Optimal Asset Allocation For Outperforming A Stochastic Benchmark Target

Chendi Ni[1], Yuying Li[2], Peter Forsyth[3], and Ray Carroll[4]

[1]Cheriton School of Computer Science, University of Waterloo, Waterloo, N2L 3G1, Canada, chendi.ni@uwaterloo.ca
[2]Cheriton School of Computer Science, University of Waterloo, Waterloo, N2L 3G1, Canada, yuying@uwaterloo.ca
[3]Cheriton School of Computer Science, University of Waterloo, Waterloo, N2L 3G1, Canada, paforsyt@uwaterloo.ca
[4]Neuberger Berman Breton Hill, Toronto, M4W 1A8, Canada, Ray.Carroll@nb.com

April 26, 2022

Abstract

We propose a data-driven Neural Network (NN) optimization framework to determine the optimal multi-period dynamic asset allocation strategy for outperforming a general stochastic target. We formulate the problem as an optimal stochastic control with an asymmetric, distribution shaping, objective function. The proposed framework is illustrated with the asset allocation problem in the accumulation phase of a defined contribution pension plan, with the goal of achieving a higher terminal wealth than a stochastic benchmark. We demonstrate that the data-driven approach is capable of learning an adaptive asset allocation strategy directly from historical market returns, without assuming any parametric model of the financial market dynamics. The optimal adaptive strategy outperforms the benchmark constant proportion strategy, achieving a higher terminal wealth with a 90% probability, a 46% higher median terminal wealth, and a significantly more right-skewed terminal wealth distribution.

1 Introduction

1.1 Literature Review and Overview

The seminal work by Markowitz (1952) uses the mean-variance approach to study the asset allocation problem and establishes the foundation for modern portfolio theory. Following Markowitz's pioneering work, Merton (1969) extends the problem to the multi-period continuous-time asset allocation setting, and uses stochastic control techniques to derive a closed-form optimal portfolio that maximizes a CRRA utility function of terminal wealth. Since then, the majority of research on multiperiod asset allocation has focused on maximizing a utility function of the terminal wealth or other absolute performance metrics (Merton, 1971; Browne, 1997; Blanchet-Scalliet et al., 2008; Ang et al., 2014).

As the companion paper to Merton (1969), Samuelson (1975) studies the discrete-time multiperiod asset allocation problem, and uses stochastic control techniques to derive the closed-form solution of the optimal allocation under a utility function for more general probability distributions. However, when incorporating more realistic constraints such as transaction costs and leverage constraints, closed-form solutions often cannot be derived for discrete-time multiperiod problems.

More recently, there has been some progress in using dynamic programming for the discrete-time multiperiod problem. In the discrete rebalancing case, dynamic programming is used to find optimal portfolio weights at each discrete time point. Usually the objective function is expressed as maximizing the expected utility function value (Mulvey and Vladimirou, 1989; Dantzig and Infanger, 1993; Cariño and Turner, 1998; Cheung and Yang, 2004). However, while dynamic programming provides flexibility from a modeling perspective, the computational complexity increases exponentially with the number of state variables and is only tractable when the number of assets (stochastic factors) is relatively small.

While most existing work on multiperiod 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 follows geometric Brownian motions. Under these assumptions, Browne (1999, 2000) derives 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 good as a stochastic benchmark. Basak et al. (2006) relaxes the minimum performance constraints used in Tepla (2001) and certain shortfall probability is allowed in return for some upside potential. Bajeux-Besnainou et al. (2013) introduces 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) uses 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 is allowed), the authors show that a portfolio which 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 assumes that the growth rate is stochastic and driven by latent factors, which addresses the short-coming of assuming a deterministic market return in Al-Aradi and Jaimungal (2018). We remark that all these work is in a continuous-time setting with unconstrained controls. To the best of our knowledge, little work is done on discrete-time multiperiod 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 which generates the optimal portfolio. However, a numerical HJB equation solution is only practical if there are a small (three or less) state variables (≤ 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 three assets and a three asset benchmark setting, the problem has six 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).

1.2 Overview of the Data-driven Methodology

To overcome the aforementioned challenges, in this work we propose a data-driven framework and use market asset return data directly to solve a scenario-based stochastic optimal control formulation, corresponding to the original stochastic control problem. With this approach, we skip the step of postulating a parametric stochastic model, and then calibrating this model to data. In addition, we solve the stochastic optimal control problem directly, without invoking dynamic programming to transform it into a PDE problem (thus avoiding the curse of dimensionality). The optimal control is represented as a neural network (NN) which is learned through training on bootstrap resampled historical data.

The features for the NN can include any state variable that influences the optimal strategy, including the state variables associated with a stochastic target. We design a specific objective function to create a desirable terminal wealth distribution. This is done by measuring the relative performance of the strategy against an elevated final wealth of the stochastic target strategy to penalize extreme losses and limit unlikely extreme gains.

We formulate a general optimal control problem for the multi-period asset allocation portfolio which outperforms a benchmark as an optimal stochastic control problem. We propose a benchmark target-based objective function which measures the difference between the terminal wealth of the adaptive strategy and a path-dependent elevated target (which is the terminal wealth of the benchmark strategy multiplied by a pre-defined growth factor). The objective function is designed as a double-sided penalty function to force the terminal wealth of the adaptive strategy to be close to the elevated target. The NN model takes three features as inputs: the current wealth of the adaptive portfolio, the current wealth of the constant proportion portfolio, and the time remaining. In the case that the underlying assets follow simple stochastic processes, it can be shown that the control is only a function of these variables.

Instead of formulating the problem as an HJB equation derived from dynamic programming, we solve the single original optimal control problem directly as in Li and Forsyth (2019). We define an objective function in terms of the terminal wealth, and then solve for the control directly, using a data-driven approach. The proposed data-driven approach does not require an estimation of the parameters of an assumed parametric model for traded assets. We represent the control using a shallow neural network (NN). We remark that shallow learning is found to outperform deep learning for asset pricing in Gu et al. (2018). We also note that good results are obtained in Hejazi and Jackson (2016) with an NN containing only one hidden layer (shallow learning), in which the shallow neural network learns a good choice of distance function for efficiently and accurately interpolating the Greeks for the input portfolio of Variable Annuity contracts.

It is common practice in the financial industry to train and test strategy performance by splitting the historical market data path into two segments - one for training and the other for testing[1]. We take a different approach. We aim to determine an investment strategy that would perform well statistically on a large set of data paths created through bootstrap resampling, rather than on a single historical data path. To achieve this, we generate additional data paths from the historical market data path by block bootstrap resampling of the historical data (see, e.g., Politis and Romano (1994); Politis and White (2004); Patton et al. (2009)). Once we have a large set of price paths from bootstrap resampling, we split them into the training data set and the testing data set.

To demonstrate the robustness of our approach, we test the optimal adaptive strategy on market data with different distributions from the training data. We first test the optimal adaptive strategy, learned

[1] This is often known as the process of backtesting.(Harvey and Liu, 2015)

from bootstrap resampled data with a given expected blocksize, on bootstrap resampled data with different expected blocksizes (thus different distributions, as noted by Politis and Romano (1994)). We then test the adaptive strategy learned from synthetic data generated from a parametric jump-diffusion stochastic process (estimated from the same single historic path) on bootstrap resampled data. Finally, we test the strategy learned on bootstrap resampling data from a segment of the historical market data path on bootstrap resampling data generated from another non-overlapping segment of the historical data path.

To the best of our knowledge, the closest work related to the research in this paper is Samo and Vervuurt (2016), in which the authors also use a data-driven machine learning approach for constructing a dynamic strategy which outperforms a benchmark. Samo and Vervuurt (2016) approximate the control by a Gaussian process and solve the optimal hyperparameters using Bayesian inference. However, they do not assess the distributional properties of the investment strategy, but rather evaluate the performance on a single historical path. In addition, they only validate the performance of the strategy for a relatively short period from 1992-2014. In contrast to our focus on the pension plan in this work, they consider the case of daily rebalancing with a large number of stocks which would not be typical of a defined contribution pension plan.

Furthermore, our approach differs from (Samo and Vervuurt, 2016) in the learning methodology, both with respect to learning algorithms and data utilization. Our approach can be applied to a general multi-period asset allocation problem with few assumptions. In addition, it can readily be scaled up to high dimensional problems (i.e. more assets and features). A shallow network is sufficient here, leading to a relatively small number of parameters and computationally efficient training. In contrast to Samo and Vervuurt (2016), we use a small number of feature variables that only depend on the state of the adaptive portfolio and the benchmark portfolio, rather than market-related signals. As a result, the trading strategy is easy to interpret, practical to implement and the model is less prone to overfitting. Furthermore, our computational results demonstrate that the optimal adaptive strategy has a higher expected terminal wealth as well as a more favorable terminal wealth distribution than the benchmark strategy.

1.3 Example: Defined Contribution (DC) Investment Plan

To illustrate the proposed framework, 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 regularly with 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.

Here we use the example of the DC plan to illustrate how an employee can use our proposed framework to construct an asset allocation plan to beat a stochastic benchmark target. We note that the employee is the investor in the DC plan since he/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 on rebalancing.

In this work, 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 he/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 up by empirical studies (Basu et al., 2011; Arnott et al., 2013; Esch and Michaud, 2014). We also provide empirical evidence in Section 6 to support this argument. Therefore, in this article, we set the benchmark target to be the constant proportion strategy as it is easy to understand and implement. Nevertheless, for readers who are interested in results when target-date funds are chosen as a benchmark, we have included results in Section 6, in which we show that the our methodology learns an adaptive strategy that has a superior terminal wealth distribution compared to the benchmark target-date fund.

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 article. This would be a typical average allocation to equities over the full accumulation phase of a lifecycle fund.[2]

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 should clarify that the proposed framework is able to handle more assets. In fact, we have included an example with three assets in Appendix A.3.

Using the proposed framework to determine the optimal multi-period dynamic asset allocation strategy for outperforming a stochastic target, we address a natural and interesting question of whether it is possible to develop a dynamic allocation strategy that outperforms the constant proportion strategy.

Finally, we remark that the stylized DC plan accumulation problem in this article 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).

Our contribution in this article is primarily methodological. We use an entirely data-driven approach (no parametric stochastic processes), and we approximate the optimal policy directly, without resorting to dynamic programming.

Hence, the stylized DC plan investment example is used to demonstrate the potential benefit of the proposed data-driven framework, which is one of the main goals of this work. As noted above, DC plan investment strategies are just a part of a true financial plan, which would consider many other critical issues, e.g. retirement dates, post-retirement plans, and labour income stability. Applying machine learning techniques to the full financial planning process is an active area of research, but beyond the scope of this paper.

1.4 Contributions

In this research, we make the following contributions:

[2]A lifecycle fund is based on the intuitive concept of allocating a high equity weight during the early employment years, and then moving to bonds as retirement nears. However, as shown in Graf (2017), this strategy does not outperform a constant weight strategy.

- Different from the commonly used one-sided quadratic shortfall objective function, we propose a new asymmetric distribution shaping objective function for the optimal asset allocation problem that is more suitable for the task of outperforming a benchmark strategy. The proposed objective function produces an optimal dynamic and adaptive strategy which yields significantly higher median terminal wealth than the stochastic benchmark, with only a small probability (and magnitude) of underperformance.

- We include a theoretical analysis to show that the probability of observing the same sequence of returns in training and testing data sets of bootstrap resampled data, with different blocksizes, is negligible for practical blocksizes. This suggests that training/testing data can be generated from a single historical path (if sufficiently long) merely by using different blocksizes and justifies the use of bootstrap resampling method.

- We use a selection of different training/testing data to validate our results, including different blocksizes (as in (Li and Forsyth, 2019)) but we also use non-lapping data periods to illustrate the robustness of the proposed methodology.

- Our work has significant empirical importance and implications. In particular, we have included constant weight strategies as well as industry standard glide path strategies as benchmarks in the numerical experiments. We show that for the example of a defined contribution pension plan, the adaptive strategy learned from the data-driven framework has a more favorable terminal wealth distribution than benchmark strategies with a higher expected terminal wealth and significantly less downside risk.

2 Formulation of Stochastic Benchmark Outperformance Problem

2.1 The Optimization Problem

Let the initial time $t_0 = 0$ and consider a set $\mathcal{T}$ of rebalancing times

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

The fraction of total wealth allocated to each asset is adjusted at times t_n, $n = 0, \ldots, N-1$, with the investment horizon $t_N = T$. Consider an investment problem in M assets.

Assume that, at time t, a fund holds wealth of amount $W_m(t)$ in asset m, $m = 1, \ldots, M$. The total value of the portfolio at t is then

$$W(t) = \sum_{m=1}^{M} W_m(t) \ . \tag{2.2}$$

For any given time t and arbitrary function $f(t)$, define $f(t^+) = \lim_{\epsilon \to 0^+} f(t+\epsilon)$, and $f(t^-) = \lim_{\epsilon \to 0^+} f(t-\epsilon)$. Assume that $W(t_0^-) = 0$, i.e., the initial value of the portfolio before any cash injection is zero, and let $q(t_n)$ represent an *a priori* specified cash injection schedule.

