 130/30 strategy yields superior investment outcomes. This result challenges the misconception that 130/30 portfolios offer little advantage over long-only portfolios.

In terms of future work, there are two immediate trajectories we can explore.

Firstly, the application of the RCNN model could be extended to construct optimal long/short portfolios of factor-based ETFs, an area of significant interest for investors seeking enhanced returns through more sophisticated strategies.

Additionally, the methodology could be extended to address complex allocation constraints in portfolios. For instance, portfolios with dynamic risk limits on exposure to specific market factors or underlying assets pose otherwise difficult challenges that our methodology could adeptly tackle, as elaborated in Remark 5.3.3. These avenues hold promise for demonstrating the applicability and effectiveness of our proposed approach.

References

Al-Aradi, A. and S. Jaimungal (2018). Outperformance and tracking: Dynamic asset allocation for active and passive portfolio management. *Applied Mathematical Finance 25*(3), 268–294.

Al-Aradi, A. and S. Jaimungal (2021). Active and passive portfolio management with latent factors. *Quantitative Finance 21*(9), 1437–1459.

Alexander, S., T. F. Coleman, and Y. Li (2006). Minimizing CVaR and VaR for a portfolio of derivatives. *Journal of Banking & Finance 30*(2), 583–605.

Ambachtsheer, K. P. (1987). Pension fund asset allocation: In defense of a 60/40 equity/debt asset mix. *Financial Analysts Journal 43*(5), 14–24.

Anarkulova, A., S. Cederburg, and M. S. O'Doherty (2022). Stocks for the long run? evidence from a broad sample of developed markets. *Journal of Financial Economics 143*(1), 409–433.

Andersson, K. and C. W. Oosterlee (2023). D-tipo: Deep time-inconsistent portfolio optimization with stocks and options. *arXiv preprint arXiv:2308.10556*.

Ang, A., D. Papanikolaou, and M. M. Westerfield (2014). Portfolio choice with illiquid assets. *Management Science 60*(11), 2737–2761.

Ang, I., A. Michalka, and A. Ross (2017). Understanding relaxed constraint equity strategies. Technical report, Working Paper, AQR Capital.

AQR (2021). Capital market assumptions for major asset classes. `https://www.aqr.com/Insights/Research/Alternative-Thinking/2021-Capital-Markets-Assumptions-for-Major-Asset-Classes`. [Online; accessed 31-Aug-2023].

Arnott, R. D., K. F. Sherrerd, and L. Wu (2013). The glidepath illusion and potential solutions. *The Journal of Retirement 1*(2), 13–28.

Atkinson, A. B. (1987). On the measurment of poverty. *Econometrica 55*(4), 749–764.

Bajeux-Besnainou, I., R. Portait, and G. Tergny (2013). Optimal portfolio allocations with tracking error volatility and stochastic hedging constraints. *Quantitative Finance 13*(10), 1599–1612.

Balduzzi, P. and J. Reuter (2012). *Heterogeneity in target-date funds and the pension protection act of 2006*. National Bureau of Economic Research. Report RRC NB11-02.

Ball, L. M., D. Leigh, and P. Mishra (2022). Understanding us inflation during the covid era. Technical report, National Bureau of Economic Research.

Banz, R. W. (1981). The relationship between return and market value of common stocks. *Journal of financial economics 9*(1), 3–18.

Basak, S., A. Shapiro, and L. Tepla (2006). Risk management with benchmarking. *Management Science 52(4)*, 542–557.

Basu, A. K., A. Byrne, and M. E. Drew (2011). Dynamic lifecycle strategies for target date retirement funds. *Journal of Portfolio Management 37*(2), 83–96.

Basu, A. K. and O. K. Wiafe (2017). Impact of persistent bad returns and volatility on retirement outcomes. *Finance Research Letters 21*, 201–205.

Bengen, W. P. (1994). Determining withdrawal rates using historical data. *Journal of Financial planning 7*(4), 171–180.

Bernstein, P. L. (2002). The 60/40 solution. *Bloomberg Personal Finance*, 28–30.

Biggs, A. G. and G. R. Springstead (2008). Alternate measures of replacement rates for social security benefits and retirement income. *Socal Security Bulletin 68*, 1.

Black, F. (1993). Estimating expected return. *Financial Analysts Journal 49*(5), 36–38.

Blanchet-Scalliet, C., N. El Karoui, M. Jeanblanc, and L. Martellini (2008). Optimal investment decisions when time-horizon is uncertain. *Journal of Mathematical Economics 44*(11), 1100–1113.

Blanchett, D., M. Finke, and W. Pfau (2018). Low returns and optimal retirement savings. In O. S. Mithcell, R. Clark, and R. Maurer (Eds.), *How Persistent Low Returns Will Shape Saving and Retirement*, pp. 26–43. Elsevier.

Blanchett, D., M. S. Finke, and W. D. Pfau (2017). Planning for a more expensive retirement. *Journal of Financial Planning 30*(3), 42–51.

