Optimal Decumulation for Retirees using Tontines: a Dynamic Neural Network Based Approach

by

Mohammad Mohibullah Khan Shirazi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Masters of Mathematics
in
Data Science

Waterloo, Ontario, Canada, 2023

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

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

Statement of Contributions

Mohammad Shirazi was the sole author of Chapters 1, 2, 5, 6, and 7 which were written under the supervision of Dr. Peter Forsyth and Dr. Yuying Li and were not written for publication.

This thesis consists in part of one manuscript written for publication. Exceptions to sole authorship of material are as follows:

Research presented in Chapters 3 and 4:

The results provided in these Chapters were developed by Mohammad Shirazi as a co-first author with Marc Chen, Peter Forsyth, and Yuying Li.

Both Forsyth and Li's work was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC), under grant numbers RGPIN-2017-03760 and RGPIN-2020-04331 respectively.

Mohammad Shirazi is also grateful to P. van Staden for supplying the initial software library for NN control problems and data generation.

Abstract

We introduce a new approach for optimizing neural networks (NN) using data to solve a stochastic control problem with stochastic constraints. We utilize customized activation functions for the output layers of the NN, enabling training through standard unconstrained optimization techniques. The resulting optimal solution provides a strategy for allocating and withdrawing assets over multiple periods for an individual with a defined contribution (DC) pension plan. The objective function of the control problem focuses on minimizing left-tail risk by considering expected withdrawals (EW) and expected shortfall (ES). Stochastic bound constraints ensure a minimum yearly withdrawal. By comparing our data-driven approach with the numerical results obtained from a computational framework based on the Hamilton-Jacobi-Bellman (HJB) Partial Differential Equation (PDE), we demonstrate that our method is capable of learning a solution that is close to optimal. We show that the proposed framework is capable of incorporating additional stochastic processes, particularly in cases related to the use of tontines. We illustrate the benefits of using tontines for the decumulation problem and quantify the decrease in risk they bring. We also extend the framework to use more assets and provide test results to show the robustness of the control.

Acknowledgements

I would like to extend my heartfelt gratitude to my professors Peter Forsyth and Yuying Li for their patience, invaluable guidance, and support during the completion of my program. Their expertise and mentorship have been instrumental in shaping the quality and direction of my research. I am truly grateful for their contributions to my academic and intellectual growth.

I would also like to thank my family for their support.

Dedication

This is dedicated to the one I love.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Access to traditional defined benefit (DB) pension plans continues to disappear for employees. In 2022, only 15% of private sector workers in the United States had access to a defined benefit plan, while 66% had access to a defined contribution (DC) plan (U.S. Bureau of Labor Statistics, 2022). In other countries, DB plans have become a thing of the past. This decline has been due to the high cost of DB plans, particularly with improved mortality leading to a greater longevity risk for the employer. A DC pension plan does not magically erase that risk but transfers it to the retiree.

Defined contribution plans leave the burden of creating a withdrawal and allocation strategy to the individual investor, which Nobel Laureate William Sharpe referred to as "the nastiest, hardest problem in finance" (Ritholz, 2017). Indeed, a review of the literature on decumulation strategies (Bernhardt and Donnelly, 2018; MacDonald et al., 2013) shows that balancing all of retirees' concerns with a single strategy is exceedingly difficult. Some strategies have been proposed (such as Bengen's 4% rule (Bengen, 1994)) but these have been shown to be suboptimal. To address these concerns and find an optimal balance between maximizing withdrawals and minimizing the risk of depletion, while guaranteeing minimum withdrawals, the approach in Forsyth (2022) determines a dynamic decumulation and allocation strategy for a standard 30-year investment horizon by formulating it as a constrained optimal stochastic control problem. Numerical solutions are obtained in Forsyth (2022) using dynamic programming, which results in a Hamilton-Jacobi-Bellman (HJB) Partial Differential Equation (PDE).

The HJB PDE framework developed in Forsyth (2022) maximizes expected withdrawals and minimizes the risk of running out of savings, measured by the left tail in the terminal wealth distribution. Maximizing withdrawals and minimizing risk are conflicting goals.

Consequently, in order to determine Pareto optimal points, a typical approach is to use a scalarization technique. A fixed lower bound is imposed on the withdrawal, providing a guaranteed income. An upper bound on withdrawal is also imposed, which can be viewed as the target withdrawal. The investment allocation is also constrained to prohibit shorting and leverage.

This constrained stochastic optimal control problem yields a dynamic stochastic strategy as a solution, which naturally aligns with retirees' concerns and objectives. Note that cash flows are not mortality weighted, consistent with Bengen (1994). This can be justified on the basis of *planning to live, not planning to die* as discussed in Pfau (2018).

This dynamic strategy can be contrasted to traditional strategies such as the *Bengen Rule* (4% Rule), which recommends withdrawing a constant 4% of initial capital each year (adjusted for inflation) and investing equal amounts into stocks and bonds (Bengen, 1994). Initially proposed in 1994, the 4% Rule is found in Scott et al. (2009) to still be a popular strategy 14 years later, and remains as the near-universal recommendation of the top brokerage and retirement planning groups. Recently there has been acknowledgment in the asset management industry that the 4% Rule is sub-optimal, but wealth managers still recommend variations of the same constant withdrawal principle (Williams and Kawashima, 2023). The strategy proposed by Forsyth (2022) is shown to be far more efficient than the Bengen 4% Rule. Unfortunately, the PDE solution in Forsyth (2022) is restricted to low dimensions (i.e. a small number of stochastic factors).

In order to remedy some of the deficiencies of PDE methods (such as in Forsyth (2022)), we propose a neural network (NN) based framework without using dynamic programming. In contrast to the PDE solution approach, our proposed NN approach has the following advantages:

(i) It is data-driven and does not depend on the availability of a parametric model for traded assets. This makes the framework versatile in selecting training data, and less susceptible to model misspecification.

(ii) The control is learned directly by solving the original multi-period optimal control problem and explicitly exploiting the low dimensionality of the control (Van Staden et al., 2023). This technique thus avoids dynamic programming and the associated error propagation. The NN approach can also be applied to higher dimensional problems, such as those with a large number of assets.

(iii) The control generated from NN is a continuous function of time, which fits naturally if the optimal control has the same continuity property. If the optimal control is

discontinuous in time [1], the NN appears capable of producing a smooth, but quite accurate, approximation.[2]

The NN generates an approximate solution to the complicated stochastic optimal control problem. Consequently, it is imperative to assess accuracy and robustness. Rarely is the quality of an NN solution assessed rigorously, since an accurate solution to the optimal control problem is often not readily available. In this thesis, we compare the NN solution to the decumulation problem against the ground-truth solutions from the provably convergent HJB PDE method. We extend our comparison to more complicated formulations targeting promising strategies for the decumulation problem. One such strategy is the use of modern tontines as presented in Forsyth et al. (2022).

A modern tontine allows the pooling of longevity risk while allowing the investor to control asset allocation decisions. There are no guarantees associated with a modern tontine, hence the expected withdrawals will be larger than, for example, an annuity. Essentially, a modern tontine is a peer-to-peer risk-sharing mechanism, with minimal involvement of financial institutions. Tontines can be viewed as another step in the road to financial disintermediation.

We compare the HJB PDE solution provided in Forsyth et al. (2022) to one produced by the NN framework and extend it to incorporate additional stochastic processes. We utilize the framework's efficiency in solving problems of higher dimensional state space by using additional assets and provide an example in the Canadian market. To establish the efficacy of this NN solution, we test it on historical data.

Although unusual, similar comparison assessments exist in different applications, see, e.g., Laurière et al. (2021) for a comparison study on a fishing control problem. As machine learning and artificial intelligence based methods continue to proliferate in finance and investment management, it is crucial to demonstrate that these methods are reliable and explainable in the financial domain (Boukherouaa et al., 2021). We believe that our proposed framework and test results make a step forward in demonstrating deep learning's potential for stochastic control problems in finance.

1.1 Contributions

This thesis incorporates key subjects from Machine Learning and Finance. We summarize the main contributions of this paper as follows:

[1] *Bang-bang* controls, frequently encountered in optimal control, are discontinuous as a function of time.
[2] For a possible explanation of this, see Ismailov (2022).

- Proposing an NN framework with suitable activation functions for decumulation and allocation controls, which yields an approximate solution to the constrained stochastic optimal decumulation problem in Forsyth (2022) and Forsyth et al. (2022) by solving a standard unconstrained optimization problem;

- Demonstrating that the NN solution achieves very high accuracy in terms of the efficient frontier and the decumulation control when compared to the solution from the provably convergent HJB PDE method. This result holds when extended to problems incorporating a tontine overlay;

- Illustrating that, with a suitably small regularization parameter, the NN allocation strategy can differ significantly from the PDE allocation strategy in the region of high wealth and near the end of the withdrawal time horizon, while the relevant risk-reward statistics remain unaffected. This is because the problem is ill-posed, with objective function insensitive to the control in these regions, unless we add a small regularization term;

- Demonstrating the NN framework's capability to generalize to problems with a higher dimension state space, by incorporating state variables related to tontines as well as a higher number of assets.

- Demonstrating the effectiveness of tontines as a strategy to solve the retirement problem. The ability of the NN framework to incorporate several stochastic processes in the state space allows the modeling of complex scenarios related to tontines, enabling us to analyze the strategy against those present in the industry.

- Testing the NN solution's robustness on out-of-sample and out-of-distribution data, as well as its versatility in using different datasets for training.

Our work differs from other NN methods for stochastic optimal problems in finance in that one NN is used for the discontinuous decumulation control while the second NN represents allocation control. While other neural network and deep learning methods for optimal stochastic control problems have been proposed before, they differ significantly from our approach in architecture. These previous approaches take a *stacked* neural network approach as in (Buehler et al., 2019; Han and Weinan, 2016; Tsang and Wong, 2020) or a hybrid dynamic programming and reinforcement learning approach (Huré et al., 2021). In contrast, our framework uses the same two neural networks at all rebalancing times in the investment scenario. Since our NNs take time as an input, the solution will be continuous in time if the control is continuous. Note that the idea of using time as an input

to the NN was also suggested in Laurière et al. (2021). According to the taxonomy of sequential decision problems proposed in Powell (2021), our approach would most closely be described as Policy Function Approximation (PFA).

Furthermore, except Laurière et al. (2021), previous papers do not provide a benchmark for numerical methods, as we do in this work. Our results show that our proposed NN method can approximate the numerical results in Forsyth (2022) with high accuracy. Especially notable, and somewhat unexpected, is that the *bang-bang* control for the withdrawal is reproduced very closely with the NN method.

1.2 Thesis outline

Chapter 2 provides a review of some important concepts discussed later in the thesis and introduces the reader to tontines; Chapter 3 formulates the retirement problem as one in stochastic optimal control and develops the HJB PDE and NN frameworks used to find the optimal control; Chapter 4 provides a comparison study between the solutions proposed by the two frameworks using a simple dynamic allocation and withdrawal strategy with no tontine overlay; Chapter 5 provides a comparison study for strategies incorporating a tontine overlay and demonstrates the NN framework's ability at incorporating nuanced concepts related to tontines; and lastly Chapter 6 analyzes the NN framework's solution when extended to a multiple asset case scenario.

Chapter 2

Background

The retirement problem discussed in the previous chapter has interested practitioners and academics for many years. This has led to the development of several strategies for decumulation and different metrics have been used to assess the effectiveness of the said strategies. In this chapter, we aim to review some important concepts related to the retirement problem and typical decumulation strategies so that we can build upon them in the later chapters.

We begin this chapter by reviewing the different risk measures that are popular within the industry and identifying their strengths and weaknesses. We then move to a discussion of existing strategies and more importantly, their drawbacks. It is these shortcomings that we aim to improve upon in this thesis. We then provide a review of optimal control problems and how these can be applied to the retirement problem. The proposed NN framework in this thesis relies heavily on optimization strategies developed using the optimal control framework. This framework is heavily reliant on data so we discuss the data generation techniques used in this thesis. Finally, we provide an introduction to tontines, discussing important concepts in their design which will be used in the later chapters as we assess the effectiveness of tontines as an alternative to the common strategies.

2.1 Measures of Risk

Evaluating the performance of any strategy requires an assessment of both the risks and rewards. In this section, we discuss some common measures of risk that will be used in this thesis. One of the oldest measures of risk is standard deviation, which measures the

spread around the mean value. In the context of portfolio optimization, this is usually the volatility of the terminal value of the portfolio. A higher standard deviation means that the portfolio can do much worse than the expected terminal value. Of course, this also means that the portfolio can do much better than the mean terminal value. While investors want to avoid losing money, no one complains about making too much money. However, volatility penalizes both upside and downside equally, hence misaligning with the investor's preferences.

Another commonly used measure of risk for target-based investment is Linear Shortfall. Linear Shortfall, as the name implies, measures how short one falls below a target on average. Let W_T represent the terminal wealth and W^* represent the target level of wealth. Then the Linear Shortfall (LS) can be mathematically expressed as:

$$\mathrm{LS}_{W^*} = \mathbb{E}[\min(W_T - W^*,0)] \tag{2.1}$$

Linear shortfall does well to quantify the average effect of not meeting the target. However, it does not say much about the probability of missing the target. Consider two strategies, one which always ends up with a value of -1, and the other which ends with a value of -10 in 10% of the cases. Compared to a target of zero, they both have the same LS. However, one strategy is completely undesirable since it never meets the target. In the context of the retirement problem, it is also difficult to establish a set target wealth that would be sufficient to see out the remaining years.

Another risk measure is the Value at Risk (VAR). Simply put, the VaR at α level, is the α-th percentile. Formally, the VaR, W^*, at the α level is the value of W^* that satisfies the following equation:

$$\int_{-\infty}^{W^*} g(W_T)\, dW_T = \alpha \tag{2.2}$$

where $g(W_T)$ represents the probability density of terminal wealth. While VaR gives a good measure of the location of the tail of the distribution, it fails to describe the distribution of the tail. For example, two distributions can have the same 5th percentile but very different behaviors in the 1st, 2nd, 3rd, and 4th percentiles. VaR, therefore, is indifferent to extreme tail values.

We can improve upon VaR by calculating the Conditional Value at Risk (CVaR). CVaR is the expected value of the outcomes that fall below VaR. Formally, if W^* is the VaR at level α, then the CVaR can be calculated with the following expression:

$$\mathrm{CVaR}_\alpha(W_T) = \frac{1}{\alpha}\int_{-\infty}^{W^*} -W_T\, g(W_T) dW_T \tag{2.3}$$

CVaR, which is typically expressed in terms of loss, summarizes the tail distribution nicely and portrays the tail risk appropriately. Additionally, the CVaR of a portfolio is a coherent measure. A coherent measure must satisfy the properties of monotonicity, sub-additivity, psoitive homogeneity, and translational invariance. Simply put, we want the risk of combining two portfolios to be at most equal to the sum of the risks of individual portfolios. Apart from this, the CVaR of a portfolio is a convex and continuous function of control, under suitable continuous distribution assumptions. Optimizing over such a measure is straight forward and numerically stable. CVaR evaluated at an appropriate level α has a more intuitive meaning in the context of retirement problems than any of the other measures alone. Its interpretation as the average of the worst $\alpha\%$ cases is easier for most people to understand. When defined with respect to wealth, instead of loss (like CVaR), we can term this measure as Expected Shortfall (ES). Therefore ES $= -$CVaR. In the remaining part of the thesis, we primarily use ES as our risk measure. However, for the sake of completeness, we do report other interesting statistics such as VaR and LS.

2.2 Existing strategies

In this section, we review some of the existing strategies that have been presented as a solution to the retirement problem. Each of these strategies has advantages and disadvantages and we provide a brief discussion here.

2.2.1 4% Rule

The 4% rule is perhaps the most widely recognized decumulation strategy. Initially proposed by Bengen in 1994 (Bengen, 1994), this strategy gained popularity due to its simple approach, both in terms of execution and objective. Bengen proposed that retirees should withdraw 4% of their assets annually (adjusted for inflation), and maintain an investment of 50% each in a value-weighted stock index and long-term bonds. Bengen showed that such a strategy would have never run out of cash over any 30-year historical rolling period in the US market. Scott et al. (2009) found this strategy was still popular in 2009 and the recommendation of most retirement planning groups. Even though there has been acknowledgment that this strategy is sub-optimal, many brokerages continue to advise some variation of this rule (Williams and Kawashima, 2023).

The main advantage of this 4% rule is that it is easy to understand and implement. The infrequent rebalancing means that transaction costs are very limited, and do not

require constant engagement with the market. It has been empirically shown to address the primary retiree concern of running out of funds. However, the backtesting done by Bengen is not a fair reflection of future market performances. Committing to such a rigid strategy that does not react to market performance for a long period is unlikely to be an optimal strategy.

2.2.2 Annuities

Annuities are another commonly suggested solution to the retirement problem. Investors who purchase annuities agree to receive a fixed payout for the remainder of their lives. This payout could be adjusted to provide some sort of inflation protection. In fact, true inflation-protected annuities are not available in Canada. The payout adjustment is usually based on a fixed expected inflation rate and may not truly reflect the real inflation rate. Despite their simple structures, annuities continue to be unpopular. Critics complain about the lack of transparency in annuities and no control over investments (MacDonald et al., 2013). Investors end up paying a high fee for purchasing the annuity which leads to lower returns than what could otherwise be expected. Moreover, purchasing an annuity generally means that the retiree forfeits their investment upon death, similar to a DB plan.

2.2.3 ARVA

The Annually Recalculated Virtual Annuity (ARVA) was a strategy first proposed by Waring and Siegel (2015). They propose that the safe withdrawal amount in any year would be the payout that will be received if the retiree purchased a virtual annuity using all their assets. Waring and Siegel (2015) argue that doing so ensures that the retiree never runs out of money, which is what the retiree should absolutely strive to avoid. ARVA allows the retiree to continue allocating an investment mix that the retiree chooses so long as they stick to the safe withdrawal amount. However, payouts from such a strategy can become arbitrarily small.