We denote the allocation at stage n by an allocation vector p_n, $n = 0, \ldots, N-1$. Given the allocation control vectors $p_0, p_1, \ldots, p_{N-1}$, the statistical properties of the terminal wealth of the adaptive portfolio $W(T)$ can be determined. Similarly, given a benchmark allocation vector $\tilde{p}_n$, the final wealth of the benchmark portfolio $W_b(T)$ can also be determined. The time evolution of $W(t)$ and $W_b(t)$ is given by

$$\begin{aligned}
&for\ n = 0, 1, ..., N-1 \\
&\quad W(t_n^+) = W(t_n^-) + q(t_n) \\
&\quad W_b(t_n^+) = W_b(t_n^-) + q(t_n) \\
&\quad W(t_{n+1}^-) = p_n^T R(t_n) W(t_n^+) \\
&\quad W_b(t_{n+1}^-) = \tilde{p}_n^T R(t_n) W_b(t_n^+) \\
&end,
\end{aligned}$$

where $R(t_n)$ is the vector of returns on assets in (t_n^+, t_{n+1}^-).

Our first goal 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 the terminal wealth of the constant proportion strategy, as in Li and Forsyth (2019). Note, however, that in our case, the benchmark is stochastic. This leads to the following optimization problem ($\mathbb{E}[\cdot]$ is the expectation operator):

$$\min_{p_0, p_1, \ldots, p_{N-1}} \mathbb{E}\left[\min\left(W(T) - W_b(T), 0\right)^2\right]. \tag{2.3}$$

Unfortunately an optimal solution[3] to (2.3) is trivially the benchmark strategy $p_n = \tilde{p}_n, \forall n$, which indicates the formulation (2.3) does not sufficiently capture properties of the desired solution.

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

$$\min_{p_0, p_1, \ldots, p_{N-1}} \mathbb{E}\left[\min\left(W(T) - e^{sT} \cdot W_b(T), 0\right)^2\right], \tag{2.4}$$

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

We note, however, that the terminal wealth distribution from (2.4) has a quite significant left tail of underperformance instances. Such result is actually expected since we do not pose a penalty on the outperformance, and thus the terminal wealth distribution is not exactly concentrated around the elevated target. Therefore, we introduce a additional linear penalty on the outperformance case, hoping to force the terminal wealth of the adaptive strategy to be closer to the elevated target. Our asymmetric distribution shaping benchmark outperforming formulation becomes

$$\min_{p_0, p_1, \ldots, p_{N-1}} \mathbb{E}\left[\min\left(W(T) - e^{sT} \cdot W_b(T), 0\right)^2 + \max\left(W(T) - e^{sT} \cdot W_b(T), 0\right)\right]. \tag{2.5}$$

Figure 2.1 illustrates this asymmetric distribution shaping objective function.

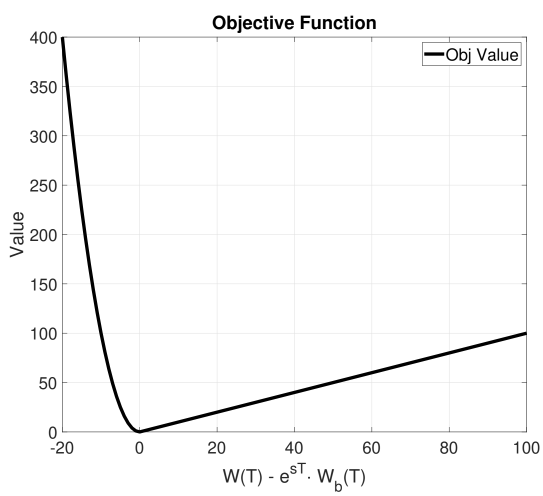


Figure 2.1: Asymmetric distribution shaping objective function with elevated target $e^{sT} \cdot W_b(T)$.

[3] In this case, there may be multiple optimal strategies which make the objective function identically zero. However, if both benchmark and outperformance portfolio start with the same initial wealth, the optimal strategy is clearly simply the benchmark strategy.

We note that such asymmetric penalties gives more favorable terminal wealth distributions than the symmetric quadratic penalty objective function of $\mathbb{E}\Big[\big(W(T) - e^{sT} \cdot W_b(T)\big)^2\Big]$, as shown in the numerical results in Appendix A.2. We believe it is because the asymmetric penalties incentivizes a more right-skewed distribution for the optimizer than the symmetric quadratic penalties because of less penalty on outperformance than underperformance.

While we choose the objective function (2.5) for outperforming a stochastic target in this paper, 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 which minimizes CVaR, for example, see (Forsyth, 2021; Forsyth and Vetzal, 2019). If an investor is concerned with path-dependent performance measures such as draw-down and variation over time, then such measures should be incorporated in the objective function. For example, the quadratic variation penalty used in (Al-Aradi and Jaimungal, 2018), which is time-averaged instantaneous volatility relative to a benchmark, can be introduced to penalize the deviation from the benchmark portfolio on a running basis.

While the discussions of these objective functions are out of the scope of this paper, we remark that our proposed data-driven neural network framework does not depend on any specific form of the objective function.

2.2 The Neural Network Approach for Solving the Optimization Problem

If we postulate parametric stochastic processes for prices of the traded assets, mathematically, the controls $p_0, ..., p_{N-1}$ can be determined using dynamic programming. This will result in a nonlinear HJB PDE (see (Al-Aradi and Jaimungal, 2018) for example). In the absence of any closed-form solution, computing a solution of 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 target portfolios (Dang and Forsyth, 2014). Consequently, even for this comparatively simple case, this would result in a four-dimensional HJB PDE.

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 $p_0, \ldots, p_{N-1}$ using dynamic programming, we propose a data driven approach as follows. We represent the optimal control as a function of several features $F(t)$, i.e., at t_n, $n = 0, 1, \ldots, N-1$,

$$p_n = p(F(t_n))$$

Example 1 (Two Asset Problem with Benchmark $W_{50/50}$)**.** *In our 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 for* $F(t)$*: (i)* $W(t_n)$*, the wealth of the adaptive portfolio at* t_n*, (ii)* $W_{50/50}(t_n)$*, the wealth of the constant proportion portfolio at* t_n*, (iii) T-t, time remaining in the investment period. In the case that simple stochastic 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) .*

We remark that our feature set $F(t)$ for Example 1 is different from the features in Samo and Vervuurt (2016) which explicitly use security prices. Instead, at time t our feature set consists of the accumulated wealth at t from allocation strategy and benchmark strategy, which depend on the returns of traded assets from prior periods. Traded asset prices are not directly used as features for the neural network model. This is essentially because, at each rebalancing time, we search for the optimal adaptive strategy amongst all strategies with the current level of wealth. In addition, since we evaluate the performance of a trading strategy based on the terminal wealth $W(T)$ only, the trading decision at time t 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.

We use a 2-layer neural network as the functional form for the optimal control. As a result, the goal of the optimization problem is to find the optimal parameters of the neural network.

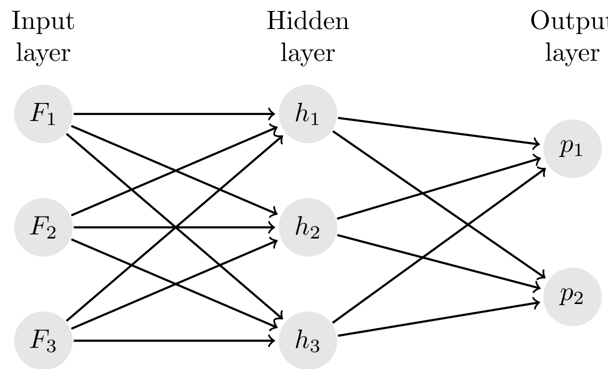


Figure 2.2: A 2-Layer NN representing the control functions

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 $F(t_n) \in R^D$ to the hidden nodes h. We use the sigmoid activation function,

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

and have

$$h_j(F(t_n)) = \sigma(F_i(t_n) z_{ij}) .$$

Here we use double summation convention, i.e.

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

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

$$\big(p(F(t_n))\big)_m = \frac{e^{x_{km} h_k(F(t_n))}}{\sum_i e^{x_{ki} h_k(F(t_n))}}, \; 1 \leq m \leq M .$$

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

$$0 \leq p(F(t_n)) \leq 1, \; 1^T p(F(t_n)) = 1 .$$

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

$$\begin{aligned} &for\ n = 0, 1, ..., N-1 \\ &\quad W(t_n^+) = W(t_n^-) + q(t_n) \\ &\quad W(t_{n+1}^-) = p(F(t_n))^T R(t_n) W(t_n^+) \\ &end\ . \end{aligned} \tag{2.6}$$

We approximate the expectation in equation (2.5) 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 $W^\ell(T), W_b^\ell(T)$ be the final wealth samples for the adaptive and benchmark strategies, obtained using equation (2.6), along the ℓ^{th} return sample path $R(t_n)^\ell, \; n = 0, 1, \ldots, N-1$.

Denote

$$g(x) \equiv \min\big(x,0\big)^2 + \max\big(x,0\big)\ . \tag{2.7}$$

The expectation in equation (2.5) is approximated by

$$\mathbb{E}\Big[g(W(T) - e^{sT}\cdot W_b(T))\Big] \simeq \frac{1}{L}\sum_{\ell=1}^{\ell=L} g(W^\ell(T) - e^{sT}\cdot W_b^\ell(T)) \tag{2.8}$$

Since the approximate function on the right hand side of (2.8) is a nonconvex, continuous but piecewise differentiable function of the NN weights, solving the optimization problem is challenging.

We recognize however that $\mathbb{E}\Big[g(W(T) - e^{sT}\cdot W_b(T))\Big]$ 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). In equation (2.8), we replace $g(x)$ by the smoothed approximation $\bar{g}(x)$, to obtain a continuously differentiable approximation,

$$\bar{g}(x) = \begin{cases} x, & \text{if } x > \epsilon\ , \\ \frac{x^2}{4\epsilon} + \frac{1}{2}x + \frac{1}{4}\epsilon, & \text{if } -\epsilon \le x \le \epsilon\ , \\ (x+\epsilon)^2, & \text{if } x < -\epsilon\ , \end{cases} \tag{2.9}$$

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

$$\min_{x,z}\ \frac{1}{L}\sum_{\ell=1}^{\ell=L} \bar{g}(W^\ell(T) - e^{sT}\cdot W_b^\ell(T))\ . \tag{2.10}$$

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

We note that Problem (2.10) is an unconstrained optimization problem with $H(D+M)$ 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 $\mathcal{O}(H(D+M)NL)$, or $\mathcal{O}(L)$ assuming a fixed NN model structure and fixed rebalancing schedule.

For the gradient evaluation, we note that

$$\nabla_{x,z}\left(\frac{1}{L}\sum_{\ell=1}^{\ell=L} \bar{g}(W^\ell(T) - e^{sT}\cdot W_b^\ell(T)\right) \tag{2.11}$$

$$= \frac{1}{L}\sum_{\ell=1}^{\ell=L} \nabla_{W^\ell(T)}\bar{g}\cdot\nabla_{x,z}W^\ell(T). \tag{2.12}$$

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

$$\nabla_{x,z}W^\ell(t_{n+1}^-) \tag{2.13}$$

$$= \nabla_{x,z}\Big(p(t_n)^T R(t_n)\big(W^\ell(t_n^-) + q(t_n)\big)\Big) \tag{2.14}$$

$$= \big(\nabla_{x,z}p(t_n)\big)^T \cdot R(t_n)W^\ell(t_n^-) + p(t_n)^T R(t_n)\cdot\big(\nabla_{x,z}W^\ell(t_n^-)\big). \tag{2.15}$$

Since the computational cost of evaluating $\nabla_{x,z}p(t_n)$ is $\mathcal{O}(H(D+M))$, we know from (2.13) that the computational cost of evaluating $\nabla_{x,z}W^\ell(T)$ is $\mathcal{O}(H(D+M)N)$. Therefore, the total computational cost for evaluating all gradients over L paths is $\mathcal{O}(H(D+M)NL)$.

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

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.

For training of the optimization problem (2.10), 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. 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 1 presents pseudocode for the stationary block bootstrap.

In Li and Forsyth (2019), the training dataset is generated using stationary block resampling with one expected blocksize and the testing dataset is generated with a different expected blocksize. As Politis and Romano (1994) points 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 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 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.