Bo, L., H. Liao, and X. Yu (2021). Optimal tracking portfolio with a ratcheting capital benchmark. *SIAM Journal on Control and Optimization 59*(3), 2346–2380.

Booth, L. (2004). Formulating retirement targets and the impact of time horizon on asset allocation. *Financial Services Review 13*(1), 1–18.

Brown, D. C., S. Cederburg, and M. S. O'Doherty (2017). Tax uncertainty and retirement savings diversification. *Journal of Financial Economics 126*(3), 689–712.

Browne, S. (1997). Survival and growth with a liability: Optimal portfolio strategies in continuous time. *Mathematics of Operations Research 22*(2), 468–493.

Browne, S. (1999). Beating a moving target: optimal portfolio strategies for outperforming a stochastic benchmark. *Finance and Stochastics 3*, 275–294.

Browne, S. (2000). Risk-constrained dynamic active portfolio management. *Management Science 46*(9), 1188–1199.

Buehler, H., L. Gonon, J. Teichmann, and B. Wood (2019). Deep hedging. *Quantitative Finance 19:8*, 1271–1291.

Cao, H. K., H. K. Cao, and B. T. Nguyen (2020). Delafo: An efficient portfolio optimization using deep neural networks. In *Advances in Knowledge Discovery and Data Mining: 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11–14, 2020, Proceedings, Part I 24*, pp. 623–635. Springer.

Cariño, D. R. and A. L. Turner (1998). Multiperiod asset allocation with derivative assets. In W. T. Ziemba and J. M. Mulvey (Eds.), *Worldwide Asset and Liability Modeling*, pp. 129–151. Cambridge University Press Cambridge.

Carney, W. J. (1998). Limited liability. *Encyclopedia of law and economics, Available at SSRN 50563*.

Cheung, K. C. and H. Yang (2004). Asset allocation with regime-switching: discrete-time case. *ASTIN Bulletin: The Journal of the IAA 34*(1), 99–111.

Coache, A. and S. Jaimungal (2021). Reinforcement learning with dynamic convex risk measures. *arXiv preprint arXiv:2112.13414*.

Cocco, J. F., F. J. Goems, and P. J. Maenhout (2005). Consumption and portfolio choice over the life cycle. *Review of Financial Studies 18*, 491–533.

Coleman, T. F. and Y. Li (1996). An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization 6*, 418–445.

Cont, R., C. Mancini, et al. (2011). Nonparametric tests for pathwise properties of semimartingales. *Bernoulli 17*(2), 781–813.

Correia, M. (2022). Target-date fund assets reach $3.27 trillion in 2021. `https://www.pionline.com/money-management/target-date-fund-assets-reach-327-trillion-2021`. [Online; accessed 27-Jun-2023].

CPP Investments (2021). Annual Report 2021. https://www.cppinvestments.com/the-fund/our-performance/financial-results/. [Online; accessed 5-Dec-2022].

CPP Investments (2022). Annual Report 2022. https://www.cppinvestments.com/wp-content/uploads/2022/06/CPP-Investments_F2022-Annual-Report-EN.pdf. [Online; accessed 5-Dec-2022].

Dang, D.-M. and P. A. Forsyth (2014). Continuous time mean-variance optimal portfolio allocation under jump diffusion: a numerical impulse control approach. *Numerical Methods for Partial Differential Equations 30*, 664–698.

Dang, D.-M. and P. A. Forsyth (2016). Better than pre-commitment mean-variance portfolio allocation strategies: A semi-self-financing Hamilton–Jacobi–Bellman equation approach. *European Journal of Operational Research 250*(3), 827–841.

Dantzig, G. B. and G. Infanger (1993). Multi-stage stochastic linear programs for portfolio optimization. *Annals of Operations Research 45*(1), 59–76.

Davis, M. and S. Lleo (2008). Risk-sensitive benchmarked asset management. *Quantitative Finance 8*(4), 415–426.

Dichtl, H., W. Drobetz, and M. Wambach (2016). Testing rebalancing strategies for stock-bond portfolos across different asset allocations. *Applied Economics 48*, 772–788.

Dimson, E., P. Marsh, and M. Staunton (2020). Credit Suisse global investment returns yearbook 2020 summary edition. https://www.credit-suisse.com/media/assets/private-banking-ux/docs/iwm-global/insights/credit-suisse-global-investment-returns-yearbook-2023-summary-edition.pdf. [Online; accessed 30-Aug-2023].

Dixon, M. F., I. Halperin, and P. Bilokon (2020). *Machine learning in Finance*, Volume 1406. Springer.

Donaldson, S., F. Kinniry, V. Maciulis, A. J. Patterson, and M. A. DiJoseph (2015). Vanguard's approach to target-date funds. https://institutional.vanguard.com/content/dam/inst/iig-transformation/insights/pdf/2022/Vanguards-approach-to-target-date-funds.pdf. [Online; accessed 31-Aug-2023].