Throughout this thesis, we compare the strategy proposed by the NN framework to the benchmarks prevalent in the industry. It will be interesting to see how a dynamic allocation and withdrawal strategy incorporating tontines improves upon these existing options.

2.3 Stochastic optimal control problem

Optimal control problems are a class of mathematical problems that involve decision making at different stages of the problem. A model is used for evolving solutions at each stage. Control and state variables are fundamental components in optimal control problems. The control variable has the responsibility of managing and shaping the state, while the state variables maintain records of the conditions at each stage of the problem. These control and state variables play a vital role in such problems by assisting the system in minimizing the overall objective function across all stages. We can formalize an optimal control problem in the following way:

$$\min_{\boldsymbol{u}} J := \phi[\,\boldsymbol{x}(T)] \,+ \int_0^T l(\boldsymbol{x},\boldsymbol{u})dt \tag{2.4}$$

$$\text{subject to } \begin{cases} \frac{d\boldsymbol{x}}{dt} = f(\boldsymbol{x},\boldsymbol{u}) \\ \boldsymbol{x}(0) = \boldsymbol{x}_0 \\ \boldsymbol{u} \in U \end{cases} . \tag{2.5}$$

where ϕ represents the cost function at terminal time T, l represents the cost function during the intermediate steps of the problem, $\boldsymbol{x}$ represents the state of the system, $\boldsymbol{u}$ represents the control, f describes the evolution of state, $\boldsymbol{x}_0$ is the initial state of the system, and U describes the set of possible values (action) that the control can take.

If we incorporate stochasticity into the system, we notice two key changes. First, the differential equation describing the evolution of state $\boldsymbol{x}$ changes to a Stochastic Differential Equation (SDE). This changes the ODE in Equation (2.5) to the following:

$$d\boldsymbol{x} = f(\boldsymbol{x},\boldsymbol{u})dt + \xi \tag{2.6}$$

where ξ represents the randomness in the system. Second, since the state now depends on the distribution of (x), we take an expectation on the payoff function J. This transforms the entire problem in (2.5) to:

$$\min_{\boldsymbol{u}} J := \mathbb{E}\left[\phi[\,\boldsymbol{x}(T)] \,+ \int_0^T l(\boldsymbol{x},\boldsymbol{u})dt\right] \tag{2.7}$$

$$\text{subject to } \begin{cases} d\boldsymbol{x} = f(\boldsymbol{x},\boldsymbol{u})dt + \xi \\ \boldsymbol{x}(0) = \boldsymbol{x}_0 \\ \boldsymbol{u} \in U \end{cases} . \tag{2.8}$$

The expectation can be calculated by using the SDE directly or can be approximated using samples. We defer the discussion of these techniques until later chapters. For now, we turn our attention to the retirement problem.

One may view the retirement problem as an optimal control problem. The portfolio value changes through the control and stochastic processes (market returns). The controls are simply the amount invested in different assets and the amount withdrawn at each stage. The state variable is the state of the portfolio, which could be as simple as the total asset value in the portfolio, or contain complex information such as the history of individual assets in the portfolio. In fact, one can extend the state variables to include variables not directly associated with the portfolio. For example, Ni et al. (2022) includes the wealth in a benchmark portfolio as a state variable. The formulation of long-term financial problems as optimal control problems has been widely studied (see Forsyth and Vetzal (2017); Forsyth (2022)). Most of these studies have used dynamic programming to find the optimal $\boldsymbol{u}$.

The Dynamic Programming (DP) principle states that from any point on an optimal trajectory, the remaining trajectory is optimal for the corresponding problem initiated at that point. This essentially means solving a series of sub-problems, with each requiring the calculation of a conditional expectation (conditional on other state variables). You can imagine a grid with each dimension representing one state variable. It is easy to see that as the number of state variables increases, the grid size grows exponentially, hence exponentially increasing the computational complexity. This is referred to as the curse of dimensionality. Alternatives to DP may focus on solving the entire problem at once instead of solving multiple sub-problems. In later chapters of this thesis, we build a framework that avoids DP, and solves the entire problem at once, relying on simple optimization.

Optimizers use a variety of techniques to approximate the optimal solution to Eq. (2.8). One commonly used method is Gradient Descent which evaluates the gradient of the total cost J w.r.t $\boldsymbol{u}$ and uses this information to move $\boldsymbol{u}$ in the direction that reduces J. Other variants of this methodology have been introduced, most notably stochastic gradient descent (SGD) which evaluates the cost and gradient on a smaller subset, thus reducing the computational power required. Gradient descent algorithms (and SGD) are susceptible to making huge jumps that may miss the optimal point entirely. To avoid this, modifications have been included that alter the learning weight based on moving averages of the gradient and squares of the gradient. One such example is the Adaptive Momentum estimation algorithm proposed by Kingma and Ba (2014) which we use extensively for finding the optimal weights in our NN framework. While it is certainly interesting to discuss the merits and drawbacks of optimizing algorithms, it is beyond the scope of this thesis and we use the optimizer merely as a tool for finding the optimal point. However, one must be cognizant that not all algorithms will perform well on the problem at hand and different

algorithms may be needed for different problems.

2.4 Data Generating Processes

In the previous section, we discussed different approaches used in finding the optimal control. Since the NN framework is heavily reliant on data, we need to use techniques to augment the single historical market path that has been observed. Additionally, generating more data gives us the opportunity to test the robustness of suggested solutions. While the problems in the following chapters use different assets, we remain consistent in our use of historical market returns for these assets. The market data is used in two different ways in subsequent investigations:

(i) *Stochastic model calibration:* Any data set referred to in this thesis as *synthetic data* is generated using Monte Carlo simulations of the parametric stochastic models (SDEs) (we discuss the exact equations later in Section 3.1.4), whose parameters are calibrated to the market data using a threshold technique (Mancini, 2009; Cont and Mancini, 2011; Dang and Forsyth, 2016). We divide the nominal data by the CPI, so the data is inflation adjusted. Details of calibrating the parameters to historical data are provided in later chapters. In order to compute the correlation ρ_{sb}, we first remove any returns which occur at jump times (in either series). See Dang and Forsyth (2016) for details of the technique for detecting jumps.

(ii) *Bootstrap resampling:* Any data set referred to in this thesis as *historical data* is generated by using the stationary block bootstrap method (Politis and Romano, 1994; Politis and White, 2004; Patton et al., 2009; Dichtl et al., 2016) to resample the historical market data series. This method involves repeatedly drawing randomly sampled blocks of random size, with replacement, from the original data series. The block size follows a geometric distribution with a specified expected block size. In the multi-asset case, we simultaneously draw returns from all the individual asset series, in order to preserve correlation effects between asset returns. This, in effect, randomly shuffles the original data and can be repeated to obtain however many resampled paths one desires. Since the order of returns in the sequence is unchanged within the sampled block, this method accounts for some possible serial correlation in market data. Detailed pseudo-code for this method of block bootstrap resampling is given in Forsyth and Vetzal (2019).

We note that block resampling is commonly used by practitioners and academics (see for example Anarkulova et al. (2022); Dichtl et al. (2016); Scott and Cavaglia

(2017); Simonian and Martirosyan (2022); Cogneau and Zakamouline (2013)). Block bootstrap resampling will be used to carry out robustness checks in later chapters. Note that for any realistic number of samples and expected block size, the probability of repeating a resampled path is negligible (Ni et al., 2022).

One important parameter for the block resampling method is the expected block size. The algorithm in Patton et al. (2009) is used to determine the optimal expected block size for the bond and stock returns separately; see Tables E.1 and E.2. Subsequently, we will also test the sensitivity of the results in a range of block sizes in numerical experiments.

To train the neural networks, we require that the number of sampled paths, N, be sufficiently large to fully represent the underlying market dynamics. Subsequently, we first generate training data through MC simulations of the calibrated parametric models described in Section 3.1.4. We emphasize however that, in the proposed data-driven NN framework, we only require return trajectories of the underlying assets. In later chapters, we present results from NNs trained on non-parametrically generated data, e.g. resampled historical data.

We now move towards a brief description of tontines.

2.5 Tontine

A tontine is a financial setup where participants combine their assets into a shared pool. They agree, without the possibility of changing their decision, to receive payments (usually the payout schedule is agreed upon by the members) from this pool during their lifetime. However, upon their death, their accounts are relinquished, and the funds from these forfeitures are divided among the remaining members. The amount of payouts is influenced by the investment performance and the mortality patterns observed within the group of participants.

With this definition, we see that many designs of tontines are possible. For example, it is possible that the asset pool pays nothing at all until only one survivor remains, who is then awarded the entirety of all the assets. Conversely, it is also possible to structure annuity-like payouts with added payments as other members die. Nevertheless, it is imperative to understand that a tontine can be engineered in a manner such that it blends the behavior of both lottery and an investment.

Naturally, it would be important to consider the use of tontine with respect to the retirement problem discussed in Chapter 1. In particular, solutions could be formulated at the following levels:

- Governments and policymakers who wish to encourage participation in lifetime income solutions. This could mean the creation of a market where annuity markets are weak or not very popular.
- DC plan sponsors and employers that wish to offer their customers/employees benefits similar to DB plans and annuity-like options without the need of selecting an insurance guarantor or managing the risk of underfunding.
- Individuals seeking assurances of lifetime income without the high costs associated with annuities.

While tontines provide an interesting perspective on the retirement problem, it must be noted that for each of the cases above, there are important trade-offs being made at each level.

2.5.1 Longevity Risk

There are two basic differences between a tontine and other forms of investment. First, investments into a tontine pool are binding and a member cannot withdraw amounts arbitrarily. This helps prevent a moral hazard issue where any member approaching the end of life would withdraw all their funds from the pool. Secondly, in general, members forfeit all their funds in the pool upon death to be redistributed amongst living members of the tontine pool. This of course means that the beneficiaries of the member's estate do not receive the funds already invested in the tontine pool. The redistributed money provides an increased return to the members of the tontine pool in the form of "mortality credits". This extra return is due to the members pooling the longevity risk.

Risk pooling is an effective method since individual lifespans are more uncertain than those of entire groups. In the case of tontines, we consider two components of longevity risk that are pooled:

- Idiosyncratic Longevity risk: This is the risk associated with each individual's lifespan. In particular, how long a person lives compared to their demographic cohort and how much one can withdraw based on that. Achieving a large diverse pool can substantially diversify this risk.

- Systematic Mortality risk: This is the risk associated with the entire group's lifespan. For example, unanticipated advancements in healthcare could mean that the entire group lives longer than expected. This risk is still borne by the members even when the tontine pool is large. This could be mitigated by payout adjustments. We discuss this in more detail later on in Section 5.4.4

2.5.2 Fair Tontine Principle

Creating a fair tontine is perhaps one of the most challenging aspects of designing a tontine. For example, a tontine where everyone is offered the same payout irrespective of their age will tend to favor the young over older members. We could extend this argument to any groups with substantial differences in mortality rates such as men and women. A *fair tontine* design should be one where no group or member is favored. Such a design is deemed to be "actuarially fair".

We begin by analyzing the concept of fairness in a simple game. For a game to be fair, the expected amount gained by winning is the same as that lost by losing the game. Consider the case where the player has to pay one dollar to enter a game, and guess the correct card drawn from a 52-card deck. The probability of winning is $\frac{1}{52}$. For this to be a fair game, the player must have a reward of 51 times the amount the player stands to lose. In other words, the player must receive \$51 on winning so that $(51 \times \frac{1}{51}) + (-1 \times \frac{51}{52}) = 0$. We generalize the notion to any game such that the player stakes an amount s, the probability of losing is q, and the reward is c, such that the following holds:

$$-qs + (1-q)c = 0 \tag{2.9}$$

We can extend the same principle to a tontine. The members of a pool invest an amount s which they risk forfeiting in the case of death. Let the risk of death be $\mathfrak{q}$. Then for Equation (2.9) to hold, the participant must be compensated with an amount c such that:

$$c = \frac{\mathfrak{q}s}{1-\mathfrak{q}} = \left(\frac{\mathfrak{q}}{1-\mathfrak{q}}\right)s \tag{2.10}$$

Therefore, upon investing an amount s, the member makes a return of $r := \frac{\mathfrak{q}}{1-\mathfrak{q}}$. This return is referred to as the nominal tontine yield, and the amount rs is the nominal tontine gain. In general, this value would be different for each member depending on their risk

of death, with the risk varying over time. Consider member j, at time t, so that the *fair tontine principle* becomes:

$$-\mathfrak{q}_{j,t}s_{j,t} + (1 - \mathfrak{q}_{j,t})c_{j,t} = 0 \tag{2.11}$$

where $s_{j,t}$ and $c_{j,t}$ represents portfolio value s of member j and the mortality credits c received by member j at time t, and $\mathfrak{q}_{j,t}$ represents the probability of death of member j in the time period $[t-1,t]$.

It must be noted that like a zero-sum game, the expected value of participating in a tontine is the same as that of not participating in a tontine. However, the conditional distributions can tend to be very different. The tontine can be very rewarding for members who live a long life, and generally poor for members who pass away early. This is exactly what makes the tontine so effective at ensuring long-term payouts.

The returns highlighted so far in this section are based on the probability of death $\mathfrak{q}$. However, the actual number of deaths may vary every year. To account for this difference, we use the method proposed by Fullmer and Sabin (2018)

2.5.3 Group Gain

Group gain is the common adjustment factor that is applied to all members of a tontine pool to account for the differences between expected and actual number of deaths. The method proposed by Fullmer and Sabin (2018) to formulate this is remarkably simple: Let s_j denote the amount in the tontine pool for member j, and D represent the members who died during the period of interest. Then, the group gain G is defined as:

$$G = \frac{\sum_{j \in D} s_j}{\sum_{j \notin D} r_j s_j} \tag{2.12}$$

where r_j represents the nominal tontine yield due to be received by member j.

As a consequence of the fair tontine principle, the expected value of G will be 1, with random deviations every year. In years with higher-than-expected deaths, G will be greater than 1. Similarly, in years with lower-than-expected deaths, G will be smaller than 1, and the survivors will receive a lower payout than what they expected. Fullmer and Sabin (2018) empirically demonstrate the relationship between large diverse pools and variation in the Group gain. We discuss the effect of this variation later in Chapter 5.

2.5.4 Survivor Benefits

A noteworthy difference between a tontine and other forms of investment is the forfeiture of all assets in a tontine pool upon death. This leads to another problem since retirees may want to leave a bequest to their survivors, particularly their partners. To mitigate this problem, a proposed solution is that payouts continue to be paid to survivors similar to those in DB plans. This reduces the funds available to be redistributed among the surviving members of the pool. Consequently, the retiree would get a lower return during their life. This is surely not desirable and lowers the effectiveness (and attractiveness) of the tontine overlay. As such we will not consider such scenarios in the remainder of this thesis. In practice, some existing tontines (such as QSuper[1] in Australia) provide an extra insurance overlay to provide a survivor benefit. This, of course, comes at the cost of additional fees.

A simple solution to providing partial survivor benefits, such as the spouse receiving 50% of the payouts after the death of the retiree, is to split the initial amount into two tontine pools. In this case, a survivor will clearly receive 50% of the value of the initial investment.

[1] https://qsuper.qld.gov.au/

Chapter 3

Retirement Problem Formulation

Having discussed the preliminaries surrounding the retirement problem and optimal control, we move towards providing a more rigorous formulation of the retirement problem as a stochastic optimal control problem. Later in the chapter, we also develop two key frameworks that are used in computing the optimal control.

3.1 Problem Formulation

3.1.1 Overview

The investment scenario described in Forsyth (2022) concerns an investor with a portfolio wealth of a specified size, upon retirement. The investment horizon is fixed with a finite number of equally spaced rebalancing times (usually annually). At each rebalancing time, the investor first chooses how much to withdraw from the portfolio and then how to allocate the remaining wealth. The investor must withdraw an amount within a specified range. The wealth in this portfolio can be allocated to any mix of the given assets, with no shorting or leverage. The assets the investor can access are a broad stock index fund and a constant maturity bond index fund. Additionally, the investor has the option of participating in a tontine pool consisting of individual tontine accounts (Fullmer and Sabin, 2018).

In the time that elapses between re-balancing times, the portfolio's wealth will change according to the dynamics of the underlying assets and any gains from tontine. If the wealth of the portfolio goes below zero (due to minimum withdrawals), no stock holdings are permitted. In the event of insolvency, all stock is sold, debt will grow at the borrowing

rate, and withdrawals are restricted to the minimum amount. At the end of the time horizon, a final withdrawal may be made and the portfolio is liquidated, yielding the terminal wealth.

We assume here that the investor has other assets, such as real estate, which are non-fungible with investment assets[1]. These other assets can be regarded as a hedge of last resort, which can be used to fund any accumulated debt (Pfeiffer et al., 2013). This is not a novel assumption and is in line with the mental bucketing idea proposed by Shefrin and Thaler (1988). The use of this assumption within literature targeting similar problems is also common (see Forsyth et al. (2022)). Of course, the objective of optimal control is to make running out of savings an unlikely event.

The investor's goal then is to maximize the weighted sum of expected total withdrawals and the mean of the worst 5% of the outcomes (in terms of terminal wealth). Recall that this tail risk measure was introduced as Expected Shortfall (ES) at the 5% level in Section 2.1. In this section, this optimization problem will be described with the mathematical details common to both the HJB and NN methods.