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

Algorithm 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 expected value
       of exp_block_size                                                   */
    blocksize = GeometricRandom( 1 / exp_block_size );
    for ( i = 0; i < blocksize; i = i + 1 ) {
        /* 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
```

datasets so that the learning algorithm may benefit from such an unfair edge. To address such concerns, we establish a theoretical bound on the probability of training and testing sample sequences being exactly the same.

Theorem 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 with a fixed blocksize of b_1 and path $\mathcal{P}_2$ be a bootstrap resampled 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.*

The proof of Theorem 1 is in Appendix A.1. To put this 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 case, in which the blocksizes are randomly generated from a shifted geometric distribution. We are able to establish the following theorem about the probability of two paths generated with stationary block bootstrap being identical.

Theorem 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*

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

The proof of Theorem 2 is also in Appendix A.1. 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 (2 years), for $N = 30 \times 12 = 360$ (30-year horizon) and $N_{tot} = 90 \times 12 = 1080$. Then the probability of a training path being identical to a testing path is 8.737×10^{-39}.

If training data set consists of a total of 100,000 training paths and testing data set consists of 10,000 testing paths, by union bound, the probability of existing a pair of training and testing path 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 dataset 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 1. *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 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 1.*

4 Performance Assessment and Comparison

We evaluate and report the performance of the proposed data-driven approach for outperforming a stochastic target in the context of a 30 year DC pension plan. In our numerical tests, we focus on portfolios with only two assets: a stock index and a bond index, as described in Example 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 $W_{50/50}(t)$.

4.1 Original Data and Its Augmentation

4.1.1 Historical Data

Our main objective here is to consider the core allocation problem between a risky 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.[4]. 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 in (Li and Forsyth, 2019), in the context of a fixed (non-stochastic) target based objective function, we 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 for either choice of risky and defensive asset. We have also carried out similar tests for our stochastic benchmark objective function. The results, using a ten year treasury as the defensive asset, can be found in Appendix A.4. For simplicity here, we will focus on the CRSP index and the 3-month T-bill case in the main article.

[4]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 article. This service and the data available thereon constitute valuable intellectual property and trade secrets of WRDS and/or its third-party suppliers.

For illustration, we consider here a two-asset allocation in which the wealth of the portfolio is allocated to the two indexes. We subsequently refer to the two assets simply as the stock and the bond.

For the stock index and bond index, Table 4.1 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 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 4.1.

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 4.1: 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 $\hat{b}$.

4.2 Experiment Setting

As discussed in Section 1.3, we use an example of an investor in a DC plan to illustrate the application of data-driven methodology. In the numerical example, we assume an investor starts with zero wealth (balance) in the DC plan, and makes a real cash injection of 10 per year[5] 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 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;
- 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 $e^{sT}W_{50/50}(T)$, 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.

[5] We will use thousands of dollars as our unit of wealth

We want to remark that in this numerical example, we are assuming annual contributions of a fixed dollar amount. We are aware that many pensioners care about the replacement rate (percentage annual employment income replaced by retirement income) which measures how well retirees can maintain their lifestyles in retirement. In fact, depending specific assumptions about the salary, one 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 4.3.2 which shows that an investor can expect to achieve a decent level of replacement income in 30 years following our DC plan strategy.

4.3 Assessment of Results

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 $\hat{b} = 2$, which is the average optimal blocksize. When discussing robustness in Section 5.1, we show that the strategy performance using alternative training-testing expected blocksize pairs is qualitatively similar.

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 4.2: 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.

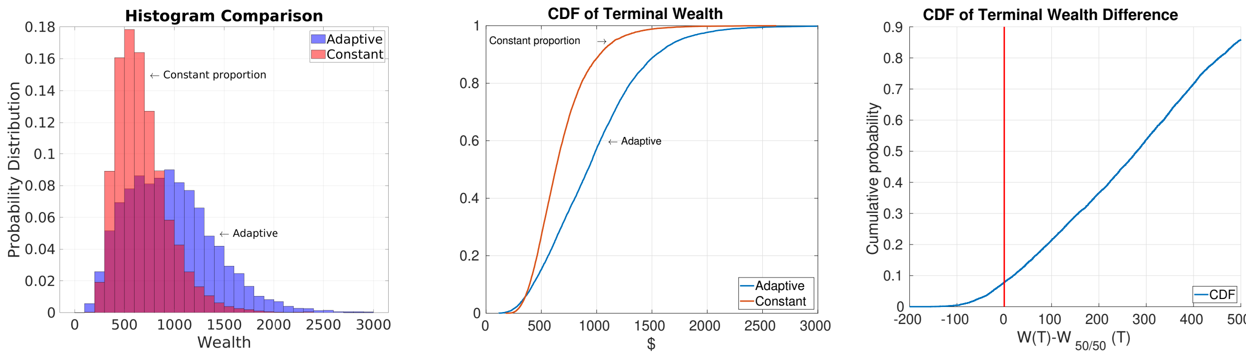
Table 4.2 reports performance statistics and the probability of the terminal wealth less than the median of the terminal wealth of both strategies. From Table 4.2 , we observe that

- The median and mean of the optimal adaptive strategy is 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 4.1a 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., 2019). 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. We also plot the CDF plot for the optimal adaptive strategy and the constant proportion strategy in Figure 4.1b.

We note that the terminal wealth distribution of the optimal adaptive strategy has a slightly worse left tail than the constant proportion strategy. The 95% VaR of terminal wealth is 326 for the optimal adaptive strategy and 338 for the constant proportion strategy.[6] In fact, from Table 4.3 we can see that the adaptive strategy has worse 95% and 99% VaR and CVaR than the constant proportion strategy.

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



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

(b) CDF of terminal wealth adaptive strategy and constant proportion strategy

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

Figure 4.1: 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)

VaR and CVaR on Testing Data				
Strategy	95% VaR	95% CVaR	99% VaR	99% CVaR
constant proportion($p = 0.5$)	338	294	265	238
adaptive	326	253	201	169

Table 4.3: VaR and CVaR of 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. We recall that the objective function in (2.5) 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 4.1c 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 4.1c we can see 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.

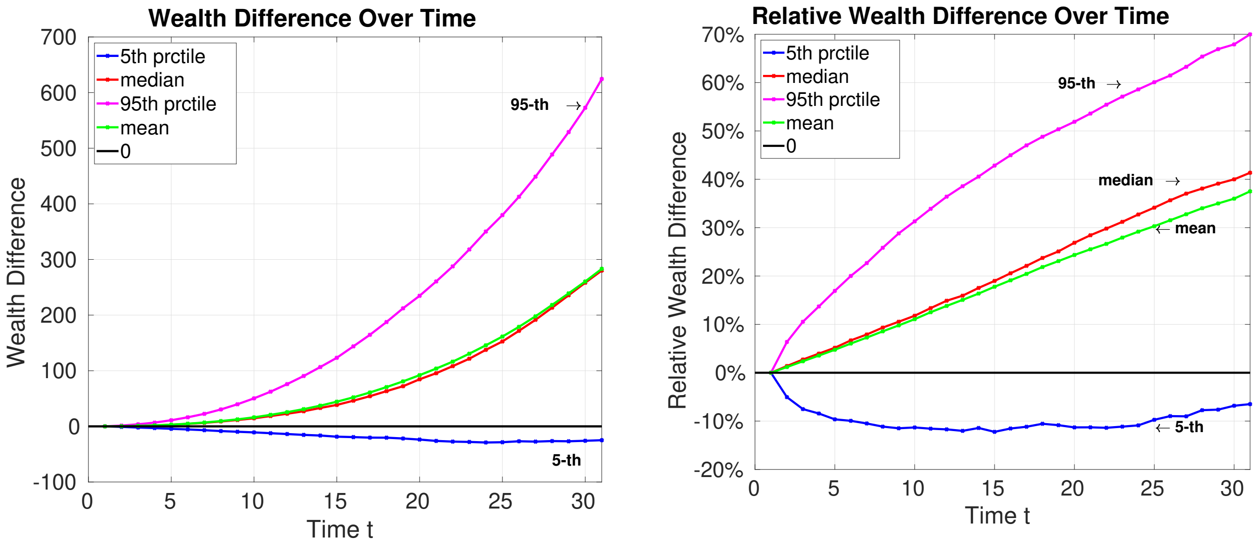
If reducing the tail risk has a higher priority in the investment plan, a tail riskmeasure CVaR can be included in the objective function accordingly. This, of course, will produce a lower probability of pathwise outperformance over the benchmark strategy. However, the proposed framework can be similarly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

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

4.3.1 Strategy Performance Over Time

Since the objective function for the optimal control (2.5) is defined from the terminal wealth, we examine how the optimal adaptive strategy performs over the entire period of investment.

Figure 4.2 graphs the average and various percentiles of the wealth difference $W(t) - W_{50/50}(t)$ in the investment time horizon. From Figure 4.2, we observe that



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

- 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 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 our objective function to focus on achieving this goal.