Duarte, V., J. Fonseca, A. S. Goodman, and J. A. Parker (2021). Simple allocation rules and optimal portfolio choice over the lifecycle. Technical report, National Bureau of Economic Research.

Easterbrook, F. H. and D. R. Fischel (1985). Limited liability and the corporation. *U. Chi. L. Rev. 52*, 89.

Esch, D. N. and R. O. Michaud (2014). The false promise of target date funds. Working paper, New Frontier Advisors, LLC.

Estrada, J. and M. Kritzman (2019). Toward determining the optimal investment strategy for retirement. *The Journal of Retirement 7*(1), 35–42.

Farago, A. and E. Hjalmarsson (2023). Small rebalanced portfolios often beat the market over long horizons. *The Review of Asset Pricing Studies 13*(2), 307–342.

Federal Reserve Board (1974). Electronic code of federal regulations. `https://www.ecfr.gov/current/title-12/chapter-II/subchapter-A/part-220`. [Online; accessed 21-Oct-2023].

Forsyth, P. A. (2020). Optimal dynamic asset allocation for DC plan accumulation/decumulation: Ambition-CVAR. *Insurance: Mathematics and Economics 93*, 230–245.

Forsyth, P. A. (2022). A stochastic control approach to defined contribution plan decumulation: "the nastiest, hardest problem in finance". *North American Actuarial Journal 26*(2), 227–251.

Forsyth, P. A., K. Vetzal, and G. Westmacott (2020). Dynamic asset allocation for dc pension decumulation with a variable spending rule. *ASTIN Bulletin 50:2*, 419–447.

Forsyth, P. A. and K. R. Vetzal (2019). Optimal asset allocation for retirement saving: Deterministic vs. time consistent adaptive strategies. *Applied Mathematical Finance 26*(1), 1–37.

Forsyth, P. A. and K. R. Vetzal (2022). Multi-period mean expected-shortfall strategies:'cut your losses and ride your gains'. *Applied Mathematical Finance*, 1–37.

Fung, W. and D. A. Hsieh (1999). A primer on hedge funds. *Journal of empirical finance 6*(3), 309–331.

Gao, Z., Y. Gao, Y. Hu, Z. Jiang, and J. Su (2020). Application of deep Q-network in portfolio management. In *2020 5th IEEE International Conference on Big Data Analytics (ICBDA)*, pp. 268–275. IEEE.

Gastineau, G. L. (2008). The short side of 130/30 investing for the conservative portfolio manager. *Journal of Portfolio Management 34*(2), 39.

Gay, C. (2012). Leverage, volatility, and the curious case of 130/30 funds. `https://finance.yahoo.com/news/leverage-volatility-curious-case-130-160221697.html`. [online, Yahoo Finance].

Ghilarducci, T., M. Papadopoulos, and A. Webb (2017). Inadequate retirement savings for workers nearing retirement. `https://www.economicpolicyresearch.org/images/docs/research/retirement_security/Account_Balances_adjusted_appendix_tables.pdf`. [Online; accessed 31-Aug-2023].

Glushkov, D. (2015). How smart are 'smart beta' ETFs? analysis of relative performance and factor exposure. *Available at SSRN 2594941*.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep learning*. MIT press.

Graf, S. (2017). Life-cycle funds: Much ado about nothing? *European Journal of Finance 23*, 974–998.

Graham, B. (2003). *The Intelligent Investor*. New York: HarperCollins. Revised edition, forward by J. Zweig.

Han, J. et al. (2016). Deep learning approximation for stochastic control problems. *arXiv preprint arXiv:1611.07422*.

Haugh, M. (2016). Mean-Variance optimization and the CAPM. `http://www.columbia.edu/~mh2078/FoundationsFE/MeanVariance-CAPM.pdf`. [Online; accessed 20-Jun-2023].

Homer, S. and R. E. Sylla (1996). *A history of interest rates*. Rutgers University Press.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural networks 4*(2), 251–257.

Javorcik, B. (2020). Reshaping of global supply chains will take place, but it will not happen fast. *Journal of Chinese Economic and Business Studies 18*(4), 321–325.

Johnson, G., S. Ericson, and V. Srimurthy (2007). An empirical analysis of 130/30 strategies. *Journal of Alternative Investments 10*(2), 31.

Johnson, S. (2013). The decline, fall and afterlife of 130/30. `https://www.ft.com/content/fdbf6284-b724-11e2-841e-00144feabdc0`. [online, Financial Times].

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Korhonen, J. A. O. and D. Kunz (2010). 130/30 investment strategies–is the active extension value adding? `https://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=2169582&fileOId=2435912`. [Online; accessed 31-Aug-2023].

Kou, S. G. (2002). A jump-diffusion model for option pricing. *Management science 48*(8), 1086–1101.

Kou, S. G. and H. Wang (2004). Option pricing under a double exponential jump diffusion model. *Management science 50*(9), 1178–1192.