3.1.2 Mortality credits (Tontine gain):

In this thesis, we consider a perpetual tontine pool of individual tontine accounts. New membership to the pool is not restricted and the pool size is large (more than, say, one million members). We expect such a pool to be managed by a large financial organization, in return for a fee. Here we build upon the fair tontine principle (Section 2.5) and describe the nominal tontine gain for participation in a tontine pool. Consider an individual j at time t_{i-1}, with investment v_{i-1}^j. If the investor were to pass away in the time period (t_{i-1}, t_i), they would forfeit their investment amount v_i^j (we assume mortality credits are distributed at the end of each year). Likewise, if the investor were to survive, they stand to gain c_i^j (forfeited by other dying members), which we refer to as tontine gain (or mortality credits). For an actuarially fair tontine, we would require that there is no expected gain from participating in the tontine. Assuming that the probability of death for investor j in (t_{i-1}, t_i) is $\mathfrak{q}_{i-1}^j$, we have:

$$-\mathfrak{q}_{i-1}^j v_i^j + (1 - \mathfrak{q}_{i-1}^j) c_i^j = 0 \tag{3.1}$$

$$c_i^j = \frac{\mathfrak{q}_{i-1}^j}{(1 - \mathfrak{q}_{i-1}^j)} v_i^j \tag{3.2}$$

[1]In practice, residential real estate can be monetized using a reverse mortgage

Since v_i^j is the investment by member j into the pool, we define the tontine gain rate $(\mathbb{T}_i^g)^j$ as:

$$(\mathbb{T}_i^g)^j = \frac{\mathfrak{q}_{i-1}^j}{1 - \mathfrak{q}_{i-1}^j} \tag{3.3}$$

In our retirement problem, we consider a single retiree and hence can drop the superscript j. Therefore:

$$\mathbb{T}_i^g = \frac{\mathfrak{q}_{i-1}}{1 - \mathfrak{q}_{i-1}} \tag{3.4}$$

The probability of death $\mathfrak{q}_i$ is based on CPM tables provided by the Canadian Institute of Actuaries. The probability of death for a 65-year-old male in the next year is 0.00844, which means that the tontine gain earned by surviving the year would be a meager 0.00851.

3.1.3 Group Gain

While the probability of death can be taken from CPM tables, the number of deaths, in reality, is random and would differ from the expected value. Since the total amount forfeited due to deaths in such a scenario would depend on the actual deaths, we need to make an adjustment. Consider a tontine pool of m investors at t_{i-1}, and define the indicator function $\mathbf{1}_i^j$ as:

$$\mathbf{1}_i^j \quad = \quad \begin{cases} 1 \ ; \ \text{Investor } j \text{ is alive at } t_{i-1} \text{ and at } t_i \\ 0 \ ; \ \text{Investor } j \text{ is alive at } t_{i-1} \text{ and dead at } t_i \end{cases} \tag{3.5}$$

The total actual gains and forfeitures from this tontine pool in the period (t_{i-1}, t_i) are thus:

$$\begin{aligned}
\text{Total Actual Gains} &= \sum_{j=1}^{m} \mathbf{1}_i^j c_i^j \\
&= \sum_{j=1}^{m} \mathbf{1}_i^j \left(\frac{\mathfrak{q}_{i-1}^j}{1-\mathfrak{q}_{i-1}^j} \right) v_i^j \qquad (3.6) \\
\text{Total Actual Forfeitures} &= \sum_{j=1}^{m} (1-\mathbf{1}_i^j) v_i^j \qquad (3.7)
\end{aligned}$$

While the expected value of both total actual gains and forfeitures is the same, the actual deaths may differ from the expected number of deaths, we will use the method suggested by Fullmer and Sabin (2018), by introducing the group gain G_i at t_i as:

$$G_i = \frac{\sum_{j=1}^{m} (1-\mathbf{1}_i^j) v_i^j}{\sum_{j=1}^{m} \mathbf{1}_i^j c_i^j} \qquad (3.8)$$

where $\mathbf{1}_i^j$ denotes the actual state of the member. Thus modifying the actual mortality credits earned by the investor, $\hat{c}_i^j$ to be:

$$\hat{c}_i^j = G_i \left(\frac{\mathfrak{q}_{i-1}^j}{1-\mathfrak{q}_{i-1}^j} \right) v_i^j \qquad (3.9)$$

This ensures that the total mortality credits paid out is equal to the total amount forfeited by members who have died, transforming Equation (3.4) into a stochastic process where:

$$\mathbb{T}_i^g = G_i \left(\frac{\mathfrak{q}_{i-1}^j}{1-\mathfrak{q}_{i-1}^j} \right) \qquad (3.10)$$

and G_i is a random number.

Incorporating the group gain ensures that any unexpected gains are shared in the same proportion as the tontine gain rate. This effectively scales the tontine gain rate.

Fullmer and Sabin (2018) show, based on simulations, that for a tontine pool of 10000 members, the mean of group gain is one, and the standard deviation is 0.1, with the standard deviation decreasing as more members are added. Forsyth et al. (2022) shows that a normally distributed group gain with mean one and standard deviation 0.1 results in no significant degradation in performance compared to assuming a constant group gain of one. We assume the group gain G follows a uniform distribution, thus making the most conservative estimates on uncertainty for a given mean and standard deviation. We further investigate the effect of random group gain later on in Section 5.5.

3.1.4 Stochastic Process Model

Let S_t and B_t represent the real (i.e. inflation-adjusted) *amounts* invested in the stock index and a constant maturity bond index, respectively. These assets are modeled with correlated jump diffusion models, in line with MacMinn et al. (2014). These parametric stochastic differential equations (SDEs) allow us to model non-normal asset returns. The SDEs are used in solving the HJB PDE and generating training data with Monte Carlo (MC) simulations in the proposed NN framework. For the remainder of this paper, we refer to simulated data using these models as *synthetic* data.

If a jump is triggered, $S_t = \xi^s S_{t^-}$, where ξ^s is a jump multiplier and $S_{t^-} = S(t-\epsilon), \epsilon \to 0^+$ (S_{t^-} is the time immediately before t). $\log(\xi^s)$ is assumed to follow a double exponential distribution (Kou, 2002; Kou and Wang, 2004). The jump is either upward or downward, with probabilities u^s and $1-u^s$ respectively. Let $y = \log(\xi^s)$, then y has density

$$f^s(y) = u^s \eta_1^s e^{-\eta_1^s y} \mathbf{1}_{y \geq 0} + (1-u^s)\eta_2^s e^{\eta_2^s y} \mathbf{1}_{y<0} \ . \tag{3.11}$$

We also define

$$\gamma_\xi^s \quad = \quad E[\xi^s - 1] = \frac{u^s \eta_1^s}{\eta_1^s - 1} + \frac{(1-u^s)\eta_2^s}{\eta_2^s + 1} - 1 \ . \tag{3.12}$$

The starting point for building the jump diffusion model is a standard geometric Brownian motion, with drift rate μ^s and volatility σ^s. A third term is added to represent the effect of jumps, and a compensator is added to the drift term to preserve the expected drift rate. For stocks, this gives the following stochastic differential equation (SDE) that describes how the amount in the stock account S_t (inflation-adjusted) evolves between rebalancing times:

$$\frac{dS_t}{S_{t^-}} = \left(\mu^s - \lambda_\xi^s \gamma_\xi^s\right) dt + \sigma^s \, dZ^s + d\left(\sum_{i=1}^{\pi_t^s} (\xi_i^s - 1)\right), \quad t \in (t_i, t_{i+1}) \tag{3.13}$$

where dZ^s is the increment of a Wiener process, π_t^s is a Poisson process with positive intensity parameter λ_ξ^s. For all i, ξ_i^s are assumed i.i.d, positive, and with distribution (3.11). In addition, it is assumed that ξ_i^s, π_t^s, and Z^s are all mutually independent.

In the practitioner literature, it is usual to model the returns of a constant maturity (real, i.e. inflation-adjusted) bond index fund, by an SDE. Following the lead of MacMinn et al. (2014); Lin et al. (2015), we model the constant maturity (real) bond index by a jump diffusion process. Let the amount in the constant maturity bond index be $B_{t^-} = B(t-\epsilon), \epsilon \to 0^+$. Between rebalancing times, the amount in the bond account B_t evolves as

$$\frac{dB_t}{B_{t^-}} = \left(\mu^b - \lambda_\xi^b \gamma_\xi^b + \mu_c^b \mathbf{1}_{\{B_{t^-}<0\}}\right) dt + \sigma^b \, dZ^b + d\left(\sum_{i=1}^{\pi_t^b} (\xi_i^b - 1)\right), \quad t \in (t_i, t_{i+1}) \tag{3.14}$$

where the corresponding terms in Equation (3.14) are defined in a similar fashion to Equation (3.13). π_t^b denotes a Poisson process, having a non-negative intensity parameter λ_ξ^b, $\gamma_\xi^b = E[\xi^b - 1]$, and $y = \log(\xi^b)$ has the same distribution as in Equation 3.11 (denoted by $f^b(y)$) with distinct parameters, u^b, η_1^b, and η_2^b. We make the assumption that ξ_i^b, π_t^b, and Z^b mutually independent, similar to assumptions placed on the SDE for S_t. The term $\mu_c^b \mathbf{1}_{\{B_{t^-}<0\}}$ represents the borrowing spread (assumed non-negative).

The correlation between the two assets' diffusion processes is ρ_{sb}, i.e., $dZ^s \cdot dZ^b = \rho_{sb} \, dt$. The jump processes are assumed to be independent. For further details concerning the justification of this market model, refer to Forsyth (2022).

The total amount in the retirement account at time t, W_t is given by

$$\text{Total wealth} \equiv W_t = S_t + B_t. \tag{3.15}$$

With the exception of an insolvency state, shorting stock and using leverage (i.e., borrowing) are not permitted, a realistic constraint in the context of DC retirement plans. Furthermore, if the wealth ever goes below zero, due to the guaranteed withdrawals, all stock holdings are sold. Debt then grows at the bond rate plus a borrowing spread. We emphasize that we are assuming that the retiree has other assets (i.e., residential real estate) which can be used to fund any accumulated debt. In practice, this could be done using a reverse mortgage (Pfeiffer et al., 2013).

3.1.5 Notational Conventions

Let $\mathcal{T}$ denote the set of discrete times at which rebalancing and withdrawals are permitted

$$\mathcal{T} = \{t_0 = 0 < t_1 < t_2 < \ldots < t_M = T\} \ . \tag{3.16}$$

The beginning of the investment period is $t_0 = 0$. We assume each rebalancing time is evenly spaced, meaning $t_i - t_{i-1} = \Delta t = T/M$ is constant. For notational simplicity, it will be convenient to denote time dependence in two forms, i.e. $S_t \equiv S(t), B_t \equiv B(t)$ and $W_t \equiv W(t)$. At each rebalancing time, $t_i \in \mathcal{T}$, we consider the following order of events. First, the investor receives mortality credits from participation in the tontine pool and withdraws an amount of cash q_i from the portfolio. Subsequently, the portfolio is then rebalanced. At time T, there is one final withdrawal, q_T, and then the portfolio is liquidated. We assume no taxes or transaction costs are incurred on rebalancing. The no-tax assumption is reasonable since retirement accounts are typically tax-advantaged. In addition, since trading is infrequent, we assume transaction costs to be negligible (Dang and Forsyth, 2014). For any function $f(t)$, we denote

$$f(t_i^+) \equiv \lim_{\epsilon \to 0^+} f(t_i + \epsilon) \ , \qquad f(t_i^-) \equiv \lim_{\epsilon \to 0^+} f(t_i - \epsilon) \ \ . \tag{3.17}$$

Let $X(t) = (S(t), B(t))$, $t \in [0,T]$ denote the multidimensional controlled underlying process. Following typical notation, let $x = (s, b)$ denote the realized state of the system.

At each rebalancing time t_i, the investor receives mortality credits, and withdraws the amount $q_i(\cdot)$, determined by the control at time t_i; that is, $q_i(\cdot) = q_i(X(t_i^-)) = q(X(t_i^-), t_i)$. This control is used to evolve the investment portfolio from W_t^- to W_t^+

$$W(t_i^+) = W(t_i^-)(1 + \mathbb{T}_i^g)\exp(-(\Delta t)_i \mathbb{T}^f) - q_i \ , \qquad t_i \in \mathcal{T} \ . \tag{3.18}$$

With some notational abuse, we define $W(t_i^-)$ as the wealth after tontine gains $\mathbb{T}_i^g$ and fee $\mathbb{T}^f$, and at the instance before withdrawal q_i:

$$W(t_i^-) = \Big(S(t_i^-) + B(t_i^-)\Big)(1 + \mathbb{T}_i^g)\exp(-(\Delta t)_i \mathbb{T}^f), \qquad t_i \in \mathcal{T} \ . \tag{3.19}$$

The withdrawal and allocation controls are formally functions of the state before withdrawal, $X(t_i^-)$. However, it is useful to note that the allocation control is specifically a function of the state after withdrawal. This is simply due to the fact that rebalancing

occurs after the withdrawal. Let $p_i(\cdot)$ represent the fraction of wealth in stocks, after rebalancing

$$\begin{aligned} S(t_i^+) &= p(X(t_i^+), t_i) W(t_i^+) \\ B(t_i^+) &= (1 - p(X(t_i^+), t_i)) W(t_i^+) \ . \end{aligned} \tag{3.20}$$

As formulated, assuming no transaction costs, it is shown in Forsyth (2022) that the control depends on wealth only, i.e., $p_i(\cdot) = p(X(t_i^+), t_i) = p_i(W_i^+)$) (assuming processes (3.13) and (3.14)). Therefore, we make another notational adjustment for the sake of simplicity and consider $q_i(\cdot)$ to be a function of wealth before withdrawal, W_i^-, and $p_i(\cdot)$ to be a function of wealth after withdrawal, W_i^+.

We assume instantaneous rebalancing, with the implication that the control at time t_i is described by a pair $(q_i(\cdot), p_i(\cdot)) \in \mathcal{Z}(W_i^-, W_i^+, t_i)$, where $\mathcal{Z}(W_i^-, W_i^+, t_i)$ represents the set of admissible control values for t_i. The constraints on the allocation control are no shorting and no leverage (except in an insolvent state). There are minimum and maximum values for the withdrawal. In the normal course of events, the no-shorting and no-leverage constraints imply that wealth is always positive. However, due to minimum withdrawals at rebalancing times, it is possible for insolvency to occur. In this case, no stock holdings are permitted, and debt accumulates at the borrowing rate. Any subsequent withdrawals are restricted to the minimum amounts. Any non-zero stock positions are liquidated at terminal time. We can mathematically state these constraints by imposing suitable bounds on the value of the controls as follows:

$$\mathcal{Z}_q(W_i^-, t_i) = \begin{cases} [q_{\min}, q_{\max}], & \text{if } t_i \in \mathcal{T}, \quad W_i^- > q_{\max} \\ [q_{\min}, W_i^-], & \text{if } t_i \in \mathcal{T}, \quad q_{\min} < W_i^- < q_{\max} \\ \{q_{\min}\}, & \text{if } t_i \in \mathcal{T}, \quad W_i^- < q_{\min} \end{cases} , \tag{3.21}$$

$$\mathcal{Z}_p(W_i^+, t_i) = \begin{cases} [0,1], & \text{if } W_i^+ > 0, \quad t_i \in \mathcal{T}, \quad t_i \neq t_M \\ \{0\}, & \text{if } W_i^+ \leq 0, \quad t_i \in \mathcal{T}, \quad t_i \neq t_M \\ \{0\}, & \text{if } t_i = t_M \end{cases} , \tag{3.22}$$

$$\mathcal{Z}(W_i^-, W_i^+, t_i) = \mathcal{Z}_q(W_i^-, t_i) \times \mathcal{Z}_p(W_i^+, t_i) \ . \tag{3.23}$$

At each t_i, we seek the optimal control for all possible combinations of $(S(t), B(t))$ having the same total wealth Forsyth (2022). Hence, the controls for both withdrawal and allocation are formally a function of wealth and time before withdrawal (W_i^-, t_i), but for

implementation purposes it will be helpful to write the allocation as a function of wealth and time after withdrawal (W_i^+,t_i). The admissible control set $\mathcal{A}$ can be written as

$$\mathcal{A} = \left\{ (q_i, p_i)_{0 \le i \le M} : (q_i, p_i) \in \mathcal{Z}(W_i^-, W_i^+, t_i) \right\} . \tag{3.24}$$

An admissible control $\mathcal{P} \in \mathcal{A}$, can be written as

$$\mathcal{P} = \{(q_i(\cdot), p_i(\cdot)) : \ (q_i(\cdot), p_i(\cdot)) \in \mathcal{Z}(W_i^-, W_i^+, t_i), \ i = 0, \ldots, M\} . \tag{3.25}$$

It will sometimes be necessary to refer to the tail of the control sequence at $[t_n, t_{n+1}, \ldots, t_M]$, which we define as

$$\mathcal{P}_n = \{(q_n(\cdot), p_n(\cdot)) \ldots, (p_M(\cdot), q_M(\cdot))\} . \tag{3.26}$$

The essence of the problem, for both the HJB and NN methods outlined in this thesis, will be to find an optimal control $\mathcal{P}^*$.

3.1.6 Risk: Expected Shortfall

Let $\mathcal{G}(W_T)$ be the probability density of terminal wealth W_T at $t = T$. For $0 < \alpha < 1$, typically $\alpha = 5\%$, let W'_α satisfy

$$\int_{-\infty}^{W'_\alpha} \mathcal{G}(W_T) \, dW_T = \alpha , \tag{3.27}$$

i.e., $Pr[W_T < W'_\alpha] = \alpha$. W'_α can be interpreted as the Value at risk (VAR) at the level α. We then define the Expected Shortfall (ES) as the mean of the worst α fraction of the terminal wealth. Mathematically,

$$\mathrm{ES}_\alpha = \frac{\int_{-\infty}^{W'_\alpha} W_T \ \mathcal{G}(W_T) \ dW_T}{\alpha} . \tag{3.28}$$

As formulated, a higher ES is more desirable than a smaller ES (Equation (3.28) is formulated in terms of final wealth not losses). For computational purposes, it is useful to use the definition of ES as devised in Rockafellar and Uryasev (2000),

$$\mathrm{ES}_\alpha \ = \ \sup_{W'} E\left[W' + \frac{1}{\alpha} \min(W_T - W', 0) \right] . \tag{3.29}$$

Under a control $\mathcal{P}$, and initial state X_0, this becomes:

$$\mathrm{ES}_\alpha(X_0^-, t_0^-) \quad = \quad \sup_{W'} E_{\mathcal{P}}^{X_0^-, t_0^-}\left[W' + \frac{1}{\alpha}\min(W_T - W', 0)\right] . \tag{3.30}$$

The candidate values of W' can be taken from the set of possible values of W_T. It is important to note here that we define $\mathrm{ES}_\alpha(X_0^-, t_0^-)$ which is the value of ES_α as seen at t_0^-. Hence, W' is fixed throughout the investment horizon. In fact, we are considering the induced time-consistent strategy, as opposed to the time-inconsistent version of an expected shortfall policy (Strub et al., 2019; Forsyth, 2020). This issue is addressed in more detail in Appendix A.