Figure 4.2 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.

4.3.2 Replacement Rate Example

One common measure to determine the effectiveness of a pension system is the replacement rate, which is the percentage 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 commonly 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 US it would be the 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[7], which translates to

[7] Wealthsimple, www.wealthsimple.com/en-ca/learn/how-much-cpp-retirement

40% of replacement rate based on the average income of $49,000 in Canada[8]. In the United States, an earlier study shows that Social Security benefits provides about 40% of replacement income according to Biggs and Springstead (2008). However, a more recent study by Ghilarducci et al. (2017) shows that the replacement rate from Social Security for middle income employees ($40,000 - $115,000) is only 29%. Nevertheless, Social Security is still a significant source of the retirement income in the United States.

Consider the example of an employee making $75,000 per year in Canada (which is well above national average of $49,000) and contributing $10,000 per year (total employee and employer contribution) to the savings plan. According to our numerical results in Table 4.2, the employee can expect a median terminal wealth over $900,000 following our 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 national average, so 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).

4.3.3 Strategy Characteristics

We further examine the characteristics of the optimal adaptive strategy. Figure 4.3a 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%.

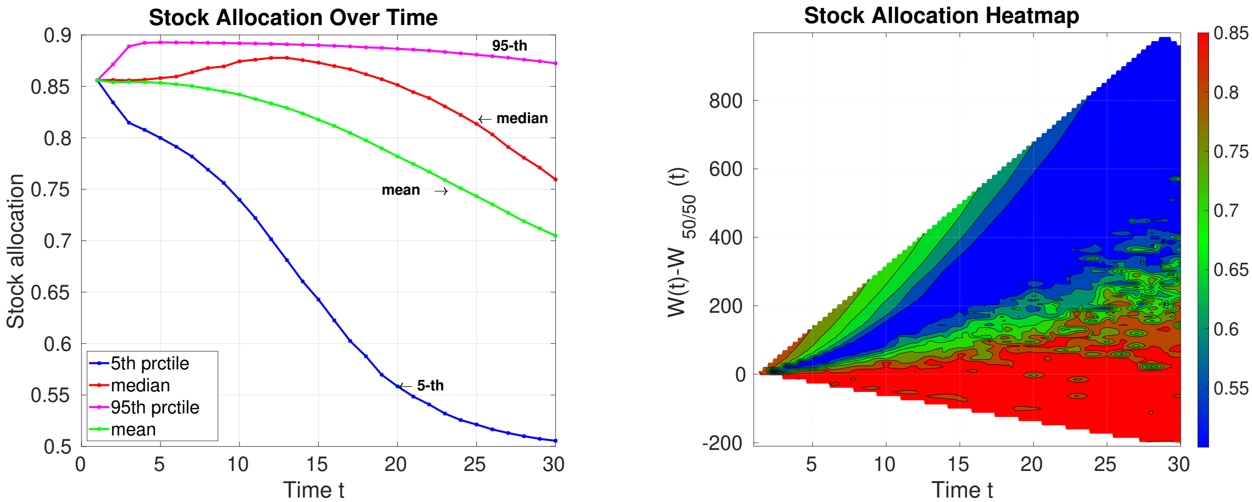
With a red-blue color scheme, Figure 4.3b shows the heatmap of the stock allocation with respect to time t and the wealth difference $W(t) - W_{50/50}(t)$. Darker shades of the red color indicate more allocation in stocks and darker shades of the blue color indicate more allocation in bonds.

From Figure 4.3b, 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 seen in Figure 4.2), decreasing the stock 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 $W(t) - W_{50/50}(t) < 0$ (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.

[8] Government of Canada, www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=1110023901



(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 4.3: Fraction invested in stocks over time for the optimal adaptive strategy: percentiles and the heatmap

4.3.4 Historical Backtest Performance

As a special out-of-sample test, we consider the actual 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 4.4, we see 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 see that historically the optimal adaptive strategy does better than the constant proportion strategy.

Note that the adaptive strategy does show a large drawdown in 2002 and 2008. However, our objective function is posed in terms of outperformance of the terminal wealth. We see 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 4.4 illustrates the time evolution of the stock allocation on the historical path. 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.

4.4 Comparison with the 80/20 Constant Proportion Strategy

While the average stock allocation from the optimal adaptive strategy varies over time, its average over time is about 80%. A natural question is how the optimal adaptive strategy compares with the 80/20 constant proportion strategy which invests 80% of the wealth in the stocks and 20% in the bonds.

Here we compare the optimal adaptive strategy with the 80/20 constant proportion strategy. Recall that in Section 4.3, the optimal adaptive strategy is trained on bootstrap resampled data with the expected blocksize of 0.5 years and the test dataset 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 dataset.

In Figure 4.5, 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 adaptive strategy and the 80/20 strategy from the 50/50 strategy respectively.

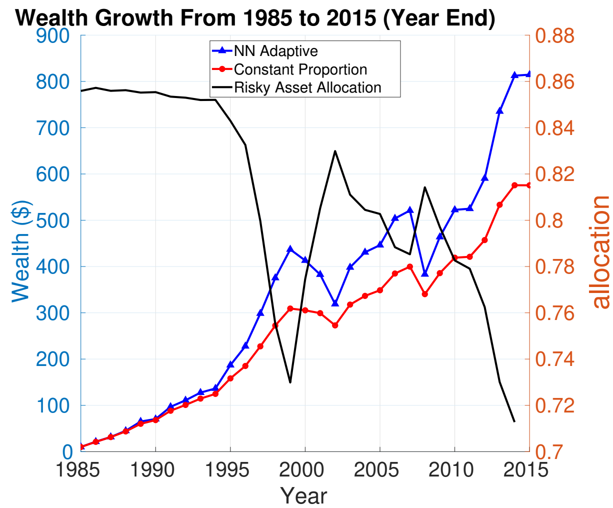
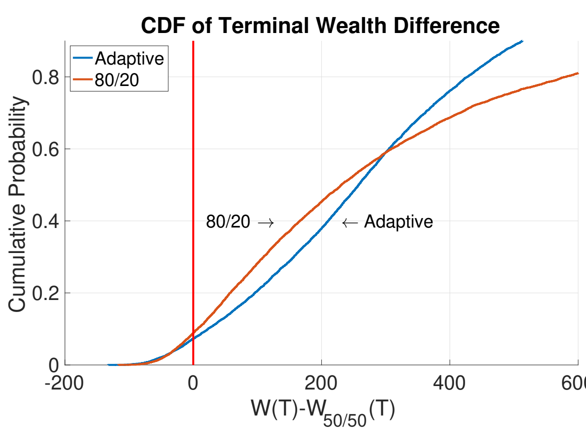
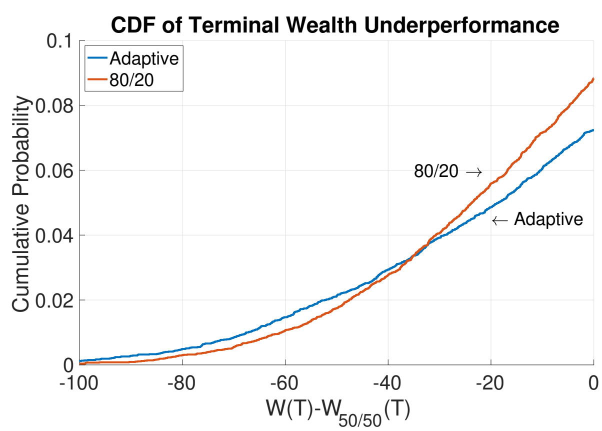


Figure 4.4: Backtest of strategy performance over the historical period from 1985-2015 (single path)



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

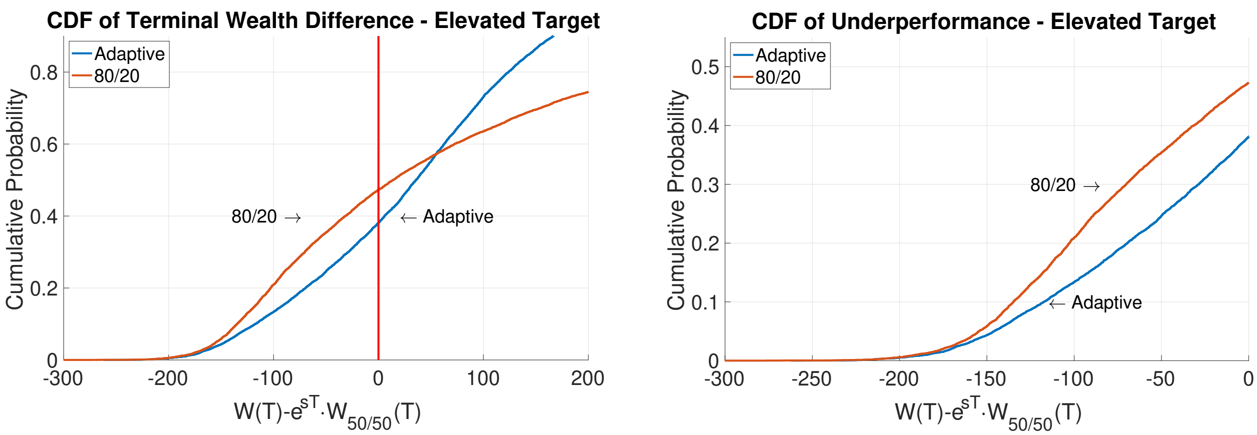


(b) CDF of terminal wealth difference - enlarged for underperformance

Figure 4.5: CDF of wealth difference of both strategies (optimal adaptive and 80/20 constant proportion) over the 50/50 strategy

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 the 80/20 strategy. When underperformance against the 50/50 strategy occurs, the magnitude of underperformance for the optimal adaptive strategy is less than the magnitude of underperformance for the 80/20 strategy, as in Figure 4.5.

It is worth noting that the 80/20 strategy has more upside than the optimal adaptive strategy. However, we should remind the readers that less upside is a natural result of our choice of the double-sided penalty objective function. As reflected in the asymmetric objective function, our goal is not to achieve extremely large outperformance over the 50/50 strategy, but to reach the elevated 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 4.6b.



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

We also observe that the optimal adaptive strategy has a smaller 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 4.6b that the optimal adaptive strategy consistently 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.

5 Robustness Assessment

To further evaluate the robustness of the optimal adaptive strategy, we assess optimal control models from the following three perspectives:

- We test the strategy learned from the bootstrap data with a given expected blocksize on bootstrap data with multiple different expected blocksizes.
- We train the model on a dataset simulated from a synthetic parametric model and test it on the bootstrap resampled dataset.
- We train the strategy learned on bootstrap data from one segment of the historical data and test the strategy on bootstrap data from another segment of the historical data.

We generate the bootstrap resampled data by sampling directly from the specified historical data sequence for training the optimal control model.

5.1 Testing Using Different Blocksizes

We test the adaptive strategy learned on bootstrap resampled data with a given blocksize on bootstrap resampled data with different blocksizes.

For illustration, here we only show the testing results of the strategy learned on bootstrap resampled data with expected blocksize of 0.5 years, where test data sets are bootstrap resampled data with blocksizes ranging from 1-10 years. We note that training on data sets using a different blocksize, and testing on other blocksizes produces qualitatively similar results.

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 5.1: 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 5.1 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 the terminal wealth higher than the median terminal wealth of the constant proportion strategy.

Table 5.1 demonstrate that the outperformance of the adaptive strategy over the benchmark strategy is robust across different expected blocksizes. We include more testing results from strategies trained with other expected blocksizes in Appendix A.5.

5.2 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.

5.2.1 Synthetic Data Generation

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 the 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

$$f(y) = p_{up}\eta_1 e^{-\eta_1 y}\mathbf{1}_{y\geq 0} + (1-p_{up})\eta_2 e^{\eta_2 y}\mathbf{1}_{y\leq 0}. \tag{5.1}$$

For future reference, note that

$$E[y = \log \xi] = \frac{p_{up}}{\eta_1} - \frac{(1-p_{up})}{\eta_2},\ E[y = \xi] = \frac{p_{up}\eta_1}{\eta_1 - 1} + \frac{(1-p_{up})\eta_2}{\eta_2 - 1} \tag{5.2}$$

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

$$\frac{dS(t)}{S(t^-)} = (\mu - \lambda E[\xi - 1])dt + \sigma dZ + d\Big(\sum_{i=1}^{\pi_t}(\xi_i - 1)\Big), \tag{5.3}$$

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 (5.1). 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