Kratsios, A. and I. Bilokopytov (2020). Non-euclidean universal approximation. *Advances in Neural Information Processing Systems 33*, 10635–10646.

Krusen, C., F. Weber, and R. A. Weigand (2008). 130/30 funds: The evolution of active equity investing. *Special Issues 2008*(1), 176–185.

Lahaye, J., S. Laurent, and C. J. Neely (2011). Jumps, cojumps and macro announcements. *Journal of Applied Econometrics 26*(6), 893–921.

Lee, W., H. Yu, X. Rival, and H. Yang (2020). On correctness of automatic differentiation for non-differentiable functions. *Advances in Neural Information Processing Systems 33*, 6719–6730.

Leibowitz, M. L., S. Emrich, and A. Bova (2009). *Modern portfolio management: active long/short 130/30 equity strategies*. John Wiley & Sons.

Levy, H. (2016). Aging population, retirement, and risk taking. *Management Science 62*(5), 1415–1430.

Lhabitant, F.-S. (2000). Derivatives in portfolio management: Why beating the market is easy. EDHEC Working paper.

Li, D. and W.-L. Ng (2000). Optimal dynamic portfolio selection: Multiperiod mean-variance formulation. *Mathematical finance 10*(3), 387–406.

Li, X., X. Y. Zhou, and A. E. Lim (2002). Dynamic mean-variance portfolio selection with no-shorting constraints. *SIAM Journal on Control and Optimization 40*(5), 1540–1555.

Li, Y. and P. A. Forsyth (2019). A data-driven neural network approach to optimal asset allocation for target based defined contribution pension plans. *Insurance: Mathematics and Economics 86*, 189–204.

Lim, A. E. and B. Wong (2010). A benchmarking approach to optimal asset allocation for insurers and pension funds. *Insurance: Mathematics and Economics 46*(2), 317–327.

Lin, Y., R. D. MacMinn, and R. Tian (2015). De-risking defined benefit plans. *Insurance: Mathematics and Economics 63*, 52–65.

Lo, A. W. and P. N. Patel (2008). 130/30: The new long-only. *INSTITUTIONAL INVESTOR-NEW YORK- 42*(5), 186.

Looney, C. A. and A. M. Hardin (2009). Decision support for retirement portfolio management: Overcoming myopic loss aversion via technology design. *Management Science 55*(10), 1688–1703.

Lu, Y. and J. Lu (2020). A universal approximation theorem of deep neural networks for expressing probability distributions. *Advances in neural information processing systems 33*, 3094–3105.

Lucarelli, G. and M. Borrotti (2019). A deep reinforcement learning approach for automated cryptocurrency trading. In *Artificial Intelligence Applications and Innovations: 15th IFIP WG 12.5 International Conference, AIAI 2019, Hersonissos, Crete, Greece, May 24–26, 2019, Proceedings 15*, pp. 247–258. Springer.

Lucarelli, G. and M. Borrotti (2020). A deep Q-learning portfolio management framework for the cryptocurrency market. *Neural Computing and Applications 32*(23), 17229–17244.

L'Her, J.-F., R. Stoyanova, K. Shaw, W. Scott, and C. Lai (2016). A bottom-up approach to the risk-adjusted performance of the buyout fund market. *Financial Analysts Journal 72*(4), 36–48.

MacColl, L. (2023). CPP retirement pay dates for 2023: How much CPP will i get? `https://www.wealthsimple.com/en-ca/learn/how-much-cpp-retirement#how_to_maximize_your_cpp_payment`. [Online; accessed 31-Aug-2023].

MacMinn, R., P. Brockett, J. Wang, Y. Lin, and R. Tian (2014). The securitization of longevity risk and its implications for retirement security. *Recreating sustainable retirement*, 134–160.

Malliaris, A. G. and M. E. Malliaris (2008). Investment principles for individual retirement accounts. *Journal of Banking & Finance 32*(3), 393–404.

Mancini, C. (2009). Non-parametric threshold estimation for models with stochastic diffusion coefficient and jumps. *Scandinavian Journal of Statistics 36*(2), 270–296.

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

McCrary, S. A. (2004). *Hedge fund course*. John Wiley & Sons.

Merton, R. C. (1969). Lifetime portfolio selection under uncertainty: The continuous-time case. *The Review of Economics and Statistics 51*(3), 247–257.

Merton, R. C. (1971). Optimum consumption and portfolio rules in a continuous-time model. *Journal of Economic Theory 3:4*, 373–413.

Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of financial economics 3*(1-2), 125–144.

Moody's (2023a). Moody's Seasoned Aaa Corporate Bond Yield [AAA], retrieved from FRED, Federal Reserve Bank of St. Louis;. https://fred.stlouisfed.org/series/AAA. [Online; accessed 18-Jan-2023].

Moody's (2023b). Moody's Seasoned Baa Corporate Bond Yield [BAA], retrieved from FRED, Federal Reserve Bank of St. Louis;. https://fred.stlouisfed.org/series/BAA. [Online; accessed 18-Jan-2023].