3.1.7 Reward Measure: Total Expected Withdrawals (EW)

As a measure of reward, we will use total expected withdrawals. Mathematically, total expected withdrawals (EW) is defined as

$$\mathrm{EW}(X_0^-, t_0^-) = E_{\mathcal{P}}^{X_0^-, t_0^-}\left[\sum_{i=0}^{M} q_i\right] . \tag{3.31}$$

Remark 3.1 (No discounting, no mortality weighting)**.** *Note that we do not discount the future cash flows in Equation* (3.31)*. We remind the reader that all quantities are assumed real (i.e. inflation-adjusted) so that we are effectively assuming a real discount rate of zero, which is a conservative assumption. This is also consistent with the approach used in the classical work of Bengen (1994). In addition, we do not mortality weight the cash flows, which is also consistent with Bengen (1994). See Pfau (2018) for a discussion of this approach (i.e.* plan to live, not plan to die*).*

3.1.8 Defining a Common Objective Function

In this section, we describe the common objective function used by both the HJB method and the NN method.

Since increasing Expected Withdrawals (EW) typically causes a simultaneous decrease in Expected Shortfall (ES), we determine Pareto optimal points for this multi-objective problem. For a given scalarization parameter κ, we seek the optimal control $\mathcal{P}_0$ such that the following is maximized,

$$\mathrm{EW}(X_0^-, t_0^-) + \kappa \mathrm{ES}_\alpha(X_0^-, t_0^-) . \tag{3.32}$$

We define (3.32) as the pre-commitment EW-ES problem ($PCEE_{t_0}(\kappa)$) and write the problem formally as

$$
\begin{aligned}
&(PCEE_{t_0}(\kappa)): \\
&J\left(s,b,t_0^-\right) = \sup_{\mathcal{P}_0\in\mathcal{A}}\sup_{W'}\left\{ E_{\mathcal{P}_0}^{X_0^-,t_0^-}\left[\sum_{i=0}^{M} q_i \;+\; \kappa\left(W' + \frac{1}{\alpha}\min(W_T - W', 0)\right) \overbrace{+\epsilon W_T}^{stabilization} \;\middle|\; X(t_0^-) = (s,b) \right]\right\} \\
&\text{subject to } \begin{cases} (S_t, B_t) \text{ follow processes (3.13) and (3.14)}; \quad t \notin \mathcal{T} \\ W_i^+ = W_i^- - q_i\,, \;\; X_i^+ = (S_i^+, B_i^+) \\ W_i^- = (S_i^- + B_i^-)(1 + \mathbb{T}_i^g)\exp(-(\Delta t)_i \mathbb{T}^f) \\ S_i^+ = p_i(\cdot)W_i^+\,, \;\; B_i^+ = (1 - p_i(\cdot))W_i^+ \\ (q_i(\cdot), p_i(\cdot)) \in \mathcal{Z}(W_i^-, W_i^+, t_i) \\ i = 0, \ldots, M, \quad t_i \in \mathcal{T} \end{cases}
\end{aligned} \tag{3.33}
$$

The ϵW_T stabilization term serves to avoid ill-posedness in the problem when $W_t \gg W'$, $t \to T$, and has little effect on optimal (ES, EW) or other summary statistics when $|\epsilon| \ll 1$. Further details about this stabilization term and its effects on both the HJB and NN framework will be discussed in Section 4.2. The objective function in (3.33) serves as the basis for the value function in the HJB framework and the loss function for the NN method.

Remark 3.2 (Induced time consistent policy)**.** *Note that a strategy based on* ($PCEE_{t_0}(\kappa)$) *is formally a pre-commitment strategy (i.e., time inconsistent). However, we will assume that the retiree actually follows the induced time consistent strategy (Strub et al., 2019; Forsyth, 2020; 2022), which is identical to the pre-commitment control at the initial time. See Appendix A for more discussion of this subtle point. Subsequently, we will refer to the strategy from (3.33) as the EW-ES optimal control, noting that it is equivalent to an induced time consistent control at any time* $t_i > t_0$.

3.2 HJB Dynamic Programming Optimization Framework

The HJB framework uses dynamic programming, creating sub-problems from each time step in the problem and moving backward in time. For the convenience of the reader, we will summarize the algorithm in Forsyth (2022) here.

3.2.1 Deriving Auxiliary Function from $PCEE_{t_0}(\kappa)$

The HJB framework begins with defining auxiliary functions based on the objective function (3.33) and the underlying stochastic processes. An equivalent problem is then formulated, which will then be solved to find the optimal value function.

We begin by interchanging the $\sup_{\mathcal{P}_0}$ and $\sup_{W'}$ operators. This will serve as the starting point for the HJB solution

$$J\left(s,b,t_0^-\right) = \sup_{W'} \sup_{\mathcal{P}_0 \in \mathcal{A}} \left\{ E_{\mathcal{P}_0}^{X_0^-,t_0^-} \left[\sum_{i=0}^{M} q_i + \kappa\left(W' + \frac{1}{\alpha}\min(W_T - W', 0)\right) + \epsilon W_T \middle| X(t_0^-) = (s,b) \right] \right\} . \tag{3.34}$$

The auxiliary function which needs to be computed in the dynamic programming framework at each time t_n will have an associated strategy for any $t_n > 0$ that is equivalent with the solution of $PCEE_{t_0}(\kappa)$ for a fixed W'. For a full discussion of pre-commitment and time-consistent ES strategies, we refer the reader to Forsyth (2020), which also includes a proof with similar steps of how the following auxiliary function is derived from (3.34). Including W' in the state space gives us the expanded state space $\hat{X} = (s,b,W')$. Define the problem domain $\Omega = [0,\infty) \times (-\infty,+\infty) \times (-\infty,+\infty) \times [0,\infty)$. The auxiliary function $V(s,b,W',t) \in \Omega$ is then defined as,

$$
\begin{aligned}
V(s,b,W',t_n^-) \;&=\; \sup_{\mathcal{P}_n\in\mathcal{A}_n}\left\{ E^{\hat{X}_n^-,t_n^-}_{\mathcal{P}_n}\left[\sum_{i=n}^{M} q_i + \kappa\left(W' + \frac{1}{\alpha}\min((W_T - W'),0)\right)\right.\right. \\
&\qquad \left.\left. +\epsilon W_T \,\middle|\, \hat{X}(t_n^-) = (s,b,W')\right]\right\} \\
\text{subject to} \quad &\begin{cases}
(S_t,B_t) \text{ follow processes (3.13) and (3.14)}; \quad t\notin\mathcal{T} \\
W_i^+ = W_i^- - q_i\,,\;\; X_i^+ = (S_i^+,B_i^+,W') \\
W_i^- = (S_i^- + B_i^-)(1+\mathbb{T}_i^g)\exp(-(\Delta t)_i\mathbb{T}^f) \\
S_i^+ = p_i(\cdot)W_i^+\,,\;\; B_i^+ = (1-p_i(\cdot))W_i^+ \\
(q_i(\cdot),p_i(\cdot)) \in \mathcal{Z}(W_i^-,W_i^+,t_i) \\
i = n,\ldots,M, \quad t_i\in\mathcal{T}
\end{cases} \qquad . \quad (3.35)
\end{aligned}
$$

3.2.2 Applying Dynamic Programming at Rebalancing Times

The principle of dynamic programming is applied at each $t_n \in \mathcal{T}$ on (3.35). As usual, the optimal control needs to be computed in reverse time order. We split the $\sup_{\mathcal{P}_n}$ operator into $\sup_{q\in\mathcal{Z}_q}\sup_{p\in\mathcal{Z}_p(w^- - q,t)}$.

$$
\begin{aligned}
V(s,b,W',t_n^-) \;&=\; \sup_{q\in\mathcal{Z}_q}\;\sup_{p\in\mathcal{Z}_p(w^- - q,t)}\left\{ q + \left[V((w^- - q)p,(w^- - q)(1-p),W',t_n^+)\right]\right\} \\
&=\; \sup_{q\in\mathcal{Z}_q}\left\{ q + \left[\sup_{p\in\mathcal{Z}_p(w^- - q,t)} V((w^- - q)p,(w^- - q)(1-p),W',t_n^+)\right]\right\} \\
&\qquad w^- = (s+b)(1+\mathbb{T}_i^g)\exp(-(\Delta t)_i\mathbb{T}^f)\;. \qquad (3.36)
\end{aligned}
$$

Let $\overline{V}$ denote the upper semi-continuous envelope of V, which will have already been computed as the algorithm progresses backward through time. The optimal allocation $p_n(w,W')$ at time t_n is then given by

$$
p_n(w,W') \;=\; \begin{cases}
\underset{p'\in[0,1]}{\arg\max}\,\overline{V}(wp',w(1-p'),W',t_n^+), & w>0, \quad t_n\neq t_M \\
0, & w\le 0 \;\text{ or } t_n = t_M
\end{cases} \qquad . \quad (3.37)
$$

Since we proceed backwards in time, the allocation control is determined first (in backwards time) followed by the withdrawal control q

$$q_n(w,W') = \arg\max_{q'\in\mathcal{Z}_q}\left\{q' + \overline{V}((w-q')p_n(w-q',W'),(w-q')(1-p_n(w-q',W')),W',t_n^+)\right\} . \tag{3.38}$$

Using these controls for t_n, the solution then moves from from t_n^+ to t_n^-

$$\begin{aligned} V(s,b,W',t_n^-) &= q_n(w^-,W') + \overline{V}(w^+p_n(w^+,W'),w^+(\,1-p_n(w^+,W')\,),W',t_n^+) + \epsilon(s+b) \\ & w^- = (s+b)(1+\mathbb{T}_i^g)\exp(-(\Delta t)_i\mathbb{T}^f),\quad w^+ = s+b-q_n(w^-,W') . \end{aligned} \tag{3.39}$$

At $t=T$, we have the terminal condition

$$V(s,b,W',T^+) = \kappa\left(W' + \frac{\min((s+b-W'),0)}{\alpha}\right) + \epsilon(s+b) . \tag{3.40}$$

3.2.3 Conditional Expectations between Rebalancing Times

For $t\in(t_{n-1},t_n)$, the tower property gives, for $0<h<(t_n-t_{n-1})$,

$$V(s,b,W',t) = E\Big[V(S(t+h),B(t+h),W',t+h)\Big|S(t)=s,B(t)=b\Big]\ ;\ t\in(t_{n-1},t_n-h) . \tag{3.41}$$

Assuming a parametric model of stock and bond SDEs, Ito's Lemma for jump processes (Tankov and Cont, 2009) is first applied to SDEs in Equations (3.13) and (3.14). This gives a partial integro differential equation (PIDE), as shown in Forsyth (2022) and Appendix B. In computational practice, the resulting PIDE is solved using Fourier methods discussed in Forsyth and Labahn (2019).

3.2.4 Equivalence with $PCEE_{t_0}(\kappa)$

Proceeding backward in time, the auxiliary function $V(s,b,W',t_0^-)$ is determined at time zero. Problem $PCEE_{t_0}(\kappa)$ is then solved using a final optimization step

$$J(s,b,t_0^-) = \sup_{W'} V(s,b,W',t_0^-) . \tag{3.42}$$

Notice that $V(s,b,W',t_0^-)$ denotes the auxiliary function for the beginning of the investment period, and represents the last step (going backward) in solving the dynamic programming formulation. To obtain this, we begin with Equation (3.40) and recursively work backwards in time. In the final step (going backwards), interchanging $\sup_{W'}$ and $\sup_{\mathcal{P}}$ gives Equation (3.33).

This formulation (3.35-3.41) is equivalent to problem $PCEE_{t_0}(\kappa)$. For a summary of computational details, refer to Appendix C or see Forsyth (2022).

3.3 Neural Network Formulation

As an alternative to the HJB framework, we develop a neural network framework to solve the stochastic optimal control problem (Equation 3.33), which has the following characteristics:

(i) The NN framework is data driven, which does not require a parametric model for traded assets being specified. This avoids explicitly postulating parametric stochastic processes and the estimation of associated parameters. In addition, this allows us to add auxiliary market signals/variables (although we do not exploit this idea in this work).

(ii) The NN framework avoids the computation of high-dimensional conditional expectations by solving for the control at all times directly from a single standard unconstrained optimization, without dynamic programming (see Van Staden et al. (2023) for a discussion of this). Since the control is low-dimensional, the approach avoids the *curse of dimensionality* by solving for the control directly, instead of via value iteration such as in the HJB dynamic programming method (Van Staden et al., 2023). Such an approach also eliminates backward error propagation through rebalancing times.

(iii) If the optimal control is a continuous function of time and state, the NN control will naturally reflect this property. If the optimal control is discontinuous, NN representation produces a smooth approximation. While not required by the original problem formulation in (3.33), this continuity property likely leads to practical implementation benefits.

(iv) The NN method is further scalable and can be easily adapted to problems with longer horizons or higher rebalancing frequency without significantly increasing the

computational complexity of the problem. This is in contrast to existing approaches using a stacked neural network approach (Tsang and Wong, 2020).

We now formally describe the proposed NN framework and demonstrate the aforementioned properties. We approximate the control in $\mathcal{P}$ directly by using feed-forward, fully-connected neural networks. Given parameters $\boldsymbol{\theta}_p$ and $\boldsymbol{\theta}_q$, i.e. NN weights and biases, $\hat{p}(W(t_i), t_i, \boldsymbol{\theta}_p)$ and $\hat{q}(W(t_i), t_i, \boldsymbol{\theta}_q)$ approximate the controls p_i and q_i respectively,

$$\begin{gathered}\hat{q}(W_i^-, t_i^-, \boldsymbol{\theta}_q) \simeq q_i(W_i^-), \quad i = 0, \ldots, M \\ \hat{p}(W_i^+, t_i^+, \boldsymbol{\theta}_p) \simeq p_i(W_i^+), \quad i = 0, \ldots, M-1 \\ \hat{\mathcal{P}} = \{(\hat{q}(\cdot), \hat{p}(\cdot))\} \simeq \mathcal{P}\end{gathered}$$

The functions $\hat{p}$ and $\hat{q}$ take time as one of the inputs, and therefore we can use just two NN functions to approximate control $\mathcal{P}$ across time instead of defining a NN at each rebalancing time. In this section, we discuss how we solve the problem (3.33) using this approximation and then provide a description of the NN architecture that is used. We discuss the precise formulation used by the NN, including activation functions that encode the stochastic constraints.

3.3.1 Neural Network Optimization for $PCEE_{t_0}(\kappa)$

We begin by describing the NN optimization problem based on the stochastic optimal control problem (3.33). We first recall that, in the formulation in Section 3.2, controls q_i and p_i are functions of wealth only. Our goal is to choose NN weights $\boldsymbol{\theta}_p$ and $\boldsymbol{\theta}_q$ by solving (3.33), with $\hat{q}(W_i^-, t_i^-, \boldsymbol{\theta}_q)$ and $\hat{p}(W_i^+, t_i^+, \boldsymbol{\theta}_p)$ approximating feasible controls $(q_i, p_i) \in \mathcal{Z}(W_i^-, W_i^+, t_i)$ for $t_i \in \mathcal{T}$. For an arbitrary set of controls $\hat{\mathcal{P}}$ and wealth level W', we define the NN performance criteria V_{NN} as

$$
\begin{aligned}
V_{NN}(\hat{\mathcal{P}}, W', s,b, t_0^-) \;\; &= \;\; E_{\hat{\mathcal{P}}_0}^{X_0^-, t_0^-}\Bigg[\sum_{i=0}^{M} \hat{q}_i \;+\; \kappa\Big(W' + \frac{1}{\alpha}\min(W_T - W', 0)\Big) \\
&\qquad\qquad +\epsilon W_T \Big| X(t_0^-) = (s,b) \Bigg]
\end{aligned}
$$

$$
\text{subject to } \begin{cases}
(S_t, B_t) \text{ follow processes (3.13) and (3.14)}, & t \notin \mathcal{T} \\
W_i^+ = W_i^- - q_i\,, \;\; X_i^+ = (S_i^+, B_i^+) & \\
W_i^- = (S_i^- + B_i^-)(1 + \mathbb{T}_i^g)\exp(-(\Delta t)_i \mathbb{T}^f) & \\
S_i^+ = \hat{p}_i(\cdot) W_i^+\,, \;\; B_i^+ = (1 - \hat{p}_i(\cdot)) W_i^+ & \\
(\hat{q}_i(\cdot), \hat{p}_i(\cdot)) \in \mathcal{Z}(W_i^-, W_i^+, t_i) & \\
i = 0, \ldots, M, \quad t_i \in \mathcal{T} &
\end{cases} \tag{3.43}
$$

The optimal value function J_{NN} (at t_0^-) is then given by

$$
J_{NN}(s, b, t_0^-) = \sup_{W'} \sup_{\hat{\mathcal{P}} \in \mathcal{A}} \; V_{NN}(\hat{\mathcal{P}}, W', s,b, t_0^-)\;. \tag{3.44}
$$

Next we describe the structure of the neural networks and feasibility encoding.

3.3.2 Neural Network Framework

Consider two fully-connected feed-forward NNs, with $\hat{p}$ and $\hat{q}$ determined by parameter vectors $\boldsymbol{\theta}_p \in \mathbb{R}^{\nu_p}$ and $\boldsymbol{\theta}_q \in \mathbb{R}^{\nu_q}$, representing NN weights and biases respectively. The two NNs can differ in the choice of activation functions and in the number of hidden layers and nodes per layer. Each NN takes input of the same form $(W(t_i),t_i)$, but the withdrawal NN $\hat{q}$ takes the state variable observed before withdrawal, $(W(t_i^-), t_i)$, and the allocation NN $\hat{p}$ takes the state variable observed after withdrawal, $(W(t_i^+), t_i)$.