$$dB(t) = rB(t)dt, \tag{5.4}$$

where r is the (constant) rate. This is obviously a simplification of the real bond market. We remind the reader that, ultimately, our NN method is entirely data-driven, and will be based on bootstrapped stock and bond indexes.

Based on (5.3) and (5.4), 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 is provided in Table 5.2.

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 5.2: 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.

5.2.2 Strategy Performance

We test the performance of the strategy trained on synthetic data on bootstrap data with expected blocksize $\hat{b} = 2$ years. 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 5.3: 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 5.3 shows that the adaptive strategy learned from synthetic data performs well on the test set of bootstrap resampled data. The adaptive strategy have 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, overall, the strategy appears to be quite robust. Further distribution comparisons can be found in Appendix A.6.

5.3 Robustness Test with Training/Testing Split

In §5.1 and §5.2, both training and testing datasets are generated from either a parametric model or 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 any sequential ordering information is destroyed. Further, Theorem 1 and 2 show that the probability of an entire path in the training dataset reappearing in the testing dataset 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: 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: 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 5.1 and Figure 5.2 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.

Note that, for Case #1 and #2, the underlying historical data for testing data has only 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 for the non-overlap window, we will reduce the investment horizon to $T = 15$ years, for both cases in this section.

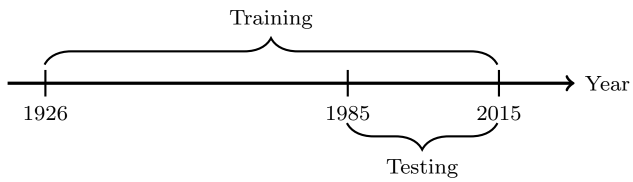


Figure 5.1: Case #1: use historical data from 1926-2015 for generating training data, and 1986-2015 for testing. There is an overlap between the underlying historical paths for training and testing.

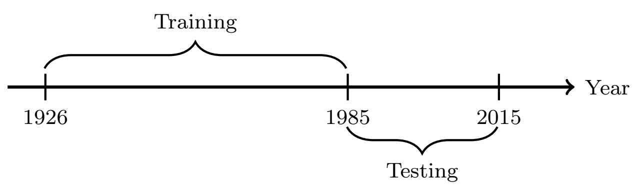
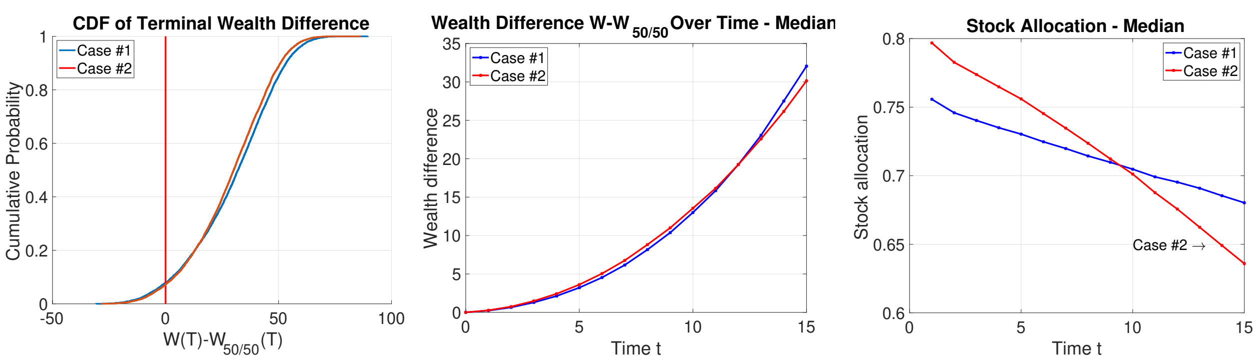


Figure 5.2: Case #2: “non-overlap” case where underlying market data for training and testing data has no overlaps. Case #2 uses the same testing dataset as case #1.



(a) CDF of wealth difference $W(T) - W_{50/50}(T)$ for the two cases

(b) Median of wealth difference $W(T) - W_{50/50}(T)$ for the two cases

(c) Median stock allocation for the two cases

Figure 5.3: Comparison of testing results for the two cases: case #1: train: 1926-2015, test: 1986-2015; case #2: train: 1926-1985, test: 1986-2015.

We first compare the CDF of the terminal wealth of the two cases. From Figure 5.3a we can observe that Case #1 and Case #2 have almost identical CDF curves. The almost identical CDF curves for Case #1 and Case #2 (the *non-overlap* case) - supports our 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 have a superior CDF than Case #2, in which the underlying market data for training data and testing data have no overlaps. In Figure 5.3b, we show the median of wealth difference between the adaptive strategy and the constant proportion strategy for both cases. Again, Case #1 and Case #2 have almost identical performances, despite that the median stock allocation for the two cases are slightly different, as shown in Figure 5.3c. In fact, we find that the different percentiles of the two cases are really close to each other, and that Case #2 has slightly better tail risk control than Case #1 (5th percentile), which further proves that the overlap does not introduce performance advantage as the *non-overlap* case actually has less tail risk. The percentile results can be found in Appendix A.7.

In conclusion, the results further illustrate the robustness of our approach and show that forward-looking bias is not a concern in our work.

6 Target-date Funds as Benchmark

In recent years, target-date funds (also known as life-cycle funds) have gained much popularity amongst investors. Target-date funds operate with the premise that the investor retires at a certain target date. The fund adjusts the asset allocation as the calendar time gets closer to the target date. Often, the fund allocates between a stock fund and a bond fund.

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 6.1)[9]. 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.

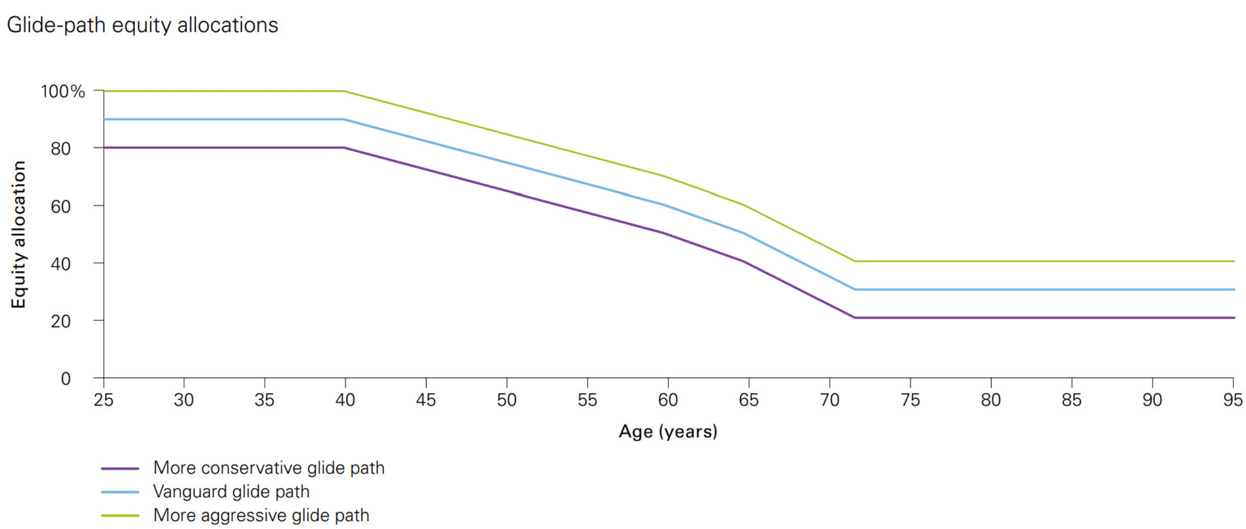


Figure 6.1: Target-date fund stock allocation from Vanguard (Donaldson et al., 2015)

[9]For simplicity, we have lumped together US and International stocks as an allocation to stocks, and the total allocation to US bonds, international bonds and TIPS as an allocation to bonds

6.1 Target-date Fund and Constant Proportion

However, recent research, based on empirical (Esch and Michaud, 2014; Arnott et al., 2013) and theoretical work (Graf, 2017; Forsyth and Vetzal, 2019) suggests that the purported advantages of target-date funds may have been oversold. This research indicates that the terminal wealth distributions of a deterministic glide-path and a constant proportion strategy having 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 6.1 (based on bootstrap resampling).

We then empirically computed 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 6.2, 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.

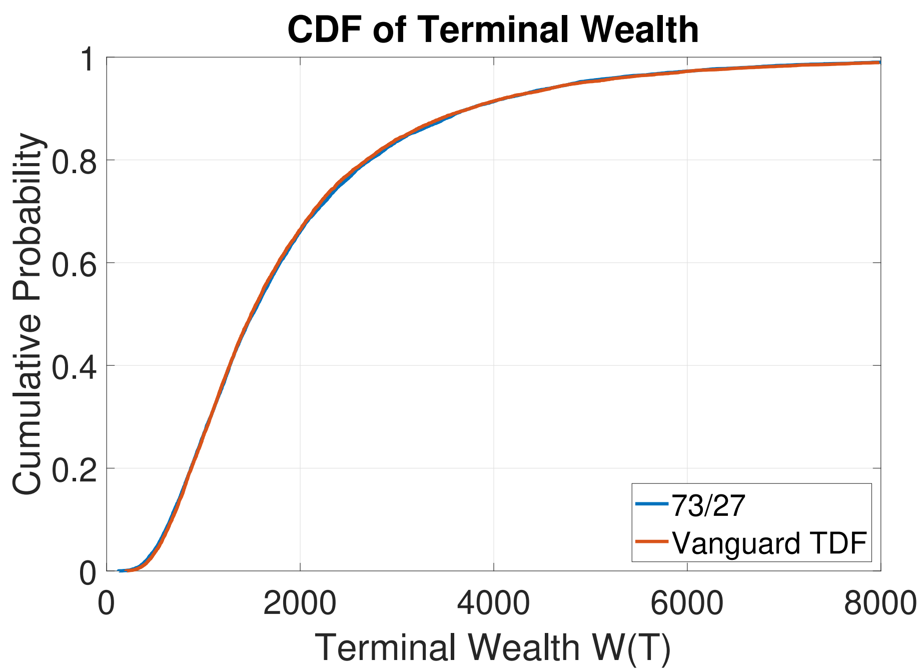


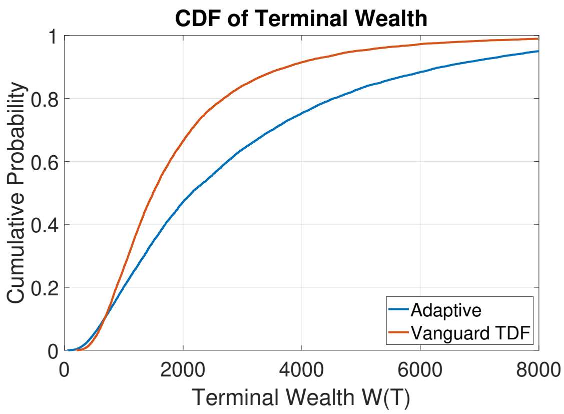
Figure 6.2: CDF of terminal wealth of Vanguard TDF and 73/27 strategy

6.2 Outperforming the Vanguard Target-date Fund

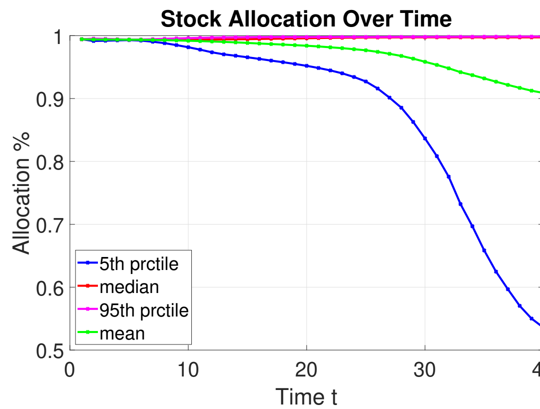
Nevertheless, outperforming the Vanguard target-date fund or the 73/27 constant proportion strategy appears to be very challenging, if we retain our constraint that use of leverage is not permitted. This is simply because such strategies are already heavy in stocks, and thus inevitably the learned strategy needs to be heavier in stocks in order to achieve a higher expected terminal wealth, but the 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 objective function to be 50 basis points. We can reasonably argue that the learned adaptive strategy has a more attractive terminal wealth distribution compared to the Vanguard target-date fund since the CDF of the adaptive strategy shown in Figure 6.3a is more right-skewed with a slightly worse left tail. However, we can also observe from Figure 6.3b 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 so adaptive as we expect[10]. 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 use of leverage. However, this is usually not advisable in a retirement savings account.



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



(b) Stock allocation of learned adaptive strategy

Figure 6.3: 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.

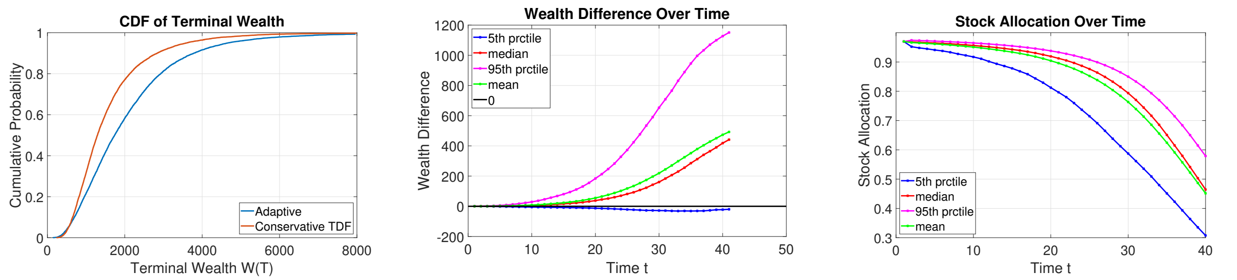
6.3 Outperforming a Conservative Target-date Fund

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 Vanguard report (Donaldson et al., 2015) and described as a *more conservative* target-date fund. We also note that even in 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 objective function (2.5) to be 50 basis points. As in Section 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.

First and foremost, we can observe from Figure 6.4a that the learned adaptive strategy has a more right-skewed terminal wealth distribution than the conservative target-date fund. When we examine the actual allocation of the adaptive strategy in Figure 6.4c, we can see that the adaptive strategy tends to hold

[10]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 2.5 and in terms of tail risk.



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

(b) Wealth Difference between adaptive strategy and conservative TDF

(c) Stock allocation of learned adaptive strategy

Figure 6.4: 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.

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 derisks (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 in this section that:

- Outperforming a deterministic glide path target-date fund in terms of terminal wealth distribution is essentially the same problem as outperforming a constant proportion strategy.
- 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 fraction in stocks of about 68%.

7 Limitations

A common limitation of machine learning applications in finance is the lack of a theoretical performance guarantee. Unfortunately, in our case, there is also no theoretical guarantee on whether the trained strategy has really converged to the theoretical optimal strategy. However, the relative performance nature of this specific use case compensates this limitation to a certain degree, as one can always empirically compare the learned strategy with the benchmark strategy.

The bootstrap resampling method we use in this framework may also prevent the application of our methodology in a wide range of investment problems. Bootstrap resampling requires a long data history and enough data points. Depending on the nature of the problem, using bootstrap resampling may not always be feasible and could largely limit the choice of the asset basket. For example, in an asset allocation problem where assets are single name stocks, it is likely that the stocks have different length of history. Bootstrap resampling is not easily modifiable to account for mismatches and gaps in the historical individual stock data.

8 Conclusions

In this article, we propose a data-driven framework for computing the optimal asset allocation for outperforming a stochastic benchmark target based on market asset return observations. The scenario-based dynamic asset allocation problem is solved directly assuming a neural network representation for the optimal control, without using dynamic programming. This leads to a method that avoids the curse of dimensionality which is a critical issue in dynamic allocation for outperforming a stochastic benchmark.

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

We use block bootstrap resampling to augment historical financial market data. The training data is 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 make a parametric asset price model as well as model parameter estimations. We further provide mathematical justifications for using block bootstrap resampling to generate both training and testing datasets.

The proposed method is illustrated in the DC pension allocation problem, which is a practically relevant and important problem on its own. We evaluate and analyze the performance of the optimal NN adaptive strategy based on 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*. Our method is straightforward to use for portfolios with more assets. We include an example with three month T-bills, 10 year treasuries, and a capitalization weighted CRSP index in Appendix A.3.

We illustrate the robustness of our approach from three different perspectives.

- We show that the adaptive strategy trained on bootstrap resampled data with a given expected blocksize performs consistently well on bootstrap resampled data with different expected blocksizes (thus different distributions).
- We show that the adaptive strategy learned on synthetic data performs well on bootstrap resampled data, despite the fact that the methodology for generating the datasets are quite different.
- We compare the performance of our strategy with the strategy learned in an *non-overlap* setting where the underlying market data for the training dataset and testing dataset has no overlap. We show that the *non-overlap* case has a comparable performance which supports our argument that forward-looking bias should not be a concern in our approach.

We remark here that results we have obtained in this article are based on the assumption that the training and testing datasets have similar distributions. In recent years we have observed the slowing down of economic growth globally, and many worry that the COVID-19 pandemic could bring an irreversible impact on the global economy. Others believe that the constantly decreasing interest rates and the unprecedented negative rates will attract more funds to stocks from fixed income investments, and lead to the further widening of the yield spread between stocks and bonds. Note that our historical data was based on the years 1926-2015, which encompasses the great depression, a world war, periods of high inflation, the dot-com bubble and the financial crisis of 2008. This data, which we use for training, certainly contains many difficult periods for investors. While it is certainly true that an optimal strategy learned from past data may not be optimal if the future financial market behaves significantly differently from the past, we should perhaps recall the quote

> *The four most expensive words in the English language are: "This time it's different."* (Sir John Templeton)

In summary, we cannot predict the future, and the best we can do is to prepare for the future by learning from history.

Basing our optimal control on a shallow Neural Network representation using only a small number of financially relevant feature variables results in a strategy that is financially intuitive and implementable.

9 Acknowledgements

This work was supported by a Collaborative Research and Development (CRD) grant from the Natural Sciences and Research Council of Canada NSERC Neuberger Berman CRD: #50492-10196 -2950-105 and by a grant from Neuberger Berman.

10 Conflicts of interest

The authors have no conflicts of interest to report.

A Appendix

A.1 Proofs for Theorem 1 and 2

We mathematically establish Theorem 1 and 2.

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

$$\begin{aligned} \hat{b} &= \text{expected blocksize in stationary block bootstrap} \\ N &= \text{number of total datapoints in the path} \\ N_{tot} &= \text{number of total datapoints to bootstrap from} \\ \mathcal{P}[i] &= \text{the } i\text{th data point in path } \mathcal{P} \end{aligned} \tag{A.1}$$

We also make the following definitions.

Definition 1. *Assume that a path $\mathcal{P}$ of length N, which contains blocks $[B_1, \ldots B_k]$, is resampled from the original data path of length N_{tot}. The **decision index list** $[I_1, \ldots, I_k]$ of the path $\mathcal{P}$ is defined as the list of starting indices of every block in the resampled path with $I_1 = 1$, $I_i = 1 + \sum_{j=1}^{i-1} |B_j|$, $i = 2, \ldots k$, 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 (Decision Index List Example)**.** *Given a decision index list $[I_1, \ldots, I_k]$, 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. *For any two paths $\mathcal{P}_1$ and $\mathcal{P}_2$, 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 $[I_1, \ldots, I_p]$ retains the order properties of the original lists, i.e. $I_{i+1} > I_i$ and $I_{p+1} = N + 1$.*

Definition 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 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 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 1. Consider fixed block resampling sequences of N points 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} the total number of combined decision points and I_j denoting the jth index within I.

From Lemma 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 $\mathbb{P}(\mathcal{P}_1[I_j] = \mathcal{P}_2[I_j]) = \frac{1}{N_{tot}}$. In addition

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

.

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 $(\frac{1}{N_{tot}})^{lcm(\frac{N}{b_1},\frac{N}{b_2})}$. □

Next, we consider the stationary block bootstrap resampling, in which the blocksizes are randomly generated from a shifted geometric distribution.

Properties 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*

$$\begin{aligned}\mathbb{P}[m = k] &= (1-p)^{k-1}p \\ \mathbb{P}[m \geq k] &= (1-p)^{k-1}\ .\end{aligned} \tag{A.2}$$

We rewrite equation (A.2) in a form amenable to manipulation. Let

$$(1-p) = e^{-\lambda}\ , \tag{A.3}$$

so that equation (A.2) becomes

$$\begin{aligned}\mathbb{P}[m = k] &= e^{-\lambda k}(e^{\lambda}-1) \\ \mathbb{P}[m \geq k] &= e^{-\lambda(k-1)} \\ \lambda &= -\log[1-p]\ .\end{aligned} \tag{A.4}$$

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

$$\lambda = -\log\left[1 - \frac{1}{\hat{b}}\right]\ . \tag{A.5}$$