Mulvey, J. M. and H. Vladimirou (1989). Stochastic network optimization models for investment planning. *Annals of Operations Research 20*(1), 187–217.

NASDAQ (2023). NASDAQ Composite Index. `https://www.nasdaq.com/market-activity/index/comp/historical`. [Online; accessed 15-Feb-2023].

Ni, C., Y. Li, P. Forsyth, and R. Carroll (2022). Optimal asset allocation for outperforming a stochastic benchmark target. *Quantitative Finance 22*(9), 1595–1626.

Ni, C., Y. Li, and P. A. Forsyth (2023). Optimal asset allocation in a high inflation regime: a leverage-feasible neural network approach. *arXiv preprint arXiv:2304.05297*.

Nicolosi, M., F. Angelini, and S. Herzel (2018). Portfolio management with benchmark related incentives under mean reverting processes. *Annals of Operations Research 266*, 373–394.

Norges Bank (2022). Investment Strategy. `https://www.nbim.no/en/the-fund/how-we-invest/investment-strategy/`. [Online; accessed 5-Dec-2022].

Oderda, G. (2015). Stochastic portfolio theory optimization and the origin of rule based investing. *Quantitative Finance 15(8)*, 1259–1266.

O'Donoghue, T. and M. Rabin (1998). Procrastination in preparing for retirement. *University of California-Berkeley Working Paper*.

Øksendal, B. K. and A. Sulem (2007). *Applied stochastic control of jump diffusions*, Volume 498. Springer.

Park, H., M. K. Sim, and D. G. Choi (2020). An intelligent financial portfolio trading strategy using deep Q-learning. *Expert Systems with Applications 158*, 113573.

Patton, A., D. N. Politis, and H. White (2009). Correction to "automatic block-length selection for the dependent bootstrap" by d. politis and h. white. *Econometric Reviews 28*(4), 372–375.

Phalippou, L. (2014). Performance of buyout funds revisited? *Review of Finance 18*(1), 189–218.

Pielichata, P. (2023). Netherlands passes law for new defined contribution system. `https://www.pionline.com/retirement-plans/netherlands-passes-law-new-defined-contribution-system`. [Online; accessed 18-Oct-2023].

Plyakha, Y., R. Uppal, and G. Vilkov (2021). Equal or value weighting? implications for asset-pricing tests. In C. Zopounidis, R. Benkraiem, and I. Kalaitzoglou (Eds.), *Financial Risk Management and Modeling*, pp. 295–347. Springer International Publishing.

Politis, D. N. and J. P. Romano (1991). *A circular block-resampling procedure for stationary data*. Purdue University. Department of Statistics.

Politis, D. N. and J. P. Romano (1994). The stationary bootstrap. *Journal of the American Statistical Association 89*(428), 1303–1313.

Politis, D. N. and H. White (2004). Automatic block-length selection for the dependent bootstrap. *Econometric Reviews 23*(1), 53–70.

Qian, E. (2011). Risk parity and diversification. *Journal of Investing 20*(1), 119.

Rekenthaler, J. (2022). Nobody likes index funds, except investors. `https://www.morningstar.com/articles/1096069/nobody-likes-index-funds-except-investors`. [online, Morningstar Inc.].

Reppen, A. M., H. M. Soner, and V. Tissot-Daguette (2023). Deep stochastic optimization in finance. *Digital Finance 5*(1), 91–111.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence 1*(5), 206–215.

Rupert, P. and G. Zanella (2015). Revisiting wage, earnings, and hours profiles. *Journal of Monetary Economics 72*, 114–130.

Samo, Y.-L. K. and A. Vervuurt (2016). Stochastic portfolio theory: A machine learning perspective. *arXiv preprint arXiv:1605.02654*.

Silver, D., G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller (2014). Deterministic policy gradient algorithms. In *International conference on machine learning*, pp. 387–395. Pmlr.

Statistics Canada (2022). Income of individuals by age group, sex and income source, canada, provinces and selected census metropolitan areas. `https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=1110023901`. [Online; accessed 31-Apr-2022].

Tabb, L. and J. Johnson (2007). Alternative Investments 2007: The Quest for Alpha. `https://research.tabbgroup.com/report/v05-008-alternative-investments-2007-quest-alpha`. [Technical report, Tabb Group].

Taljaard, B. H. and E. Mare (2021). Why has the equal weight portfolio underperformed and what can we do about it? *Quantitative Finance 21*(11), 1855–1868.

Tepla, L. (2001). Optimal investment with minimum performance constraints. *Journal of Economic Dynamics and Control 25*(10), 1629–1645.

The Federal Reserve (2011). What is an acceptable level of inflation? https://www.federalreserve.gov/faqs/5D58E72F066A4DBDA80BBA659C55F774.htm. [Online; accessed 5-Dec-2022].