In order for the NN to generate a feasible control as specified in (3.46), we use a modified sigmoid activation function to scale the output from the withdrawal NN $\hat{q}$ according to the $PCEE_{t_0}(\kappa)$ problem's constraints on the withdrawal amount q_i, as given in Equation (3.21). This ultimately allows us to perform unconstrained optimization on the NN training parameters.

Specifically, assuming $x \in [0,1]$, the function $f(x) = a + (b-a)x$ scales the output to $[a,b]$. We restrict withdrawal to $\hat{q}$ in $[q_{\min}, q_{\max}]$. We note that this withdrawal range

$q_{\max} - q_{\min}$ depends on wealth W^-, see from (3.21). Specifically, define the range of permitted withdrawal as follows,

$$\text{range} = \begin{cases} q_{\max} - q_{\min}, & \text{if } W_i^- > q_{\max} \\ W^- - q_{\min}, & \text{if } q_{\min} < W_i^- < q_{\max} \\ 0, & \text{if } W_i^- < q_{\min} \end{cases} .$$

More concisely, we have the following mathematical expression:

$$\text{range} = \max\left((\min(q_{\max}, W^-) - q_{\min}), 0\right) .$$

Let $z \in \mathbb{R}$ be the NN output before the final output layer of $\hat{q}$. Note that z depends on the input features, state, and time, before being transformed by the output activation function. We then have the following expression for the withdrawal,

$$\begin{aligned} \hat{q}(W^-, t, \boldsymbol{\theta_q}) &= q_{\min} + \text{range} \cdot \left(\frac{1}{1+e^{-z}}\right) \\ &= q_{\min} + \max\left((\min(q_{\max}, W^-) - q_{\min}), 0\right) \left(\frac{1}{1+e^{-z}}\right) . \end{aligned}$$

Note that the sigmoid function $\frac{1}{1+e^{-z}}$ is a mapping from $\mathbb{R} \to [0,1]$.

Similarly, we use a softmax activation function on the NN output for $\hat{p}$, ensuring no-shorting and no-leverage constraints are automatically satisfied.

With these output activation functions, it can be easily verified that $(\hat{q}_i(\cdot), \hat{p}_i(\cdot)) \in \mathcal{Z}\left(W_i^-, W_i^+, t_i\right)$ always. Using the defined NN, this transforms the problem (3.44) of finding an optimal $\hat{\mathcal{P}}$ into the optimization problem:

$$\begin{aligned} \hat{J}_{NN}(s, b, t_0^-) &= \sup_{W' \in \mathbb{R}} \sup_{\boldsymbol{\theta_q} \in \mathbb{R}^{\nu_q}} \sup_{\boldsymbol{\theta_p} \in \mathbb{R}^{\nu_p}} \hat{V}_{NN}(\boldsymbol{\theta}_q, \boldsymbol{\theta}_p, W', s, b, t_0^-) \\ &= \sup_{(W', \boldsymbol{\theta_q}, \boldsymbol{\theta_p}) \in \mathbb{R}^{\nu_q + \nu_p + 1}} \hat{V}_{NN}(\boldsymbol{\theta}_q, \boldsymbol{\theta}_p, W', s, b, t_0^-) . \end{aligned} \tag{3.45}$$

It is worth noting here that, while the original control $\mathcal{P}$ is constrained in (3.24), the formulation (3.45) is an unconstrained optimization over $\boldsymbol{\theta}_q$, $\boldsymbol{\theta}_p$, and W'. Hence we can solve the problem (3.45) directly using a gradient descent method. The numerical experiments detailed in later chapters use Adam stochastic gradient descent (Kingma and Ba, 2014) to determine the optimal points $\boldsymbol{\theta}_{\boldsymbol{q}}^*$, $\boldsymbol{\theta}_{\boldsymbol{p}}^*$, and W'.

Note that the output of NN $\hat{q}$ yields the amount to withdraw, while the output of NN $\hat{p}$ produces asset allocation weights.

Figure 3.1 presents the proposed NN. We emphasize the following key aspects of this NN structure.

(i) Time is an *input* to both NNs in the framework. The parameter vectors $\boldsymbol{\theta}_q$ and $\boldsymbol{\theta}_p$ are constant and do not vary with time.

(ii) At each rebalancing time, the wealth observation before withdrawal is used to construct the feature vector for $\hat{q}$. The resulting withdrawal is then used to calculate wealth after withdrawal, which is an input feature for $\hat{p}$.

(iii) Standard sigmoid activation functions are used at each *hidden layer* output.

(iv) The output activation function for withdrawal is different from the activation function for allocation. Control $\hat{q}$ uses a modified sigmoid function, which is chosen to transform its output according to (3.21). Control $\hat{p}$ uses a softmax activation which ensures that its output gives only positive weights for each portfolio asset and the weights sum to one, as specified in (3.22). By constraining the NN output this way through proposed activation functions, we can use unconstrained optimization to train the NN.

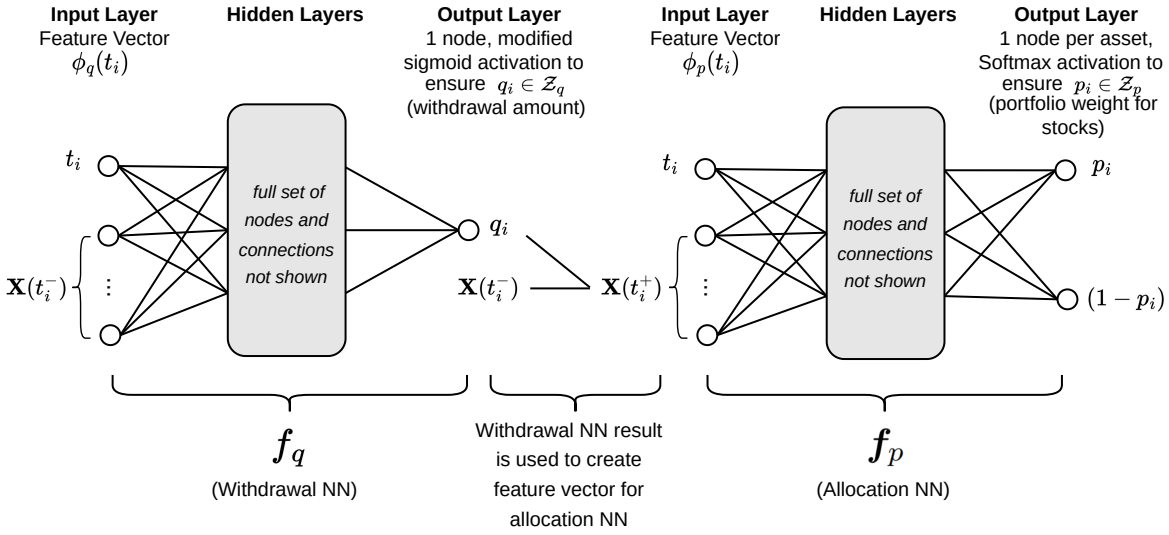


FIGURE 3.1: *Illustration of the NN framework as per Section 3.3.2. Additional technical details can be found in Appendix D.*

3.3.3 NN Estimate of the Optimal Control

Now we describe the NN training optimization problem for the decumulation problem, which is independent of the underlying data generation process. We assume that a set of asset return trajectories are available, which are used to approximate the expectation in (3.43) for any given control. For NN training, we approximate the expectation in (3.43) based on a finite number of samples as follows:

$$
\begin{aligned}
&\hat{V}_{NN}(\boldsymbol{\theta}_q,\boldsymbol{\theta}_p, W', s,b, t_0^-) = \\
&\frac{1}{N}\sum_{j=1}^{N}\left[\sum_{i=0}^{M}\hat{q}((W_i)^j, t_i; \boldsymbol{\theta}_q) \;+\; \kappa\left(W' + \frac{1}{\alpha}\min((W_T)^j - W', 0)\right) + \epsilon(W_T)^j \middle| X(t_0^-) = (s,b)\right] \\
&\text{subject to } \begin{cases}
((S_t)^j, (B_t)^j) \text{ drawn from the } j^{th} \text{ sample of returns;} \quad t \notin \mathcal{T} \\
(W_i^+)^j = (S_i^-)^j + (B_i^-)^j - \hat{q}\left((W_{t_i}^-)^j, t_i, \boldsymbol{\theta_q}\right)\,, \;\; (X_i^+)^j = (S_i^+, B_i^+)^j \\
(W_i^+)^j = (W_i^-)^j - \hat{q}_i\,, \;\; X_i^+ = (S_i^+, B_i^+, W') \\
(W_i^-)^j = \left((S_i^-)^j + (B_i^-)^j\right)(1 + \mathbb{T}_i^g)\exp(-(\Delta t)_i \mathbb{T}^f) \\
(S_i^+)^j = \hat{p}\left((W_i^+)^j, t_i, \boldsymbol{\theta_p}\right)\;(W_i^+)^j\,, \;\; (B_i^+)^j = (1 - \hat{p}\left((W_i^+)^j, t_i, \boldsymbol{\theta_p}\right))\;(W_i^+)^j \\
(\hat{q}_i(\cdot), \hat{p}_i(\cdot)) \in \mathcal{Z}\left((W_i^-)^j, (W_i^+)^j, t_i\right) \\
i = 0, \ldots, M, \quad t_i \in \mathcal{T}
\end{cases}
\end{aligned}
\tag{3.46}
$$

where the superscript j represents the j^{th} path of joint asset returns and N is the total number of sampled paths. For subsequent benchmark comparison, we generate price paths using processes (3.13) and (3.14). However, any method can be used to generate these paths. We are not restricted to parametric SDEs. We assume that the random sample paths are independent, but that correlations can exist between the returns of different assets. In addition, the correlation between the returns of different time periods can also be represented, e.g., block bootstrap resampling is designed to capture autocorrelation in the time series data.

The optimal parameters obtained by training the neural network are used to generate the control functions $\hat{q}^*(\cdot) := \hat{q}(\cdot; \boldsymbol{\theta}_{\boldsymbol{q}}^*)$ and $\hat{p}^*(\cdot) := \hat{p}(\cdot; \boldsymbol{\theta}_{\boldsymbol{p}}^*)$, respectively. With these functions, we can evaluate the performance of the generated control on testing data sets that are out-of-sample or out-of-distribution. We present the detailed results of such tests in Section 4.3.

Chapter 4

Simple Decumulation Problem

In order to validate the NN approach, we will first compare the NN solutions with the ground truth HJB solution for a simple case. We consider the case of an investor who does not participate in a tontine pool but otherwise determines the optimal withdrawals and asset allocation. This simply involves setting $\mathbb{T}^g = 0$ in Equation (3.33).

4.1 Data

For the computational study in this chapter, we use data from the Center for Research in Security Prices (CRSP) on a monthly basis from 1926:1 to 2019:12.[1] The specific indices used are the CRSP 10-year U.S. Treasury index for the bond asset[2] and the CRSP cap-weighted total return index for the stock asset[3]. Retirees are, naturally, concerned with preserving real (not nominal) spending power. Hence, we use the US CPI index (from CSRP) to adjust these indexes for inflation.

The result of calibrating the data to processes (3.13) and (3.14) from Forsyth (2022) are shown below in Table 4.1.

[1] More specifically, results presented here were calculated based on data from Historical Indexes, ©2020 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.

[2] The 10-year Treasury index was calculated using monthly returns from CRSP dating back to 1941. The data for 1926-1941 were interpolated from annual returns in Homer and Sylla (2005). The bond index is constructed by (i) purchasing a 10-year Treasury at the start of each month, (ii) collecting interest during the month and (iii) selling the Treasury at the end of the month.

[3] The stock index includes all distributions for all domestic stocks trading on major U.S. exchanges.

Calibrated Model Parameters

CRSP	μ^s	σ^s	λ^s	u^s	η_1^s	η_2^s	ρ_{sb}
	0.0877	0.1459	0.3191	0.2333	4.3608	5.504	0.04554
10-year Treasury	μ^b	σ^b	λ^b	u^b	η_1^b	η_2^b	ρ_{sb}
	0.0239	0.0538	0.3830	0.6111	16.19	17.27	0.04554

TABLE 4.1: *Calibrated (annualized) parameters for double exponential jump diffusion model. CPI adjusted CRSP US Total Market Index and CRSP US 10-year treasury, also inflation adjusted. Data from 1926:1 to 2019:12.*

4.2 Computational Results

We compute the optimal control using both the HJB PDE and NN method on synthetic data, with investment specifications given in Table 4.2. The parameters for the NN solution are given in Appendix D. We now present and compare the performance of the optimal control computed using the two methods. Each strategy's performance is measured w.r.t. to the objective function in (3.33), which is a weighted reward (EW) and risk (ES) measure. To trace out an efficient frontier in the (EW,ES) plane, we vary κ (the curve represents the (EW,ES) performance on a set of optimal Pareto points).

We first present strategies computed from the HJB framework described in Chapter 3. We verify that the numerical solutions are sufficiently accurate so that this solution can be regarded as ground truth. We then present strategies computed using the NN framework of Section 3.3, and demonstrate their accuracy by comparing them to the ground truth computed from the HJB equation. We carry out further analysis by selecting an *interesting* point on the (EW,ES) efficient frontier, in particular $\kappa = 1.0$, to study in greater detail. The point $\kappa = 1.0$ is at the *knee* of the efficient frontier, which makes it desirable in terms of risk-reward tradeoff (picking the exact κ will be a matter of investor preference, however). This notion of the knee point is loosely based on the concept of a *compromise solution* of multi-objective optimization problems, which selects the point on the efficient frontier with the minimum distance to an unattainable ideal point (Marler and Arora, 2004). For this knee point of $\kappa = 1.0$, we analyze the controls and wealth outcomes under both frameworks. We also discuss some key differences between the HJB and NN frameworks' results and their implications.

Retiree	65-year-old Canadian male
Tontine Gain $\mathbb{T}^g$	0
Tontine Fee $\mathbb{T}^f$	0
Investment horizon T (years)	30
Equity market index	CRSP Cap-weighted index (real)
Bond index	10-year Treasury (US) (real)
Initial portfolio value W_0	1000
Cash withdrawal times	$t = 0,1,\ldots,30$
Withdrawal range	$[35, 60]$
Equity fraction range	[0,1]
Borrowing spread μ_c^b	0.0
Rebalancing interval (years)	1
Market parameters	See Table 4.1

TABLE 4.2: *Problem setup and input data. Monetary units: thousands of dollars.*

4.2.1 Strategies Computed from HJB Equation

We carry out a convergence test for the HJB framework by tracing the efficient frontier (i.e. varying the scalarization parameter κ) for solutions of varying refinement levels (i.e. number of grid points in the (s,b) directions). Figure 4.1 shows these efficient frontiers. As the efficient frontiers from various grid sizes all practically overlap each other, this demonstrates the convergence of solutions computed from solving HJB equations. Table F.1 shows a convergence test for a single point on the frontier. The convergence is roughly first-order. This convergence test justifies the use of the HJB framework results as a ground truth.

Remark 4.1 (Effect of Stabilization Term ϵW_T)**.** *Recall the stabilization term, ϵW_T, introduced in (3.33). We now provide motivation for its inclusion, and observe its effect on the control $\hat{\mathcal{P}}$. When $W_t \gg W^*$ and $t \to T$, the control will only weakly affect the objective function. This is because, in this situation, $Pr[W_T < W^*] \simeq 0$ and thus the allocation control will have little effect on the ES term in the objective (recall that W^* is held constant for the induced time consistent strategy, see Appendix A). In addition, the withdrawal is capped at $q_{\max}$ for very high values of W_t, so the withdrawal control does not depend on W_t in this case either. The stabilization term can be used to alleviate the ill-posedness of the problem in this region.*

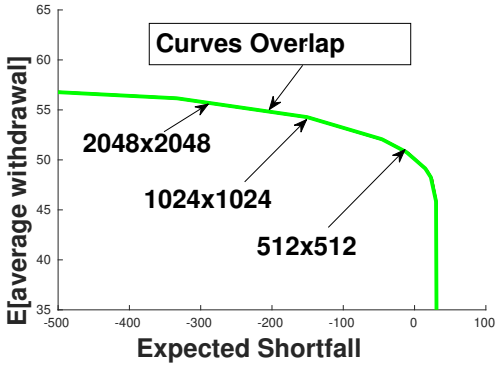
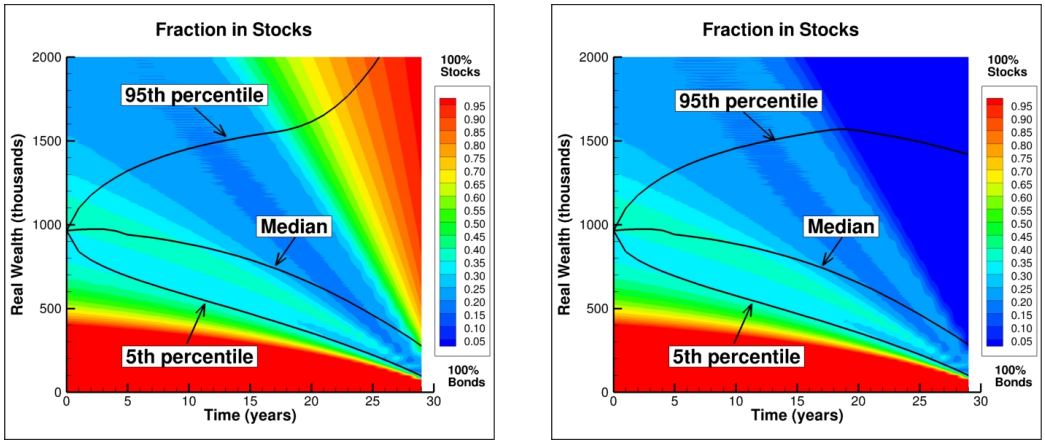


FIGURE 4.1: *EW-ES frontier, computed from problem (3.33). Note: Scenario in Table 4.2. Comparison of HJB solution performance with varying grid sizes. HJB solution performance computed on* 2.56×10^6 *observations of synthetic data. Parameters for synthetic data based on cap-weighted real CRSP, real 10-year treasuries (see Table 4.1).* $q_{min} = 35, q_{\max} = 60$. $\epsilon = 10^{-6}$. *Units: thousands of dollars.*

In Figure 4.2, we present the heat map of the allocation control computed from the HJB framework. 4.2a presents the allocation control heat map for a small positive stabilization parameter $\epsilon = 10^{-6}$, while 4.2b presents the allocation control heat map with $\epsilon = -10^{-6}$. In the ill-posed region (the top right region of the heat maps), the presence of ϵW_T, with $\epsilon = 10^{-6}$, forces the control to invest 100% in stocks to generate high terminal wealth. Conversely, changing the stabilization parameter to be negative ($\epsilon = -10^{-6}$) forces the control to invest completely in bonds at high wealth levels.