Lemma 2. *Suppose $[I_1, \ldots, I_k]$ be 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 1,

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

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

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

□

Lemma 2 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 k of the decision index list.

Lemma 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*

$$\begin{aligned} \mathbb{P}(N_{cdi}(\mathcal{P}_1, \mathcal{P}_2) = k) \quad &= \quad \binom{N-1}{k-1} e^{-(\lambda_1+\lambda_2)(N-1)}(e^{\lambda_1+\lambda_2} - 1)^{k-1} \\ & \lambda_1 = -\log\left[1 - \frac{1}{\hat{b}_1}\right] \; ; \; \lambda_2 = -\log\left[1 - \frac{1}{\hat{b}_2}\right] . \end{aligned} \tag{A.6}$$

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).

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 non-overlapped indices in $\mathcal{P}_1$, the probability of a combined decision index list $[I_1, \ldots, I_k]$ occurring equals

$$\sum_{v=1}^{k} \Big(\binom{k-1}{v-1} \sum_{i=0}^{k-v} \binom{k-v}{i} f(\hat{b}_1, v+i) f(\hat{b}_2, k-i) \Big). \tag{A.7}$$

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 1 and Lemma 3, we establish the probability of two paths generated with stationary block bootstrap resampling being identical.

THEOREM 2. Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two paths of the length N generated from the stationary block bootstrap resampling from a sequence of N_{tot} distinct observations with the expected blocksizes of $\hat{b}_1$ and $\hat{b}_2$ respectively. The probability of $\mathcal{P}_1$ and $\mathcal{P}_2$ being identical is

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

Proof. Using Lemma 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 index 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 3, we have

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

□

A.2 Results from Symmetric Quadratic 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 $\mathbb{E}\Big[\big(W(T) - e^{sT} \cdot W_b(T)\big)^2\Big]$.

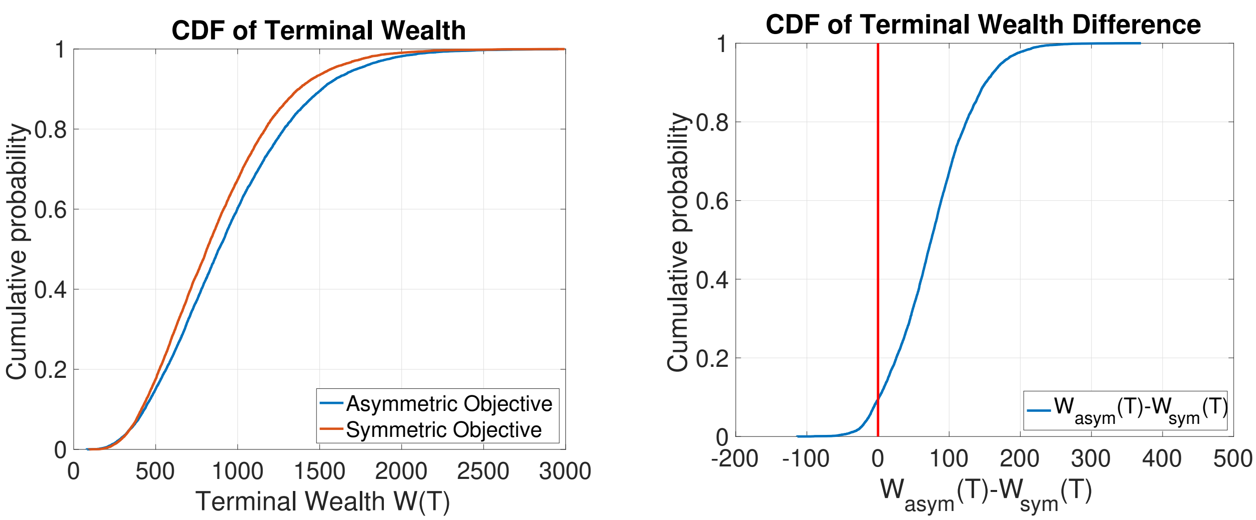
We train two adaptive strategies under our proposed asymmetric objective function (2.5) and the quadratic symmetric objective function with the same bootstrap resampled dataset (expected blocksize of 0.5 years), and test the two strategies on the same bootstrap resampled dataset with expected blocksize of 2 years. The following results are all 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. In fact, if we compare the path-wise terminal wealth, as shown in Figure A.1b, we can clearly 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 terminal wealth of both objective functions

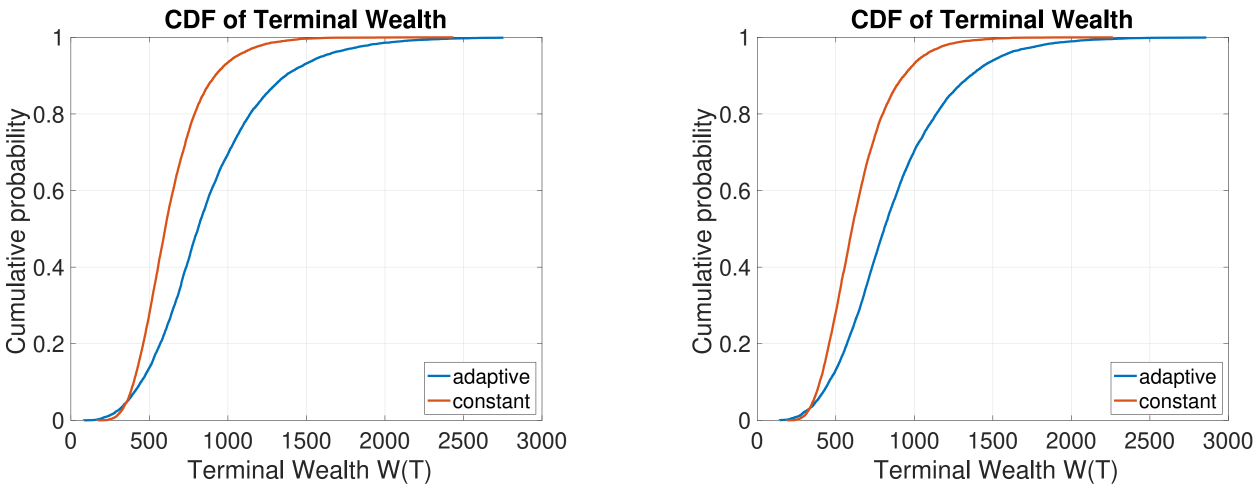
(b) Wealth difference between two objective functions

Figure A.1: Terminal wealth and terminal wealth difference, comparing symmetric and asymmetric objective functions.

A.3 The Three-asset Case

While we only present an example with two assets in the main article, 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.

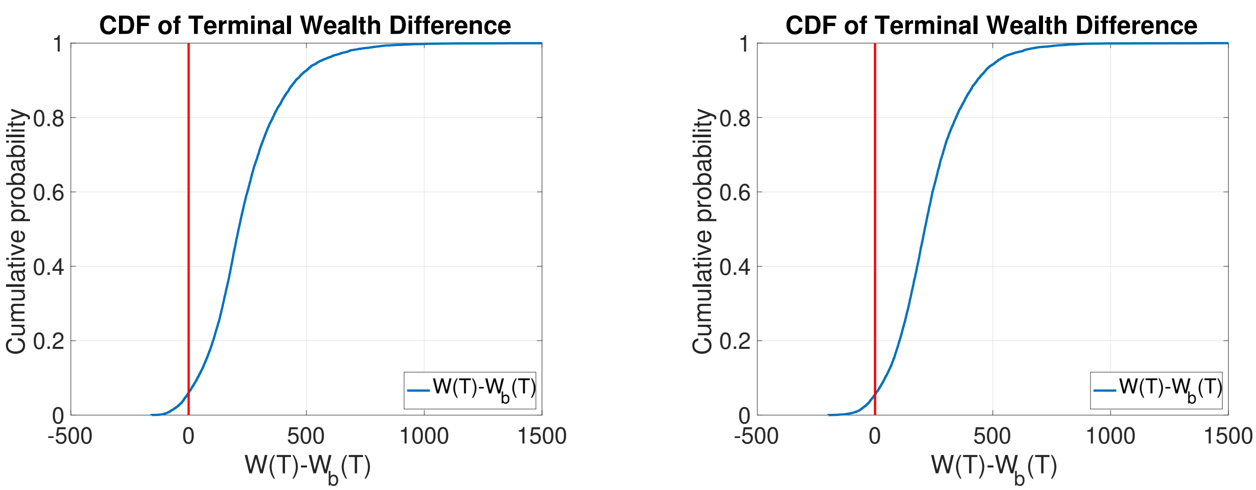
We train the neural network model on bootstrap resampled data with an expected blocksize of 0.5 years, with the proposed asymmetric objective function 2.5. We then test the learned adaptive strategy on bootstrap resampled data with an expected blocksize of 2 years.



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

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

Figure A.2: 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 A.3: CDF of terminal wealth difference for the 3 asset case, where $W_b(T)$ indicate the terminal wealth of the constant proportion benchmark strategy

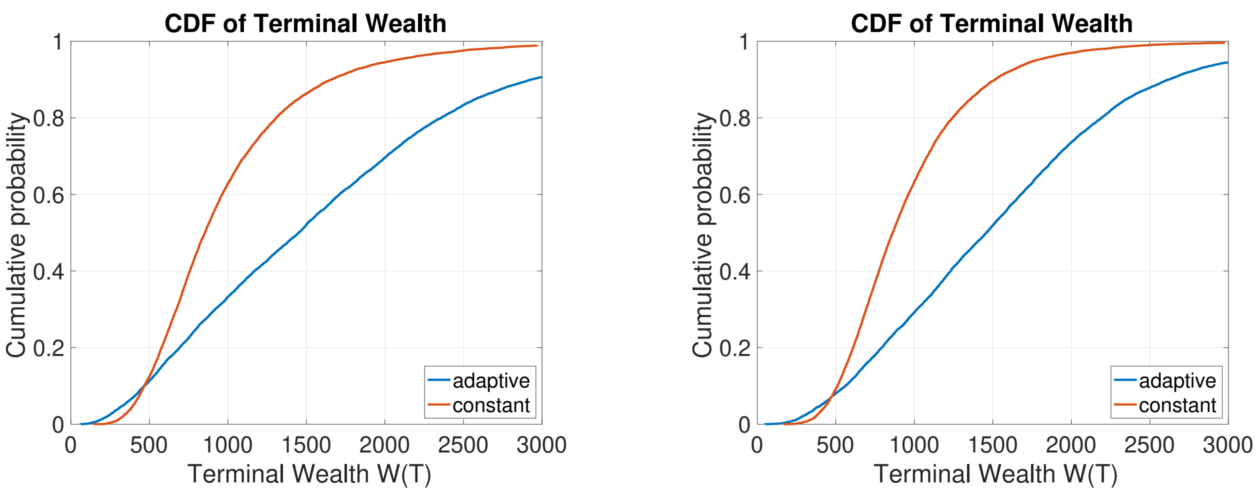
We can see from the Figure A.2 that the adaptive strategy has a consistently more right-skewed distribution of the terminal wealth compared with the constant proportion benchmark strategy. The path-wise comparison of terminal wealth difference also shows consistent outperformance compared to the adaptive 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 work.

A.4 Results from Alternative Datasets

Here we show the results based on alternative historical datasets - the equal-weighted CRSP stock index and the 10-year treasury bond index. The historical outperformance of equal-weighting has been attributed to such portfolios having higher exposure to value, size, and market factors (Plyakha et al., 2014). While historically the 10-year (real) T-bond has not always had the same behavior as the 3-month T-bill, we find that the learned adaptive strategy has also consistently outperforms the benchmark strategy on the alternative datasets.

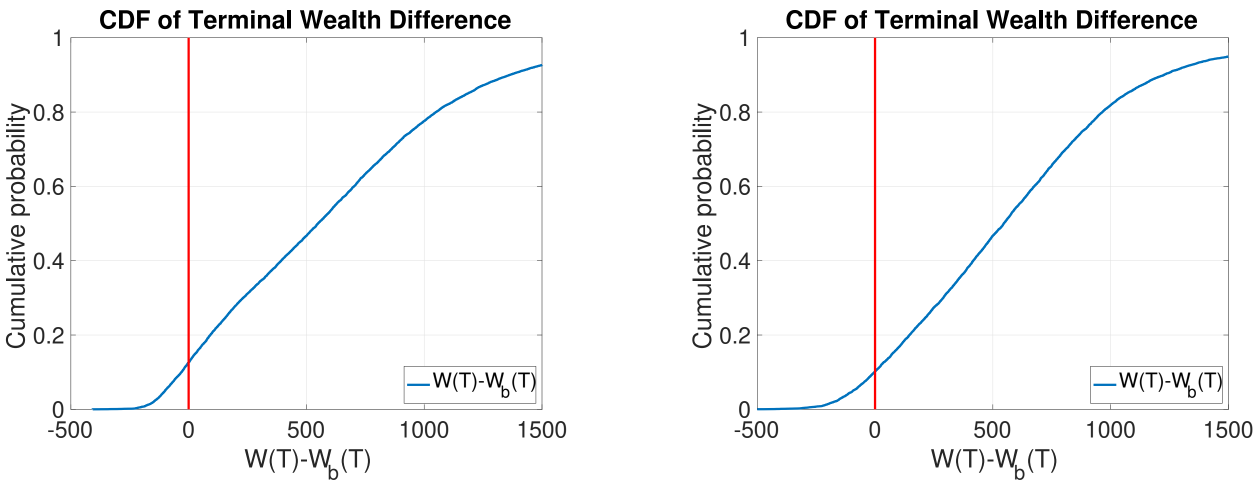
We train the neural network model on bootstrap resampled data from the alternative datasets with an expected blocksize of 0.5 years, with the proposed asymmetric objective function 2.5. We then test the learned adaptive strategy on bootstrap resampled data from the alternative datasets with an expected 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 A.4: 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 A.5: 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

From Figure A.4, we can clearly see that the learned adaptive strategy has a more right-skewed terminal wealth distribution in both training and testing. From Figure A.5, we can see that the adaptive strategy outperforms the benchmark strategy with more than 90% probability in both training and testing. Such results show us that the framework is capable of learning a good adaptive strategy that outperforms the benchmark strategy with different underlying historical datasets.

A.5 Additional Robustness Testing Results

As mentioned in section 4.3, we only showed terminal wealth statistics for the strategy trained with bootstrap resampled with expected blocksize $\hat{b} = 0.5$ years. Here we show the testing performance of strategies trained on bootstrap data with different blocksizes on different testing sets (bootstrap resampled from different blocksizes). The results show that the adaptive strategy consistently outperforms the constant proportion strategy.

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

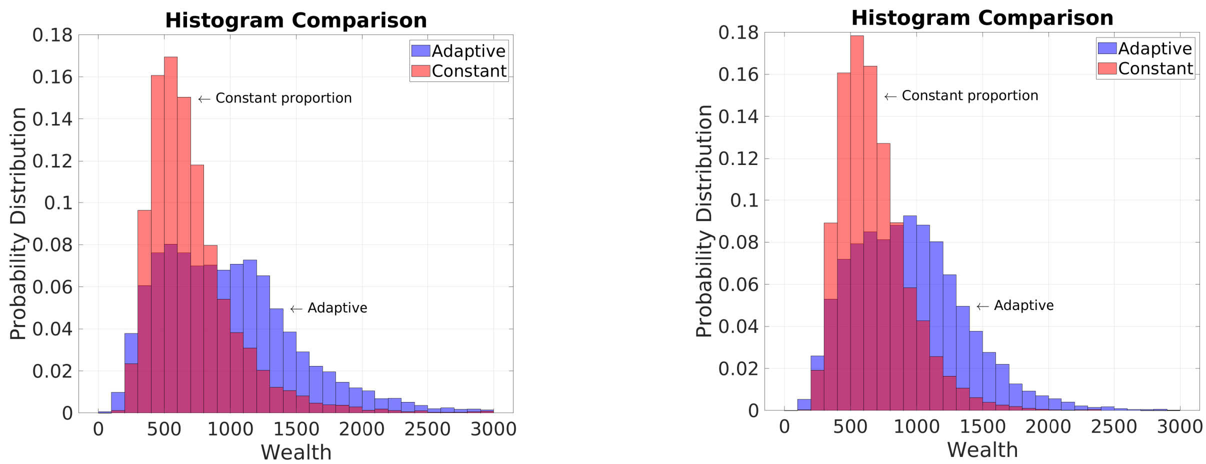
Table A.6: Trained on bootstrap resampled data with $\hat{b} = 10$ years

A.6 Robustness: Distribution Comparison Based on Test Results From the Synthetic Model

We observe from Figure A.6 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 in both training and testing.

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

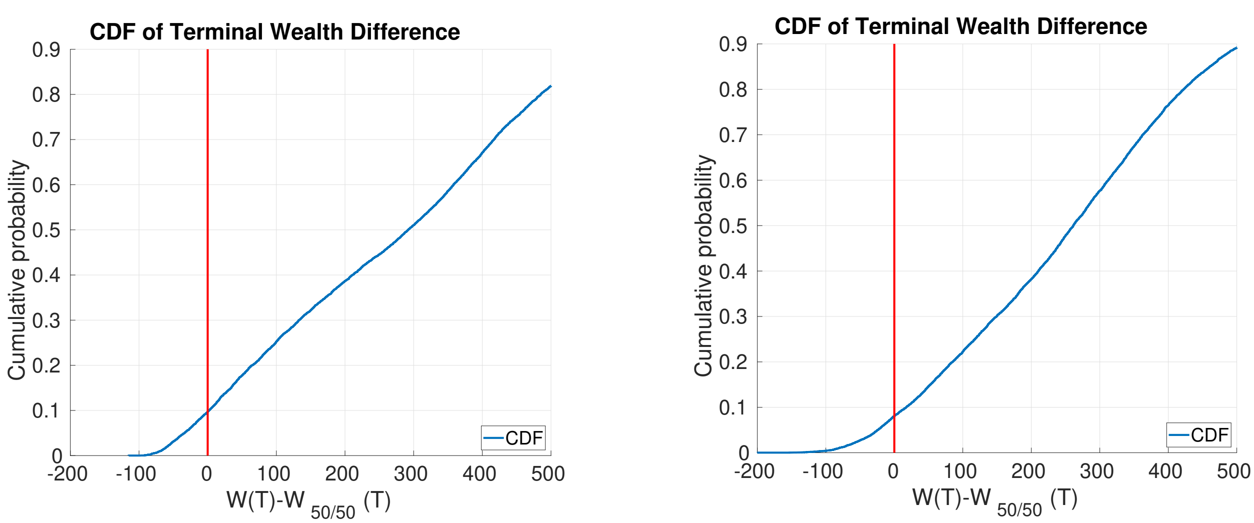
From Figure A.7 we can see that the probability of the adaptive strategy underperforming the constant proportion strategy is less than 10% for both training and testing. When underperformance occurs, the scale of underperformance is small compared to the scale of potential outperformance. Therefore, we conclude that the adaptive strategy controls tail risks consistently in both training and testing, despite the fact that the training dataset is synthetically generated and the testing dataset is bootstrap resampled data.



(a) Training on synthetics data

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

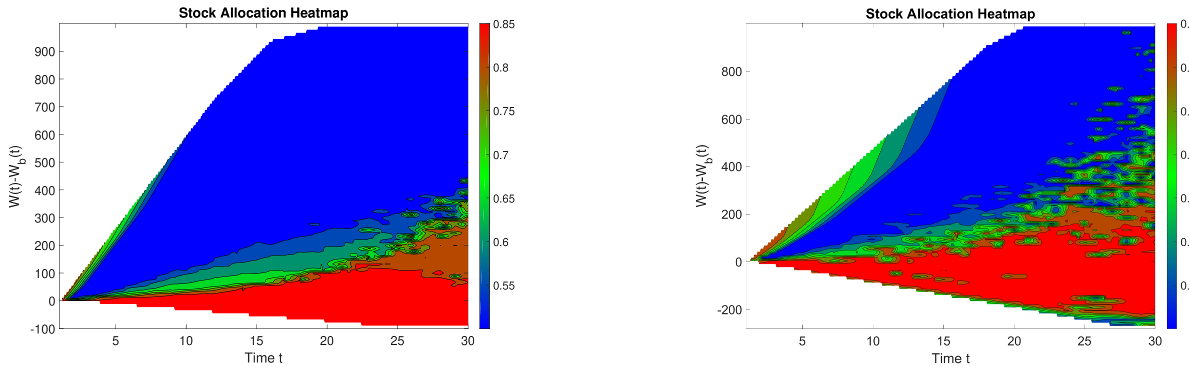
Figure A.6: Histogram of terminal wealth. Model trained on synthetic data and tested on bootstrap resampled data with expected blocksize of 2 years



(a) Training on synthetics data

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

Figure A.7: CDF of terminal wealth difference $W(T) - W_{50/50}(T)$



(a) Stock allocation heatmap - training

(b) Stock allocation heatmap - testing

Figure A.8: Stock allocation heatmap w.r.t. wealth difference

In terms of the allocation strategy, we can see from A.8 that this policy is consistent with the results in Figure 4.3b (bootstrap resampling case) in the sense that the learned strategy is a contrarian strategy that takes more risk when behind, and derisks when ahead. We do want to point out that the heatmap in Figure A.8b is not as smooth as the heatmap in the training case in Figure A.8a.

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.7 Percentile Results with Training/Testing Split