Tsang, K. H. and H. Y. Wong (2020). Deep-learning solution to portfolio selection with serially dependent returns. *SIAM Journal on Financial Mathematics 11*(2), 593–619.

van Staden, P. M., D.-M. Dang, and P. A. Forsyth (2021). On the distribution of terminal wealth under dynamic mean-variance optimal investment strategies. *SIAM Journal on Financial Mathematics 12*(2), 566–603.

van Staden, P. M., P. A. Forsyth, and Y. Li (2021). A data-driven neural network approach to dynamic factor investing. *Working paper, University of Waterloo, available at https://cs.uwaterloo.ca/~paforsyt/Factor_NN.pdf*.

van Staden, P. M., P. A. Forsyth, and Y. Li (2023a). Across-time risk-aware strategies for outperforming a benchmark. *European Journal of Operational Research*.

van Staden, P. M., P. A. Forsyth, and Y. Li (2023b). Beating a benchmark: dynamic programming may not be the right numerical approach. *SIAM Journal on Financial Mathematics 14*(2), 407–451.

Vanguard (2019). Historical index risk/return. https://advisors.vanguard.com/VGApp/iip/advisor/csa/analysisTools/portfolioAnalytics/historicalRiskReturn. [Online; accessed 5-May-2023].

Vigna, E. (2014). On efficiency of mean–variance based portfolio selection in defined contribution pension schemes. *Quantitative finance 14*(2), 237–258.

Wang, J. and P. A. Forsyth (2010). Numerical solution of the Hamilton-Jacobi-Bellman formulation for continuous time mean variance asset allocation. *Journal of Economic Dynamics and control 34*(2), 207–230.

Wang, J. and P. A. Forsyth (2012). Comparison of mean variance like strategies for optimal asset allocation problems. *International journal of theoretical and applied finance 15*(02), 1250014.

Wiafe, O. K., A. K. Basu, and E. Te Chen (2020). Portfolio choice after retirement: Should self-annuitisation strategies hold more equities? *Economic Analysis and Policy 65*, 241–255.

Yang, H., X.-Y. Liu, and Q. Wu (2018). A practical machine learning approach for dynamic stock recommendation. In *2018 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE)*, pp. 1693–1697. IEEE.

Yao, D. D., S. Zhang, and X. Y. Zhou (2006). Tracking a financial benchmark using a few assets. *Operations Research 54*(2), 232–246.

Zhang, C., Z. Zhang, M. Cucuringu, and S. Zohren (2021). A universal end-to-end approach to portfolio optimization via deep learning. *arXiv preprint arXiv:2111.09170*.

Zhang, Q. and Y. Gao (2017). Portfolio selection based on a benchmark process with dynamic value-at-risk constraints. *Journal of Computational and Applied Mathematics 313*, 440–447.

Zhang, Z., S. Zohren, and S. Roberts (2020). Deep learning for portfolio optimization. *The Journal of Financial Data Science 2*(4), 8–20.

Zhao, Y. (2007). A dynamic model of active portfolio management with benchmark orientation. *Journal of Banking & Finance 31*(11), 3336–3356.

Zhou, X. Y. and D. Li (2000). Continuous-time mean-variance portfolio selection: A stochastic LQ framework. *Applied Mathematics and Optimization 42*, 19–33.

APPENDICES

Appendix A

Supplementary materials for Chapter 2

A.1 Minimizing symmetric quadratic penalty objective function

In this section, we show that the asymmetric penalties give a more favorable terminal wealth distribution compared to a symmetric quadratic penalty objective function $\mathbb{E}\Big[\big(W(T) - e^{sT} \cdot W_b(T)\big)^2\Big]$.

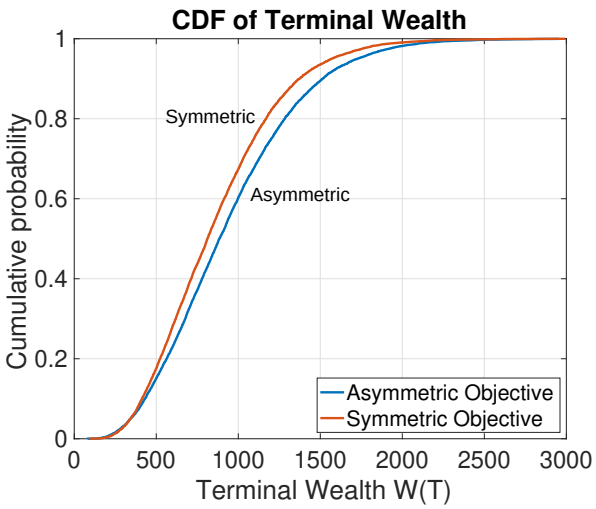
We train two adaptive strategies under our proposed asymmetric objective function (2.6) and the quadratic symmetric objective function separately with the same bootstrap resampled dataset (expected blocksize of 0.5 years), and test the two strategies on the same bootstrap resampled dataset with expected blocksize of 2 years. We present the following testing results.