We observe that the control behaves differently only at a high level of wealth as $t \to T$ in both cases. The 5th and the 50th percentiles of control on the synthetic data set behave similarly in both the positive and negative ϵ cases. The 95th percentile curve tends towards higher wealth during later phases of the investment period when the ϵ is positive (Figure 4.2a), whereas the curve tends downward when ϵ is negative (Figure 4.2b). When the magnitude of ϵ is sufficiently small, its inclusion of ϵW_T in the objective function does not change summary statistics (to four decimal places when $|\epsilon| = 10^{-6}$). While the choice of negative or positive ϵ with small magnitude can lead to different allocation control scenarios at high wealth level lead near the end of the time horizon, the choice makes little difference from the perspective of the problem $PCEE_{t_0}(\kappa)$. If the investor reaches very high wealth near T, the choice between 100% stocks and 100% bonds does not matter as the investor always ends with $W_T \gg W^*$. Our experiments show that the control q is unaffected when the magnitude of ϵ is small and continues to call for maximum withdrawals at high levels of wealth as $t \to T$, just as described in Remark 4.1.

Comparing the optimal withdrawal strategy determined by solving stochastic optimal control problem (3.46) with a fixed withdrawal strategy (both strategies with dynamic asset allocation), Forsyth (2022) finds that the stochastic optimal strategy (3.46) is much more efficient in withdrawing cash over the investment horizon. Accepting a very small amount of additional risk, the retiree can dramatically increase total withdrawals. For a more detailed discussion of the optimal control, we refer the reader to Forsyth (2022).



a *Fraction in stocks, HJB Control,* $\epsilon = 10^{-6}$

b *Fraction in stocks, HJB Control,* $\epsilon = -10^{-6}$

FIGURE 4.2: *Effect of* ϵ*: fraction in stocks computed from the problem (3.33). Note: investment setup is as in Table 4.2. HJB solution performance computed on* 2.56×10^6 *observations of synthetic data. Parameters for synthetic data based on cap-weighted real CRSP, real 10-year treasuries (see Table 4.1).* $q_{min} = 35, q_{\max} = 60$*,* $\kappa = 1.0$*.* $W^* = 58.0$ *for PIDE results. (a)* $\epsilon = 10^{-6}$*. (b)* $\epsilon = -10^{-6}$*. Units: thousands of dollars.*

4.2.2 Accuracy of Strategy Computed from NN framework

We compute the NN control following the framework discussed in Section 3.3. We compare the efficient frontiers obtained from the HJB equation solution, and the NN solution. From Figure 4.3, the NN control efficient frontier is almost indistinguishable from the HJB control efficient frontier. Detailed summary statistics for each computed point on the frontier can be found in Appendix G.2, and a comparison of objective function values, for the NN and HJB control at each frontier point can be found in Appendix G.3. For most points on the frontier, the difference in objective function values, from NN and HJB, is less than 0.1%. This demonstrates that the accuracy of the NN framework approximation of the

ground-truth solution is more than adequate, considering that the difference between the NN solution and the PDE solution is about the same as the estimated PDE error (see Table F.1).

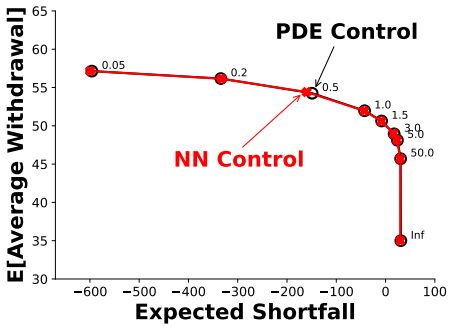


FIGURE 4.3: *Comparison of EW-ES frontier for the Neural Network (NN) and Hamilton-Jacobi-Bellman (HJB) Partial Differential Equation (PDE) methods, computed from the problem (3.33). Note: investment setup in Table 4.2. HJB solution performance computed on* 2.56×10^6 *observations of synthetic data. Parameters for synthetic data based on cap-weighted real CRSP, real 10-year treasuries (see Table 4.1). Control computed from the NN model, trained on* 2.56×10^6 *observations of synthetic data.* $q_{min} = 35, q_{\max} = 60$. $\epsilon = 10^{-6}$. *Units: thousands of dollars. Labels on nodes indicate* κ *parameter.*

We now further analyze the control $\hat{\mathcal{P}}$ produced by the NN framework for $\kappa = 1$. Comparing Figure 4.4b with Figure 4.4d, we observe that the withdrawal control $\hat{q}$ produced by the NN is practically identical that produced by the HJB framework. However, there are differences in the allocation control heat maps. The NN heat map for allocation control p (Figure 4.4a) appears most similar to that of the HJB allocation heat map for negative ϵ (Figure 4.2b), but it is clear that the NN allocation heat map differs significantly from the HJB heat map for positive ϵ (Figure 4.2a) at high level of wealth as $t \to T$. The NN allocation control behaves differently from the HJB controls in this region, choosing a mix of stocks and bonds instead of choosing a 100% allocation in a single asset. Noting this difference is only at a higher level of wealth near T, we see that the 5th percentile and the median wealth curves are indistinguishable. The NN control's 95th percentile curve, however, is different and indeed the curve is in between the 95th percentile curves from the negative and positive versions of the HJB-generated control.

Drawing from this, we attribute the NN framework's inability to fully replicate the HJB control to the ill-posedness of the optimal control problem in the (top-right) region of high wealth level near T. The small value of ϵ means that the stabilization term contributes

a very small fraction of the objective function value and thus has a very small gradient, relative to the first two terms in the objective function. Since we use stochastic gradient descent for optimization, we see a very small impact of ϵ. Moreover, the data for high levels of wealth as $t \to T$ is very sparse and so the effect of the small gradient is further reduced. As a result, the NN appears to smoothly extrapolate in this region and therefore avoids investment into a single asset. Recall that in Section 4.2.1, we stated that the choice in the signs of ϵ, with small ϵ, in the stabilization term is somewhat arbitrary and does not affect summary statistics. Therefore, we see that the controls produced by the two methods only differ in irrelevant aspects, at least based on the EW and ES reward-risk consideration.

It is interesting to observe that the proposed neural network framework is able to produce the *bang-bang* withdrawal control computed in Forsyth (2022), especially since we are using the continuous function $\hat{q}$ as an approximation.[4] A *bang-bang* control switches abruptly as shown here: the optimal strategy is to withdraw the minimum if the wealth is below a threshold, or else withdraw the maximum. As expected, the control threshold decreases as we move forward in time. We can see that the NN and HJB withdrawal controls behave very similarly at the 95th, 50th, and 5th percentiles of wealth (Figures 4.5c and 4.5f). Essentially, the optimal strategy withdraws at either $q_{\max}$ or $q_{\min}$, with a very small transition zone. This is in line with our expectations. By withdrawing less and investing more initially, the individual decreases the chance of running out of savings.

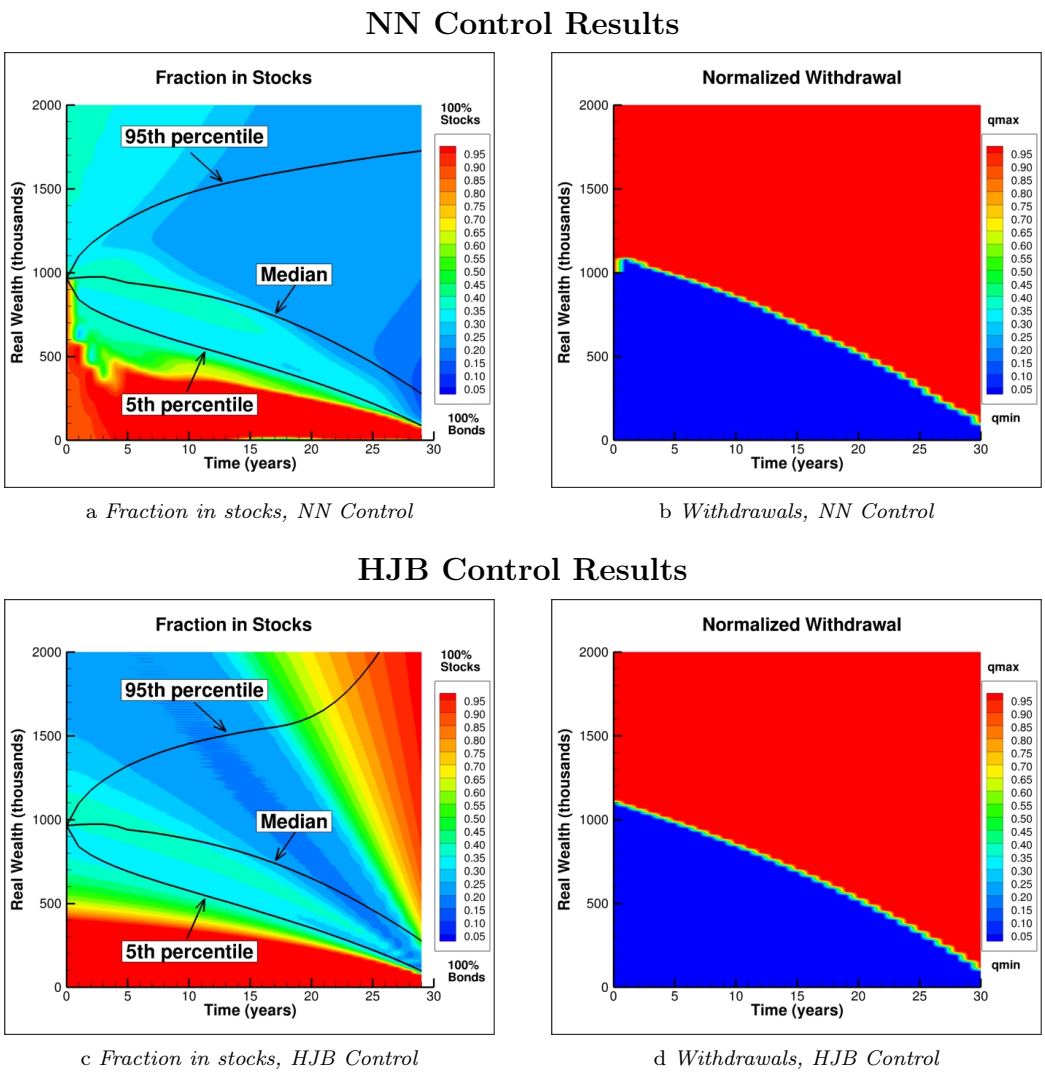
We also note that the NN allocation control presents a small spread between the 5th and 95th percentile of the fraction in stocks (Figure 4.5a). In fact, the maximum stock allocation for the 95th percentile never exceeds 40%, indicating that this is a stable low-risk strategy, which as we shall see, outperforms the Bengen (1994) strategy.

4.3 Model Robustness

A common pitfall of neural networks is over-fitting to the training data. Neural networks that are over-fitted do not have the ability to generalize to previously unseen data. Since future asset return paths cannot be predicted, it is important to ascertain that the computed strategy is not overfitted to the training data and can perform well on unseen return paths. In this section, we demonstrate the robustness of the NN model's generated controls.

We conduct three types of robustness tests: (i) out-of-sample testing, (ii) out-of-distribution testing, and (iii) control sensitivity to training distribution.

[4]Note that Forsyth (2022) shows that that in the continuous withdrawal limit, the withdrawal control is bang-bang.



a *Fraction in stocks, NN Control*

b *Withdrawals, NN Control*

c *Fraction in stocks, HJB Control*

d *Withdrawals, HJB Control*

FIGURE 4.4: *Heat map of controls: fraction in stocks and withdrawals, computed from the problem (3.33). Note: problem setup described in Table 4.2. HJB solution performance computed on* 2.56×10^6 *observations of synthetic data. Parameters for synthetic data based on cap-weighted real CRSP, real 10-year treasuries (see Table 4.1). NN model trained on* 2.56×10^6 *observations of synthetic data.* $q_{min} = 35, q_{\max} = 60$, $\kappa = 1.0$. $W^* = 59.1$ *for NN results.* $W^* = 58.0$ *for the HJB results.* $\epsilon = 10^{-6}$. *Normalized withdrawal* $(q - q_{\min})/(q_{\max} - q_{\min})$. *Units: thousands of dollars.*

4.3.1 Out-of-sample testing

Out-of-sample tests involve testing model performance on an unseen data set sampled from the same distribution. In our case, this means training the NN on one set of SDE paths sampled from the parametric model and testing on another set of paths generated using a different random seed. We present the efficient frontier generated by computed controls on this new data set in Figure 4.6, which shows almost unchanged performance on the out-of-sample test set.

4.3.2 Out-of-distribution testing

Out-of-distribution testing involves evaluating the performance of computed control on an entirely new data set sampled from a different distribution. Specifically, test data is not generated from the parametric model used to produce training data, but is instead bootstrap resampled from historical market returns via the method described in Section 2.4. We vary the expected block sizes to generate multiple testing data sets of 2.56×10^5 paths.

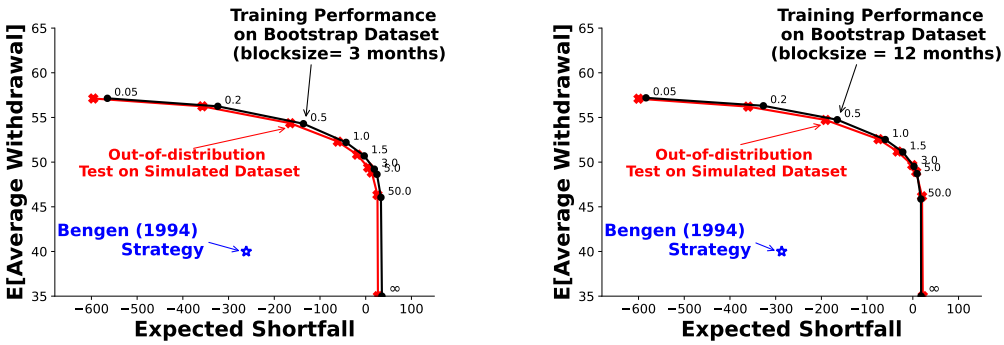
In Figure 4.7, we see that for each block size tested, the efficient frontiers are fairly close, indicating that the controls are relatively robust. Note that the efficient frontiers for test performance in the historical market with an expected block size of 1 and 3 months plot slightly above the synthetic market frontier. We conjecture that this may be due to more pessimistic tail events in the synthetic market.

The out-of-sample and out-of-distribution tests verify that the neural network is not over-fitting to the training data, and is generating an effective strategy, at least based on our block resampling data.

4.3.3 Control sensitivity to training distribution

To further test the NN framework's adaptability to other training data sets, we train the NN framework on historical data (with expected block sizes of both 3 months and 12 months) and then test the resulting control on synthetic data. In Figure 4.8, we compare the training performance and the test performance. The EW-ES frontiers for the test results on the synthetic data are very close to the results on the bootstrap market data (training data set). This shows the NN framework's adaptability to use alternative data sets to learn, with the added advantage of not being reliant on a parametric model, which is prone to miscalibration. Figure 4.8 also shows that, in all cases, in the synthetic or

historical market, the EW-ES control significantly outperforms the Bengen *4% Rule* [5] (Bengen, 1994).



a *Historical training data, block size = 3 months*

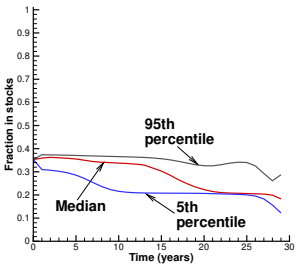
b *Historical training data, block size = 12 months*

FIGURE 4.8: *Training on historical data. EW-ES frontiers of controls generated by the NN model trained on* 2.56×10^5 *observations of historical data with expected block sizes of a) 3 months and b) 12 months, each tested on* 2.56×10^5 *observations of synthetic data. Parameters based on real CRSP index and real 10-year U.S. Treasuries (see Table 4.1). Historical data in range 1926:1-2019:12. Units: thousands of dollars.* $q_{min} = 35; q_{max} = 60$. *The Bengen (1994) results are based on bootstrap resampling of the historical data. Labels on nodes indicate* κ *parameter values. Simulated testing data refers to Monte Carlo simulations using the SDEs (3.13) and (3.14).*

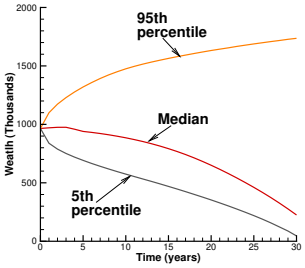
We began this chapter with a common scenario faced by the retiree. We show that in the simple case where the retiree has access to two assets (U.S cap-weighted stock index and U.S 10-year treasury bonds), the NN framework approximates the optimal dynamic decumulation strategy suggested by the HJB PDE approach in Forsyth (2022). We compared the differences between the two solutions and their impact on the overall strategy of the retiree. Lastly, we test the strategy proposed by the NN framework and establish its accuracy. Having established the accuracy of the NN framework, we turn our attention to a more complicated problem.