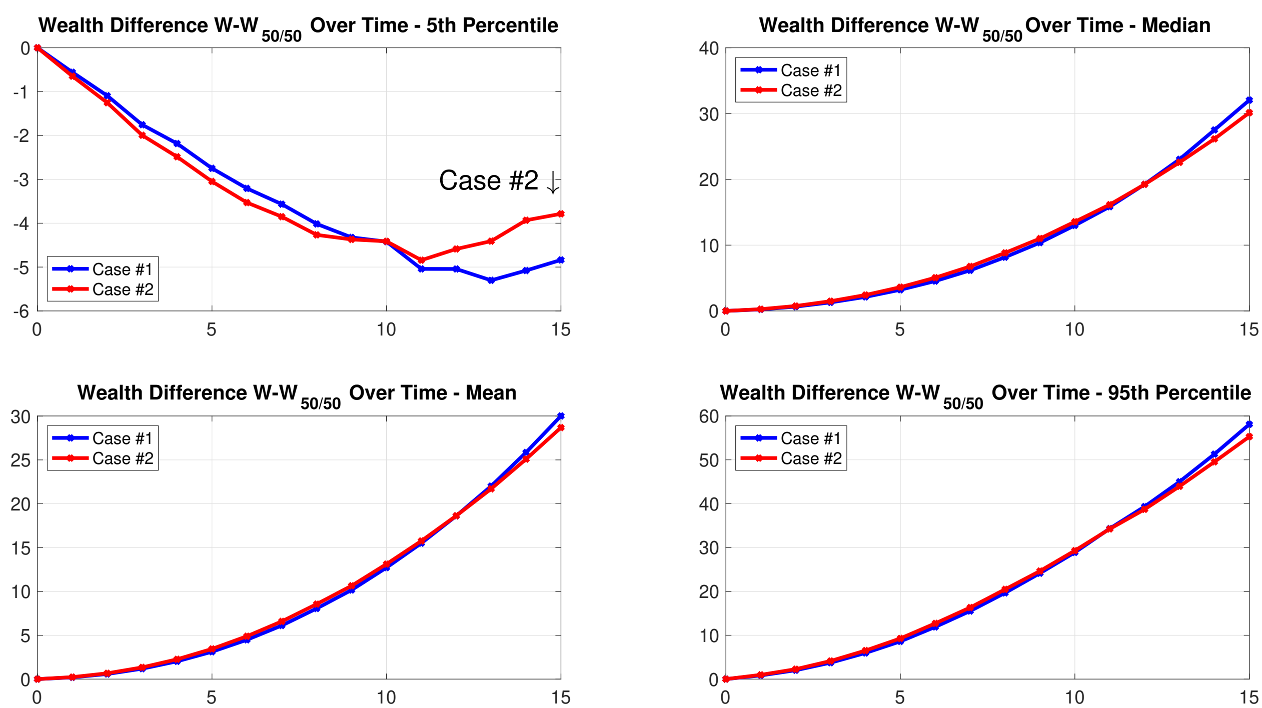


Figure A.9: Percentiles of wealth difference $W(T) - W_{50/50}(T)$ for the two cases

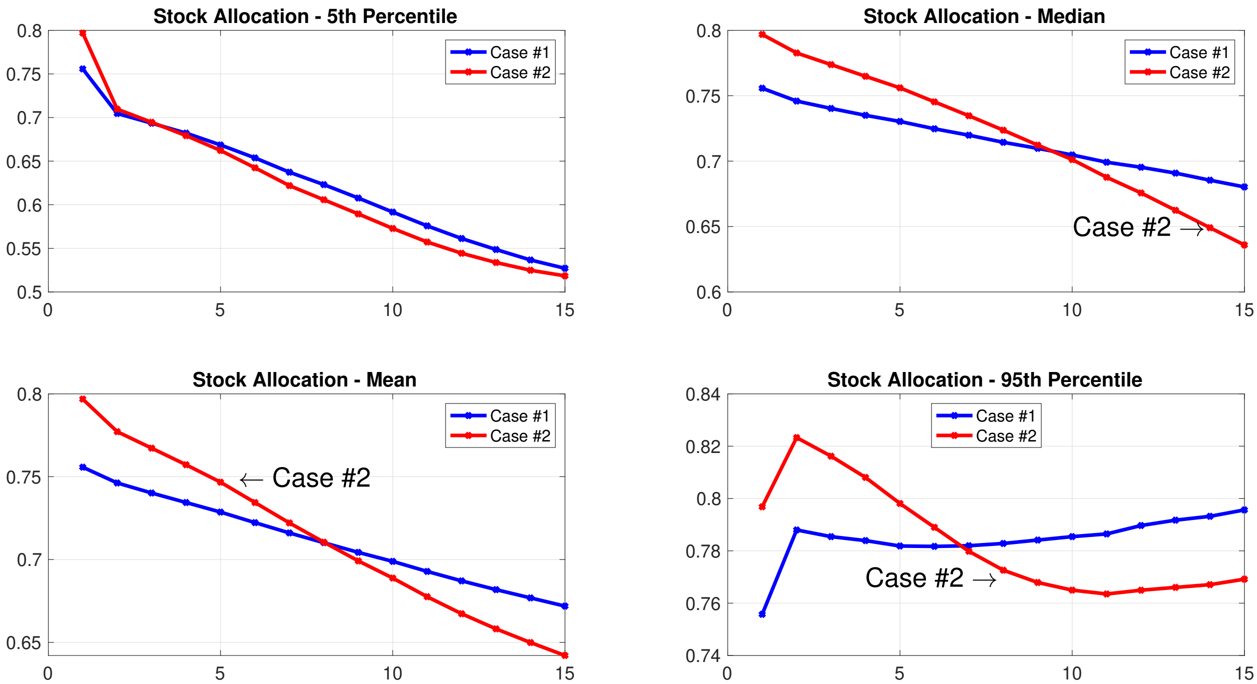


Figure A.10: Stock allocation for the two cases

In Figure A.9, we can see that both cases have almost identical wealth difference in different percentiles, except that Case #2 has slightly better tail risk control (%5 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 A.10, 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 derisk (decrease allocation in the stocks) more aggressively over time than Case #1. We believe the difference comes from the difference in the distributions between the different segments of the underlying historical market returns. However, the difference between allocation strategies is not significant. In fact, the average stock holding over time are quite similar for both cases. In addition, we have already observed similar strategy performances in terms of terminal wealth distributions from figure 5.3a and figure A.9.

A.8 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 expectation 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 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.

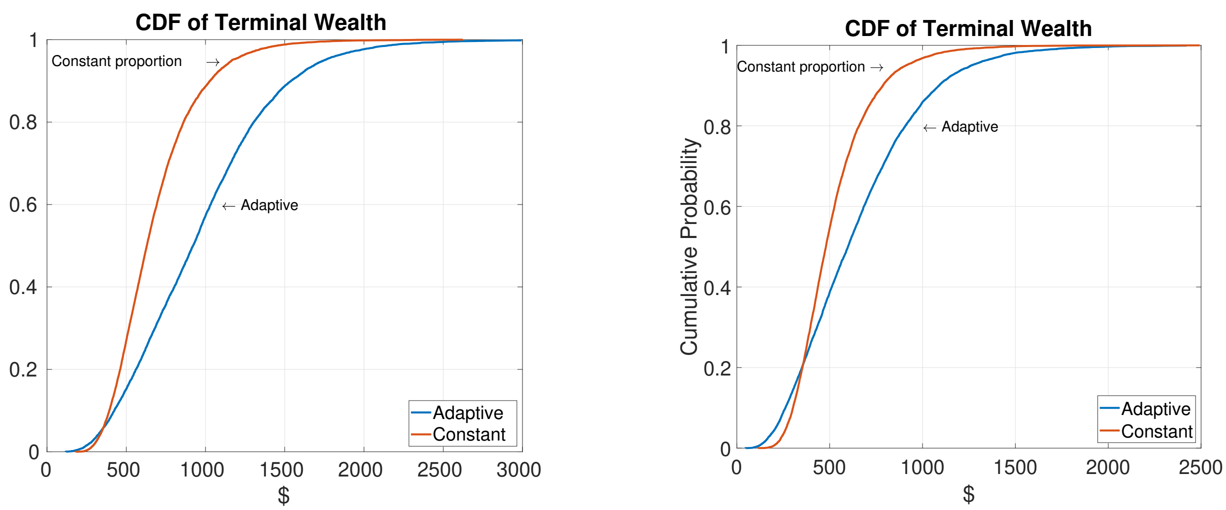
We remark that a lower level of stock returns do not change the main observations in this aritcle. Specifically, in the context of outperforming a stochastic benchmark strategy, lower stock market returns adversely affect the performance of the benchmark strategy as well as the learned adaptive strategy. Consequently, the proposed neural network methodology is still able to learn an adaptive strategy that beats the benchmark strategy.

In the following numerical example, we apply the same experiment setting in Section 4.3, but reduce all historical stock returns by 300 basis points[11]. We train and test the neural network on two separate sets of bootstrap resampled data from historical data with reduced returns.

Testing Results on Bootstrap Data with Original Historical Price					
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 A.7: Terminal wealth statistics of the optimal adaptive strategy. Table shows comparison between testing results on bootstrap data of original historical data and historical data with stock returns adjusted by -300 bps.

In Table A.7, we have included typical statistics on the terminal wealth in the case of reduced stock returns. We have included the original results from Section 4.3 for comparison. As can be seen from Table A.7, while the terminal wealth levels of the adaptive strategy drops, the benchmark constant proportion strategy also drops significantly.



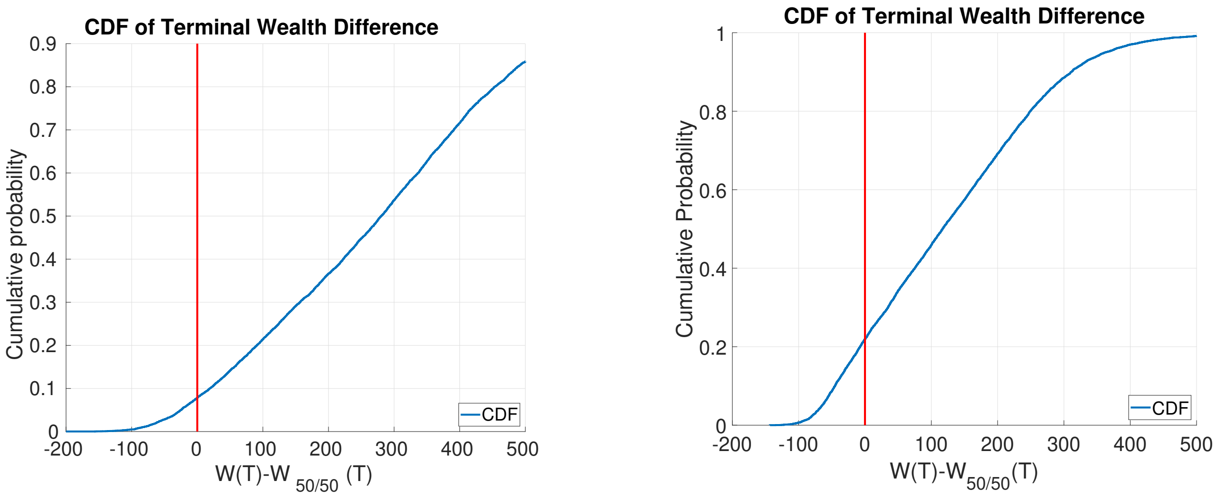
(a) Testing results with original historical data

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

Figure A.11: CDF of Terminal Wealth

From Figure A.11, 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 A.12 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 A.12, the adaptive strategy has only less than 10% of probability of underperforming the benchmark constant proportion strategy with the original historical data, but this probability of underperforming the benchmark strategy increases to around 20% in the case of reduced stock returns. We believe that this is due to the narrower gap between stock returns and bond

[11] We remark that this is not what we the authors expect of future market returns. Nor do such scenarios form any investment suggestions. Our purpose is to use such very conservative market assumptions to address some potential concerns regarding the performance of our proposed methodology under an extreme market scenario. This is essentially a robustness check.



(a) Testing results with original historical data

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

Figure A.12: CDF of Terminal Wealth Difference

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: 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.

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*, 1–23.

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.

Ang, A., D. Papanikolaou, and M. M. Westerfield (2014). Portfolio choice with illiquid assets. *Management Science 60*(11), 2737–2761.

AQR (2021). Capital market assumptions for major asset classes.

Arnott, R. D., K. F. Sherrerd, and L. Wu (2013). The glidepath illusion and potential solutions. *The Journal of Retirement 1*(2), 13–28.

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.

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.

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).

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. ODoherty (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.

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.

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.

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.

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.

Dimson, E., P. Marsh, and M. Staunton (2020). Summary edition credit suisse global investment returns yearbook 2020.

Donaldson, S., F. Kinniry, V. Maciulis, A. J. Patterson, and M. A. DiJoseph (2015). Vanguards approach to target-date funds. *Vanguard Research, Vanguard Group, Valley Forge, PA*.

Duarte, V., J. Fonseca, A. Goodman, and J. A. Parker (2021). Simple allocation rules and optimal portfolio choice over the lifecycle.

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.

Forsyth, P. A. (2021). A stochastic control approach to defined contribution plan decumulation: the nastiest, hardest problem in finance. *North American Actuarial Journal*. to appear.

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.

Ghilarducci, T., M. Papadopoulos, and A. Webb (2017). Inadequate retirement savings for workers nearing retirement. *Schwartz Center for Economic Policy Analysis and Department of Economics, The New School for Social Research, Policy Note Series*.

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.

Gu, S., R. Kelly, and D. Xu (2018). Empirical asset pricing via machine learning. SSRN:3159577.

Harvey, C. R. and Y. Liu (2015). Backtesting. *The Journal of Portfolio Management 42*(1), 13–28.

Hejazi, S. and K. R. Jackson (2016). A neural network approach to efficient valuation of large portfolios of variable annuities. *Insurance: Mathematics and Economics 70*, 169–181.

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.

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, 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.

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.

Malliaris, A. G. and M. E. Malliaris (2008). Investment principles for individual retirement accounts. *Journal of Banking & Finance 32*(3), 393–404.

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

Merton, R. C. (1969). Lifetime portfolio selection under uncertainty: The continuous-time case. *The Review of Economics and Statistics*, 247–257.

Merton, R. C. (1971). Optimum consumption and portfolio rules in a continuous-time model. *Journal of Economic Theory 3:4*, 373–413.

Mulvey, J. M. and H. Vladimirou (1989). Stochastic network optimization models for investment planning. *Annals of Operations Research 20*(1), 187–217.

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*.

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.

Plyakha, Y., R. Uppal, and G. Vilkov (2014). Equal or value weighting? implications for asset-pricing tests. Working paper, EDHEC.

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.

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 1605.02654.

Samuelson, P. A. (1975). Lifetime portfolio selection by dynamic stochastic programming. *Stochastic Optimization Models in Finance*, 517–524.

Tepla, L. (2001). Optimal investment with minimum performance constraints. *Journal of Economic Dynamics and Control 25*(10), 1629–1645.

van Staden, P. M., D.-M. Dang, and P. A. Forsyth (2019). On the distribution of terminal wealth under dynamic mean-variance optimal strategies. *SIAM Journal on Financial Mathematics 11:2*, 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`.

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.

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.