We can see from Table A.1 that the terminal wealth of the adaptive strategy trained with the asymmetric objective function achieves a higher expected and median terminal wealth. We can also observe that the terminal wealth distribution from the asymmetric objective function is more right-skewed than the distribution from the quadratic symmetric objective function from Figure A.1a. If we compare the path-wise terminal wealth, as shown in Figure A.1b, we can see that the asymmetric objective function leads to higher terminal wealth most of the time.

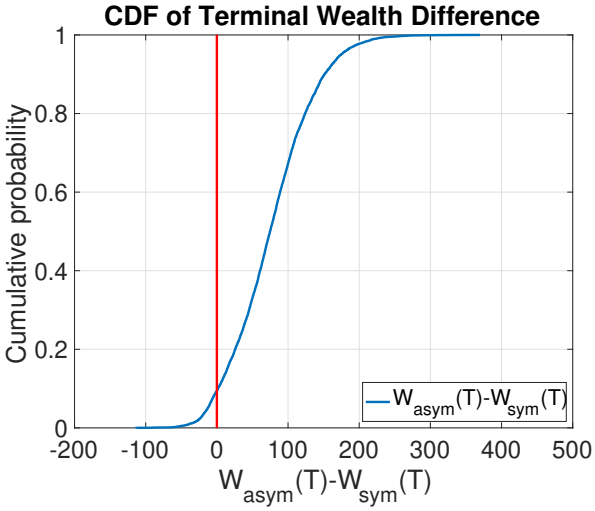
We believe the superior performance from the asymmetric objective function is because the linear penalty on outperformance incentivizes a more right-skewed distribution for the optimizer than the symmetric quadratic penalties, in terms of both underperformance and outperformance.

Testing Results on Bootstrap Data with Expected Blocksize = 2 years					
Strategy	$E(W_T)$	$std(W_T)$	$median(W_T)$	$Pr(W_T < 500)$	$Pr(W_T < 600)$
asymmetric objective	940	430	885	0.15	0.23
symmetric objective	864	387	811	0.18	0.28

Table A.1: Terminal wealth statistics of adaptive strategies trained on bootstrap resampled data with expected blocksize $\hat{b} = 0.5$ years and tested on bootstrap resampled data with expected blocksize $\hat{b} = 2$ years



(a) CDF of the terminal wealth of both objective functions



(b) Wealth difference between two objective functions

Figure A.1: Terminal wealth and terminal wealth difference, comparing symmetric and asymmetric objective functions.

A.2 Additional tables for robustness test with different blocksizes

Test Results: Market Cap Weighted					
Strategy	$E(W_T)$	$std(W_T)$	$median(W_T)$	$Pr(W_T < median(W_T^{CP}))$	$Pr(W_T < median(W_T^{NN}))$
Expected Blocksize $\hat{b} = 0.5$ years					
constant proportion($p = .5$)	678	286	623.07	0.50	0.81
NN adaptive	949	478	874.84	0.27	0.50
Expected Blocksize $\hat{b} = 1$ years					
constant proportion($p = .5$)	674	273	623.99	0.50	0.81
NN adaptive	942	459	878.60	0.27	0.50
Expected Blocksize $\hat{b} = 2$ years					
constant proportion($p = .5$)	676	263	631.06	0.50	0.81
NN adaptive	945	438	882.74	0.26	0.50
Expected Blocksize $\hat{b} = 5$ years					
constant proportion($p = .5$)	669	244	626.11	0.50	0.83
NN adaptive	940	404	881.87	0.23	0.50
Expected Blocksize $\hat{b} = 8$ years					
constant proportion($p = .5$)	669	233	632.24	0.50	0.84
NN adaptive	945	388	892.84	0.22	0.50
Expected Blocksize $\hat{b} = 10$ years					
constant proportion($p = .5$)	667	223	635.29	0.50	0.85
NN adaptive	942	373	895.88	0.22	0.50

Table A.2: Trained on bootstrap resampled data with $\hat{b} = 1$ years

Test Results: Market Cap Weighted					
Strategy	$E(W_T)$	$std(W_T)$	$median(W_T)$	$Pr(W_T < median(W_T^{CP}))$	$Pr(W_T < median(W_T^{NN}))$
Expected Blocksize $\hat{b} = 0.5$ years					
constant proportion($p = .5$)	678	286	623.07	0.50	0.83
NN adaptive	962	491	903.07	0.27	0.50
Expected Blocksize $\hat{b} = 1$ years					
constant proportion($p = .5$)	674	273	623.99	0.50	0.83
NN adaptive	954	470	905.02	0.27	0.50
Expected Blocksize $\hat{b} = 2$ years					
constant proportion($p = .5$)	676	263	631.06	0.50	0.84
NN adaptive	958	446	912.31	0.26	0.50
Expected Blocksize $\hat{b} = 5$ years					
constant proportion($p = .5$)	669	244	626.11	0.50	0.85
NN adaptive	954	409	914.34	0.23	0.50
Expected Blocksize $\hat{b} = 8$ years					
constant proportion($p = .5$)	669	233	632.24	0.50	0.87
NN adaptive	961	392	928.89	0.22	0.50
Expected Blocksize $\hat{b} = 10$ years					
constant proportion($p = .5$)	667	223	635.29	0.50	0.88
NN adaptive	961	380	930.15	0.21	0.50