[5] The results for the Bengen strategy on the historical test data were computed with fixed 4% withdrawals and constant allocation of 30% in stocks for an expected block size of 3 months, and 35% in stocks for an expected block size of 12 months. These were found to be the best-performing constant allocations when paired with constant 4% withdrawals, in terms of ES efficiency.

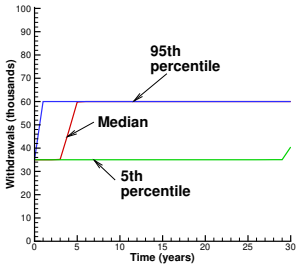
NN Control Results



a *Percentiles fraction in stocks, NN Control,* $\epsilon = 10^{-6}$

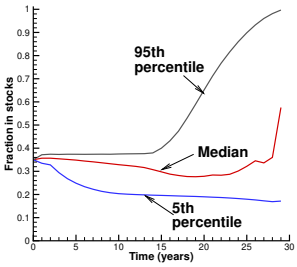


b *Percentiles wealth, NN Control,* $\epsilon = 10^{-6}$

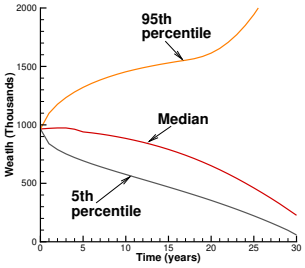


c *Percentiles withdrawals, NN Control,* $\epsilon = 10^{-6}$

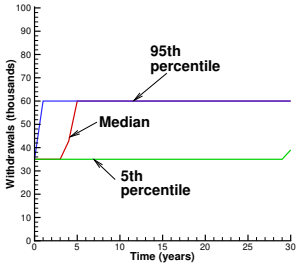
HJB Control Results (Positive and Negative Stabilization)



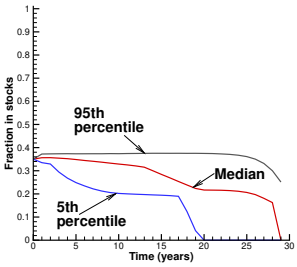
d *Percentiles fraction in stocks, HJB Control,* $\epsilon = 10^{-6}$



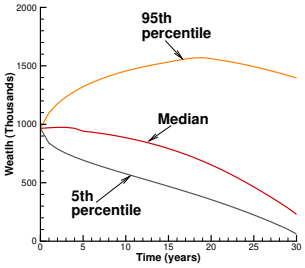
e *Percentiles wealth, HJB Control,* $\epsilon = 10^{-6}$



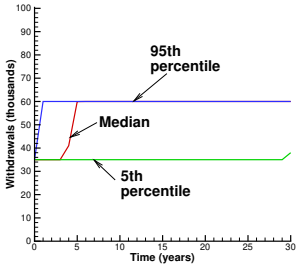
f *Percentiles withdrawals, HJB Control,* $\epsilon = 10^{-6}$



g *Percentiles fraction in stocks, HJB Control,* $\epsilon = -10^{-6}$



h *Percentiles wealth, HJB Control,* $\epsilon = -10^{-6}$



i *Percentiles withdrawals, HJB Control,* $\epsilon = -10^{-6}$

FIGURE 4.5: *Scenario in Table 4.2. NN and HJB controls computed from the problem (3.33). Parameters based on the real CRSP index, and real 10-year treasuries (see Table 4.1). NN model trained on* 2.56×10^5 *observations of synthetic data. HJB framework results from* 2.56×10^6 *observations of synthetic data.* $q_{min} = 35, q_{\max} = 60$, $\kappa = 1.0$. $W^* = 59.1$ *for NN results.* $W^* = 58.0$ *for HJB results. Units: thousands of dollars.*

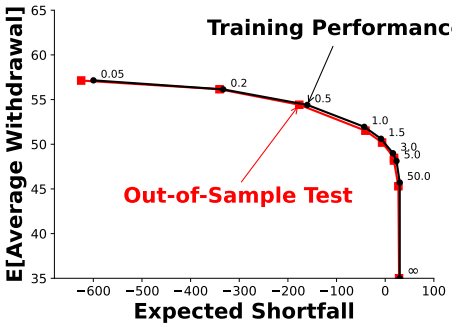


FIGURE 4.6: *Out-of-sample test. EW-ES frontiers, computed from the problem (3.33). Note: Scenario in Table 4.2. Comparison of NN training performance results vs. out-of-sample test. Both training and testing data are* 2.56×10^5 *observations of synthetic data, generated with a different random seed. Parameters for synthetic data based on cap-weighted real CRSP, real 10-year treasuries (see Table 4.1).* $q_{min} = 35, q_{\max} = 60$. $\epsilon = 10^{-6}$. *Units: thousands of dollars. Labels on nodes indicate* κ *parameter values.*

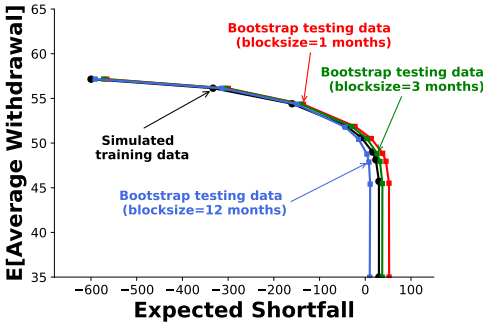


FIGURE 4.7: *Out-of-distribution test. EW-ES frontiers of controls generated by the NN model trained on* 2.56×10^5 *observations of synthetic data, tested on* 2.56×10^5 *observations of historical data with varying expected block sizes. Computed from the problem (3.33). Note: Setup as in Table 4.2. Parameters based on real CRSP index and real 10-year U.S. Treasuries (see Table 4.1). Historical data in range 1926:1-2019:12. Units: thousands of dollars.* $q_{min} = 35; q_{max} = 60$. *Simulated training data refers to Monte Carlo simulations using the SDEs (3.13) and (3.14).*

Chapter 5

Tontine

5.1 Description of tontine

We begin this section by building upon the results of Chapter 4, where we presented an optimal strategy for the decumulation of a DC plan using both HJB PDE and NN. In particular, Section 4.2.1 established that the resulting HJB PDE strategy outperformed the constant allocation strategy suggested by Bengen (1994). Furthermore, in Section 4.2.2 we establish that the NN framework produced similar results as those from the HJB PDE. In the following chapter, we will extend our experiment to other proposed solutions for the retirement problem.

A suggested improvement to retirement planning strategies has been the inclusion of tontine accounts, originally introduced in Section 2.5. We recall that investors in tontine make a binding investment into a pooled investment for a fixed time. In the scenario that the investor passes, their assets are redistributed among all the survivors in the tontine pool. This additional gain is commonly referred to as mortality credits. These mortality credits ensure that the investor receives a higher return than they would otherwise, in return for the added risk of investment forfeiture in the case of death. The real value of these mortality credits is itself a stochastic process, depending on the actual number of deaths. To avoid moral hazard issues, withdrawals from the tontine pool can not be increased from any agreed upon schedule (or agreed upon limit).

Forsyth et al. (2022) considers the case of individual tontine accounts that allow the investor to maintain control over their portion of investments in the asset pool. Forsyth et al. (2022) considers a 30-year period, with the rewards being represented by the total amount withdrawn (inflation-adjusted), while the risk is measured by the Expected

Shortfall at the 5% level at the end of the investment horizon. The ES at the 5% level represents the average of the worst 5% outcomes at the end of the 30-year period. The withdrawals are restricted to a range, instead of a fixed amount. Therefore, the investor has two controls; the amount invested in each asset, and how much to withdraw. Forsyth compares the effect of a tontine overlay by comparing the results with those without the tontine and established that there is a sizable reduction of risk of portfolio depletion by participating in a tontine pool.

We aim to extend our proposed NN framework to incorporate the tontine overlay. The mortality credits earned due to the tontine overlay add another source of randomness to the underlying process. By the end of this chapter, we intend to establish that the NN framework can provide accurate results even with a different wealth process and incorporating more uncertainties.

5.2 Data

We use monthly data from the Center for Research in Security Prices (CRSP) over the 1926:1-2020:12 period.[1] The specific indices used are the CRSP US 30-day T-bill for the bond asset and the CRSP value-weighted total return index for the stock asset[2]. All of these indexes are in nominal terms, so we adjust them for inflation by using the U.S. CPI index, also supplied by CRSP. We use real indexes since investors funding retirement spending should be focused on real (not nominal) wealth goals.

We use the above market data for stochastic model calibration and bootstrap resampling, the details of which have already been presented in Section 2.4. The result of calibrating models to processes (3.13) and (3.14) from Forsyth et al. (2022) are shown below in Table 5.1.

We also use the CPM2014[3] table from the Canadian Institute of Actuaries for incorporating mortality rates in the tontine overlay.

[1] More specifically, results presented here were calculated based on data from Historical Indexes, ©2020 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.

[2] The stock index includes all distributions for all domestic stocks trading on major U.S. exchanges.

[3] https://www.cia-ica.ca/docs/default-source/2014/214013e.pdf

Calibrated Model Parameters

CRSP	μ^s	σ^s	λ^s	u^s	η_1^s	η_2^s	ρ_{sb}
	0.08912	0.1460	0.3263	0.2258	4.3625	5.5335	0.08420
30 day T-bill	μ^b	σ^b	λ^b	u^b	η_1^b	η_2^b	ρ_{sb}
	0.0046	0.0130	0.5053	0.3958	65.801	57.793	0.08420

TABLE 5.1: *Estimated annualized parameters for double exponential jump diffusion model. Value-weighted CRSP index, 30-day T-bill index deflated by the CPI. Sample period 1926:1 to 2020:12.*

5.3 Results

We first briefly describe the investment specification that is used to generate the optimal control for both HJB PDE and NN in Table 5.2. Using a non-stochastic $G = 1$ for our base case corresponds to an infinitely large tontine pool, where idiosyncratic mortality risk can be ignored.

The optimal solution using the HJB PDE method was computed in Forsyth et al. (2022). We continue to use the same NN as in Chapter 4, the details of which can be found in Appendix D. We split the results into two parts. The first part is a comparison of portfolio performance with and without the tontine overlay. The second part establishes the accuracy of the NN solution by comparing it with that of the HJB PDE.

5.3.1 Effect of tontine overlay

We show the following three efficient EW-ES frontiers computed using the control generated by the NN framework on simulated data in Figure 5.1 below:

- Constant allocation and withdrawal: We vary the proportion of wealth invested into stocks keeping withdrawals constant at 40 to find the best-performing allocation. We plot the best result on the plot. The point performs better than the Bengen strategy. The detailed table can be found in H.1. We have labeled the point on the plot.
- No Tontine: This curve is traced out by varying κ in Equation (3.32), without a tontine overlay. Excluding the tontine overlay reduces the problem to that in Chapter 4.

Retiree	65-year-old Canadian male
Tontine Gain $\mathbb{T}^g$	Equation (3.10)
Group Gain G	1.0
Mortality Table	CPM 2014
Investment horizon T (years)	30
Equity market index	CRSP Cap-weighted index (real)
Bond index	30 day T-bill (US) (real)
Initial portfolio value W_0	1000
Cash withdrawal\rebalancing times	$t = 0,1,\ldots,29$
Withdrawal range	$[40, 80]$
Equity fraction range	[0,1]
Borrowing spread μ_c^b	0.02
Rebalancing interval (years)	1
Fees $\mathbb{T}^f$	50 bps per year
Stabilization ϵ	-10^{-4}
Market parameters	See 5.1

TABLE 5.2: *Problem setup and input data. Monetary units: thousands of dollars.*

- Tontine: This curve is traced out by varying κ in Equation (3.32) using the problem equation in (3.33).

We see a repetition of our previous result from Chapter 4 here, i.e.; the constant allocation and withdrawal strategy can be improved upon by dynamically withdrawing and allocating. However, our results here establish that the risk can be further reduced (established by an increase in ES) by participating in a tontine. Interestingly, almost all points on the curve yield an ES > 0, compared to the no tontine case where practically no point results in a positive ES. Of course, there is no free lunch here. This improvement in ES comes from pooling the longevity risk and incurring an additional risk, i.e.; the risk of losing all investment in case of death. We show the detailed tables for these frontiers in Appendix G.2.

5.3.2 Comparison of NN and HJB PDE solutions

We compare the performance of optimal control generated by the HJB PDE and the NN methods on simulated data. We carry a similar exercise as in Section 4.2.2 to establish the

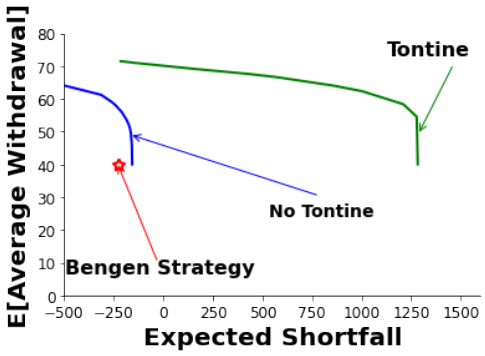


FIGURE 5.1: *EW-ES frontiers computed from the problem (3.33). Note: Scenario in Table 5.2. Comparison of NN training performance results for tontine vs no tontine. Both data sets are* 2.56×10^5 *observations of synthetic data. Const q, const p case has q=40 and p=0.10, with no tontine gains. Parameters for synthetic data based on cap-weighted real CRSP, real 30-day T-bills (see Table 5.1).* $q_{min} = 40, q_{\max} = 80$. $\epsilon = -10^{-4}$. *Units: thousands of dollars.*

accuracy of the NN method for a simple EW-ES optimization problem. Now we repeat the exercise to establish the accuracy of the solution produced by NN in the presence of a tontine overlay. We plot the EW-ES curves produced by the two methods and see that they are almost indistinguishable (Figure 5.2). The detailed results can be viewed in Appendix G.2 for both methods, and we see that the difference in the objective function is about the same as the estimated PDE error (see Appendix F.2).

We compare the allocation and withdrawal functions produced by the two methods for a fixed point as we did in Section 4.2.2 ($\kappa = 0.18, (EW, ES) = (68.61, 26.78)$). We can see that the two methods produce almost identical controls (Figure 5.3). Compared to the no-tontine case of Chapter 4, we see that the NN framework produces a better approximation of the PDE control. We conjecture that this is due to easier extrapolation in the case of $\epsilon < 0$ since there is no change in behavior at extremely higher wealth levels ($W > 2000$) compared to medium wealth level ($W \approx 1000$) We also compare the percentiles for the allocation and withdrawal controls, as well as those of the wealth generated by the two controls (Figure 5.4). We see that the results produced by the two controls are very similar. Both controls suggest around 60% allocation to stocks initially, with the 95th percentile staying at this level until the last few years (Figures 5.4d and 5.4a). Both controls produce very similar wealth levels on at the 95th, 50th and 5th percentiles (Figures 5.4b and 5.4e). The median withdrawals by both controls suggest withdrawing the maximum allowed amount and continuing to do so after the first few years (Figures 5.4c and 5.4f).

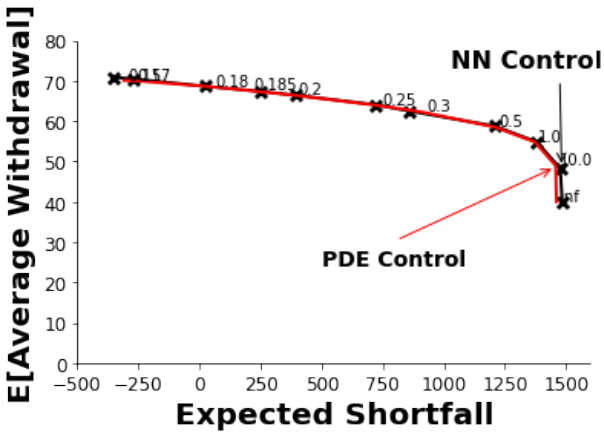


FIGURE 5.2: *Comparison of EW-ES frontier for the Neural Network (NN) and Hamilton-Jacobi-Bellman (HJB) Partial Differential Equation (PDE) methods, computed from the problem (3.33). Note: investment setup in Table 5.2. HJB solution performance computed on* 2.56×10^6 *observations of synthetic data (Forsyth et al., 2022). Parameters for synthetic data based on cap-weighted real CRSP, real 30-day T-bills (see Table 5.1). Control computed from the NN model trained on* 2.56×10^5 *observations of synthetic data.* $\epsilon = -10^{-4}$. *Monetary units:* USD *in thousands. Labels on nodes indicate* κ *parameter.*

5.4 Robustness test

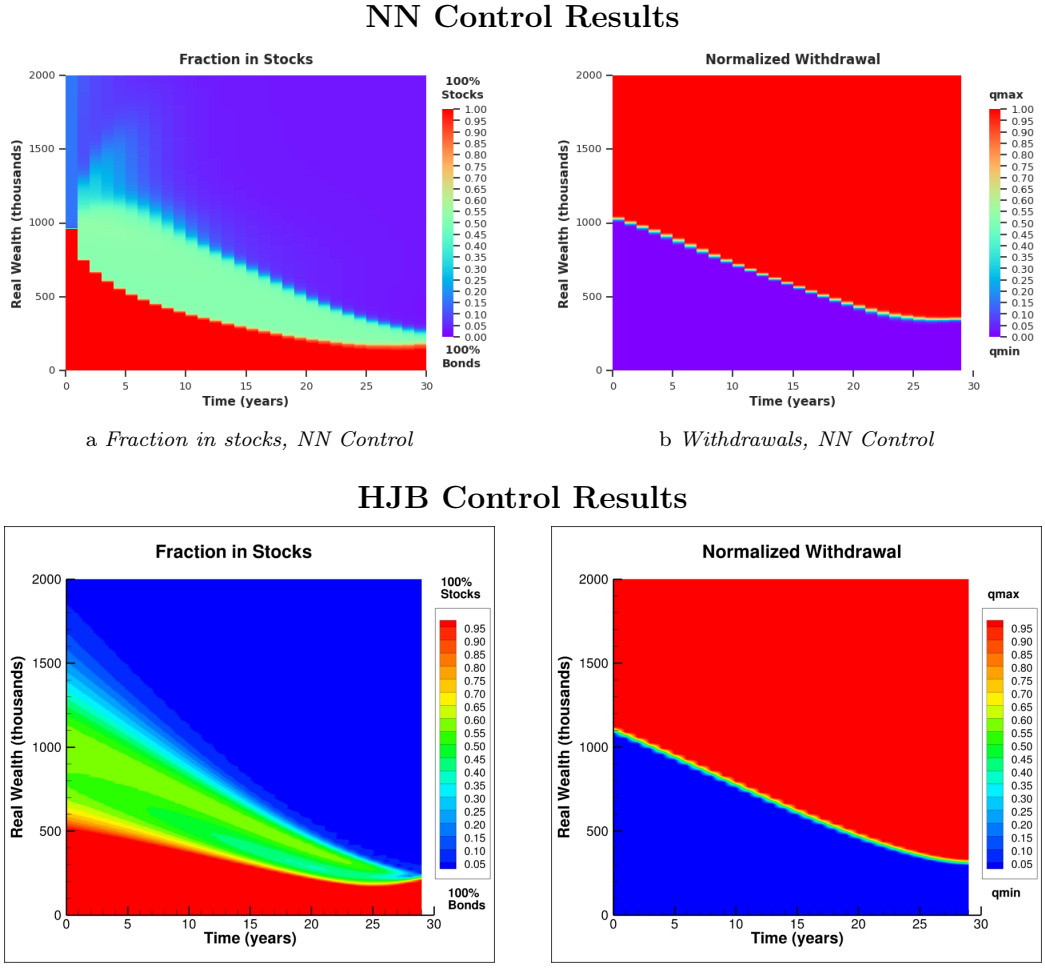
We test the robustness of the NN solution to establish that it is not overfitting to the data and is in fact generalizable to the underlying distributions (market models and historical). We recall the three tests first carried out in Section 4.3:

- Out of sample test
- Out of distribution test
- Control sensitivity to training distribution

In addition to this, we also test the robustness of the control as mortality rates are reduced.