Table A.3: Trained on bootstrap resampled data with $\hat{b} = 2$ years

Test Results: Market Cap Weighted					
Strategy	$E(W_T)$	$std(W_T)$	$median(W_T)$	$Pr(W_T < median(W_T^{CP}))$	$Pr(W_T < median(W_T^{NN}))$
Expected Blocksize $\hat{b} = 0.5$ years					
constant proportion($p = .5$)	678	286	623.07	0.50	0.86
NN adaptive	995	495	963.03	0.26	0.50
Expected Blocksize $\hat{b} = 1$ years					
constant proportion($p = .5$)	674	273	623.99	0.50	0.87
NN adaptive	988	478	963.28	0.25	0.50
Expected Blocksize $\hat{b} = 2$ years					
constant proportion($p = .5$)	676	263	631.06	0.50	0.88
NN adaptive	994	458	973.65	0.25	0.50
Expected Blocksize $\hat{b} = 5$ years					
constant proportion($p = .5$)	669	244	626.11	0.50	0.89
NN adaptive	997	427	976.51	0.22	0.50
Expected Blocksize $\hat{b} = 8$ years					
constant proportion($p = .5$)	669	233	632.24	0.50	0.90
NN adaptive	1011	415	993.88	0.21	0.50
Expected Blocksize $\hat{b} = 10$ years					
constant proportion($p = .5$)	667	223	635.29	0.50	0.92
NN adaptive	1015	409	996.57	0.20	0.50

Table A.4: Trained on bootstrap resampled data with $\hat{b} = 5$ years

Test Results: Market Cap Weighted					
Strategy	$E(W_T)$	$std(W_T)$	$median(W_T)$	$Pr(W_T < median(W_T^{CP}))$	$Pr(W_T < median(W_T^{NN}))$
Expected Blocksize $\hat{b} = 0.5$ years					
constant proportion($p = .5$)	678	286	623.07	0.50	0.86
NN adaptive	980	480	945.12	0.25	0.50
Expected Blocksize $\hat{b} = 1$ years					
constant proportion($p = .5$)	674	273	623.99	0.50	0.86
NN adaptive	973	464	947.99	0.25	0.50
Expected Blocksize $\hat{b} = 2$ years					
constant proportion($p = .5$)	676	263	631.06	0.50	0.87
NN adaptive	979	443	957.32	0.25	0.50
Expected Blocksize $\hat{b} = 5$ years					
constant proportion($p = .5$)	669	244	626.11	0.50	0.88
NN adaptive	981	412	959.86	0.21	0.50
Expected Blocksize $\hat{b} = 8$ years					
constant proportion($p = .5$)	669	233	632.24	0.50	0.90
NN adaptive	994	399	976.44	0.21	0.50
Expected Blocksize $\hat{b} = 10$ years					
constant proportion($p = .5$)	667	223	635.29	0.50	0.91
NN adaptive	996	390	980.07	0.20	0.50

Table A.5: Trained on bootstrap resampled data with $\hat{b} = 8$ years

Test Results: Market Cap Weighted					
Strategy	$E(W_T)$	$std(W_T)$	$median(W_T)$	$Pr(W_T < median(W_T^{CP}))$	$Pr(W_T < median(W_T^{NN}))$
Expected Blocksize $\hat{b} = 0.5$ years					
constant proportion($p = .5$)	678	286	623.07	0.50	0.84
NN adaptive	963	468	920.86	0.25	0.50
Expected Blocksize $\hat{b} = 1$ years					
constant proportion($p = .5$)	674	273	623.99	0.50	0.84
NN adaptive	957	451	923.63	0.25	0.50
Expected Blocksize $\hat{b} = 2$ years					
constant proportion($p = .5$)	676	263	631.06	0.50	0.85
NN adaptive	962	431	932.13	0.25	0.50
Expected Blocksize $\hat{b} = 5$ years					
constant proportion($p = .5$)	669	244	626.11	0.50	0.87
NN adaptive	962	399	937.08	0.22	0.50
Expected Blocksize $\hat{b} = 8$ years					
constant proportion($p = .5$)	669	233	632.24	0.50	0.88
NN adaptive	973	384	951.40	0.21	0.50
Expected Blocksize $\hat{b} = 10$ years					
constant proportion($p = .5$)	667	223	635.29	0.50	0.90
NN adaptive	973	373	954.63	0.20	0.50

Table A.6: Trained on bootstrap resampled data with $\hat{b} = 10$ years