5.4.1 Out of sample test

We test the solution proposed by the NN framework on an entirely new data set. We do so by changing the seed used to generate the test data set. We show the results in Figure



a *Fraction in stocks, NN Control*

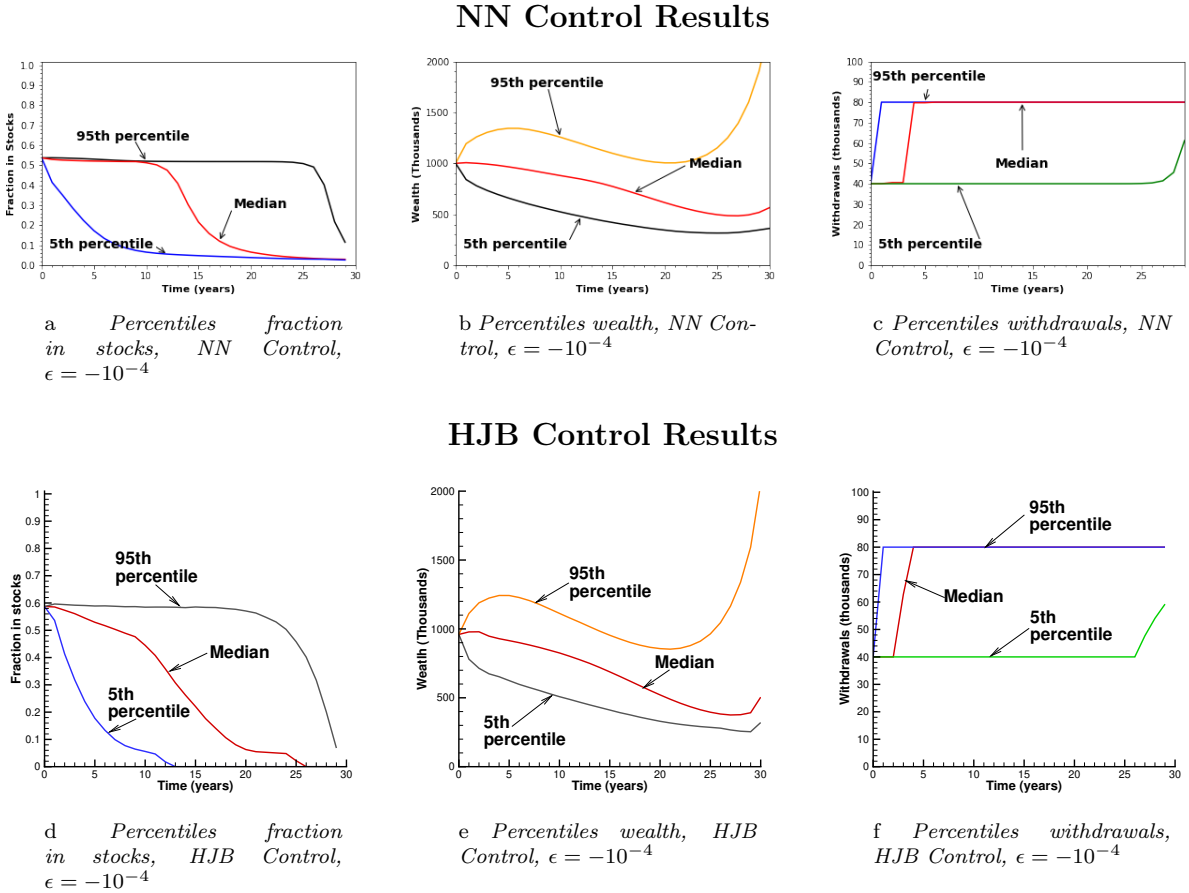
b *Withdrawals, NN Control*

c *Fraction in stocks, HJB Control*

d *Withdrawals, HJB Control*

FIGURE 5.3: *Heat map of controls: fraction in stocks and withdrawals, computed from the problem (3.33). Note: problem setup described in Table 5.2. HJB solution performance computed on* 2.56×10^6 *observations of synthetic data. Parameters for synthetic data based on cap-weighted real CRSP, real 10 year treasuries (see Table 5.1). NN model trained on* 2.56×10^5 *observations of synthetic data.* $q_{min} = 40, q_{\max} = 80$, $\kappa = 0.18$, $(EW, ES) =$ (68.61,26.78) $\epsilon = -10^{-4}$. *Normalized withdrawal* $(q - q_{\min})/(q_{\max} - q_{\min})$. *Units: thousands of dollars.*

5.5. We can see that the performance on the test data is very accurate and the two curves practically overlap.



a *Percentiles fraction in stocks, NN Control,* $\epsilon = -10^{-4}$

b *Percentiles wealth, NN Control,* $\epsilon = -10^{-4}$

c *Percentiles withdrawals, NN Control,* $\epsilon = -10^{-4}$

d *Percentiles fraction in stocks, HJB Control,* $\epsilon = -10^{-4}$

e *Percentiles wealth, HJB Control,* $\epsilon = -10^{-4}$

f *Percentiles withdrawals, HJB Control,* $\epsilon = -10^{-4}$

FIGURE 5.4: *Scenario in Table 5.2. NN and HJB controls computed from the problem (3.33). Parameters based on the real CRSP index, and real 30-day T-bills (see Table 5.1). NN model trained on* 2.56×10^5 *observations of synthetic data. HJB framework results from* 2.56×10^6 *observations of synthetic data.* $q_{min} = 40, q_{\max} = 80$, $\kappa = 0.18$. *Units: thousands of dollars.*

5.4.2 Out of distribution test

For this test, we use the NN model (trained on synthetic data) and test it on historical bootstrapped data. We see in Figure 5.6 that the performance on historical bootstrapped data is very good. We use an expected blocksize of two years as suggested in Forsyth et al. (2022). We see that the control performs slightly better for lower values of ES and worse for higher values of ES, particularly when $\kappa \to \infty$. However, these large values of κ are not of practical importance. The curves overlap for $ES < 1000$, which means that the 95

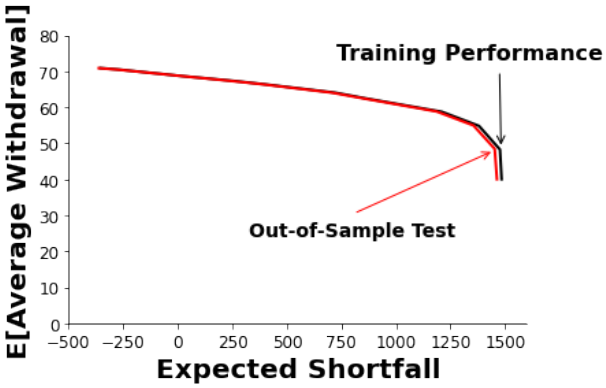


FIGURE 5.5: *Out-of-Sample test. EW-ES frontiers, computed from the problem (3.33). Note: Scenario in Table 5.2. Comparison of NN training performance results vs. out-of-sample test. Both training and testing data are* 2.56×10^5 *observations of synthetic data, generated with a different random seed. Parameters for synthetic data based on cap-weighted real CRSP, real 30-day T-bills (see Table 5.1).* $q_{min} = 40, q_{\max} = 80$. $\epsilon = -10^{-4}$. *Units: thousands of dollars.*

year old end up with one million dollars with high probability.

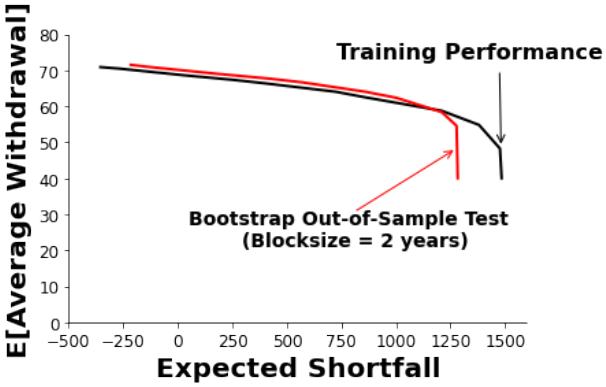


FIGURE 5.6: *Out-of-distribution test. EW-ES frontiers of controls generated by NN model trained on* 2.56×10^5 *observations of synthetic data, tested on* 2.56×10^5 *observations of historical data with expected block size = 2 years. Computed from the problem (3.33). Note: Setup as in Table 5.2. Parameters based on inflation-adjusted CRSP index and inflation-adjusted 30-day T-bills (see Table 5.1). Historical data between 1926:1 to 2020:12. Monetary units:* USD *in thousands.* $q_{min} = 40; q_{max} = 80$. $\epsilon = -10^{-4}$. *Units: thousands of dollars.*

5.4.3 Control Sensitivity to training distribution

In the previous section, we compared the performance of the NN control trained on simulated data and evaluated on bootstrapped historical data. In this section, we train the control on bootstrapped historical data and will evaluate it on the simulated data. This is a repeat of the experiment from Section 4.3.3 and is aimed at establishing the control's effectiveness in using real-world data for training. This confidence allows us to directly use historical data in the next chapter. While not discussed here, it must be remembered that parametric models are prone to misspecification and thus remain open to critique from practitioners.

We can see in Figure 5.7 that the control performs worse for all points on the simulated test set compared to the historical training set. However, the degradation in performance is not very significant and is still better than using a constant proportion and allocation strategy as suggested by Bengen.

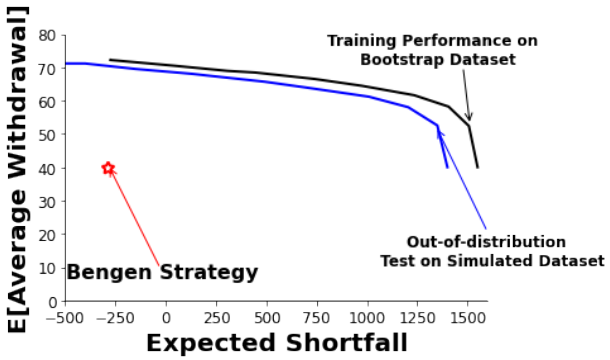


FIGURE 5.7: *Training on historical data. EW-ES frontiers of controls generated by NN model trained on* 2.56×10^5 *observations of historical data with expected block sizes of 2 years, tested on* 2.56×10^5 *observations of synthetic data. Parameters based on the inflation-adjusted CRSP index and inflation-adjusted 30-day T-bills (see Table 5.1). Note: Setup as in Table 5.2 Historical data between 1926:1 to 2020:12. Monetary units:* USD *in thousands. Minimum withdrawal: 40. Maximum withdrawal: 80. Bengen point Bengen (1994) is based on bootstrap resampling of the historical data.* $\epsilon = -10^{-4}$.

Having established the robustness of the solution for the base case, we further test our assumptions regarding the scenario being discussed. In particular, we discuss the effects of changes in systemic mortality. We expand on this in the next section.

5.4.4 Changes in Systemic Mortality

The first change that we consider to the scenario outlined in this chapter is related to changes in systemic mortality. The CPM2014 table published by the Canadian Institute of Actuaries incorporates expected improvements in mortality based on mortality data from the CPP. This could be brought upon by different factors, such as improvements in healthcare technology. We consider two scenarios in this section.

The first scenario is what we term as *unexpected* improvements in mortality. This means that the change in mortality is not incorporated into the problem (3.33). In this scenario, we use the control trained on the original problem and then test it on another data set with lower mortality rates. It can be readily seen that lower mortality rates would reduce the mortality credits received by the members of the pool. In our case, we consider a 10% reduction in all mortality rates. In particular, we define the actual mortality rate, $\mathfrak{q}_{act}$ as:

$$\mathfrak{q}_{\text{act}} = \mathfrak{q}_{\text{CPM2014}} \times 0.9 \tag{5.1}$$

We term the second scenario as *expected* improvements in mortality. This improvement is beyond what is already incorporated in the CPM2014 table. These improvements in mortality can be incorporated into training the control, and therefore, we can generate the optimal control for this case. We once again reduce the mortality rates by 10%.

We see in Figure 5.8 that the two controls (with reduced mortality) have very similar performances on the EW-ES frontier. This implies that the original control trained on (3.33) is robust to decreasing mortality rates in the future. Of course, the efficient frontiers with reduced mortality plot significantly below the base case, which is expected. The tontine members bear this systematic risk. This is a very crude test of the effect of systemic risk. A more sophisticated approach would use a stochastic model of mortality improvements such as in Dahl (2004). We would then incorporate this information as an extra state variable and determine the optimal control, similar to our analysis in the next section.

5.5 Effect of Random G

In Section 3.1.3, we explained the effect of differences in expected and actual number of deaths. To account for the differences in these values, we incorporated the Group Gain, G, on the mortality credits earned. Fullmer and Sabin (2018) show that the variance in

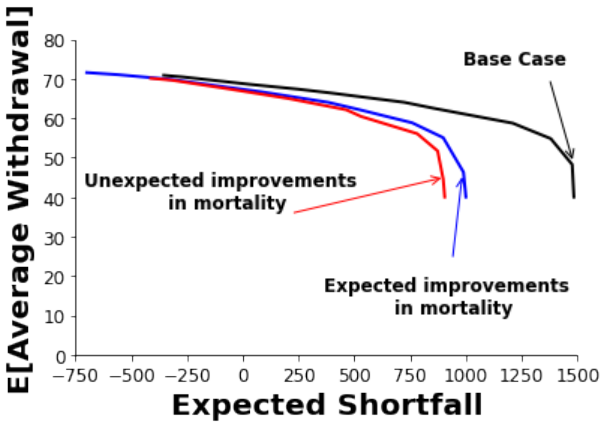


FIGURE 5.8: *Effect of varying mortality rates. EW-ES frontiers, computed from the problem (3.33). Note: Scenario in Table 5.2. Comparison of NN training performance results on synthetic data with lower mortality vs base case performance on synthetic data with lower mortality rates. Parameters for synthetic data based on cap-weighted real CRSP, real 30-day T-bills (see Table 5.1).* $q_{min} = 40, q_{\max} = 80$. $\epsilon = -10^{-4}$. *Units: thousands of dollars.*

group gain decreases as the members in the tontine pool increase. Surprisingly, only 10000 members can lead to a small standard deviation of 0.1. Forsyth et al. (2022) shows that the effect of this variance is negligible on large tontine pools by applying the control from the base case (constant G) to Monte Carlo simulations with randomized group gain. Since we were considering a large tontine pool, it was sufficient to assume the group gain to be a constant. However, it will be prudent to consider the effect of smaller tontine pools. Fullmer and Sabin (2018) show that the standard deviation can go as high as 0.39 for a pool size of 1000. In such a scenario, the control from the base case may not be optimal. Incorporating the randomness in the group gain introduces an extra state variable and thus would increase the complexity of the HJB PDE framework. In contrast to the method in Forsyth et al. (2022), the NN framework would not face this problem. In fact, if we had a stochastic model for the randomness in G, we could simulate the values of G and use them to train the NN. We can add G as an additional state variable to our control p (i.e. define $p(\cdot) = p(W, G, t)$), but, in some sense, the group gain is an additional random variable affecting wealth (i.e. has the same effect of adding more randomness to the wealth SDE), with a more pronounced effect on older retirees. We, therefore, continue to use the same definition for our control p (i.e. $p(\cdot) = p(W,t)$).

Fullmer and Sabin (2018) further show that the group gain is uncorrelated with portfolio returns. In such a case, it would be sufficient to model the effect using a randomly generated

value for group gain. Since the uniform distribution is the most conservative for a given mean and standard deviation, we use this to model the group gain G. The effect can be seen by plotting the EW-ES efficient frontier in Figure 5.9. Notice that using a standard deviation of 0.4 is a very extreme test since it corresponds to a pool size smaller than 1000 members. We see that increasing the uncertainty in group gain leads to lower performance in terms of the objective function, however, the effect is small, and noticeable for the tail end of our efficient frontier. It must be noted that the points impacted by this all have ES > 1000, which is more than what the retiree has at the start of the time horizon. More importantly, it must be seen in Figure 5.10 that even by the most conservative assumptions on group gain, participating in the tontine outperforms a simple dynamic decumulation strategy and the benchmark constant allocation strategy.

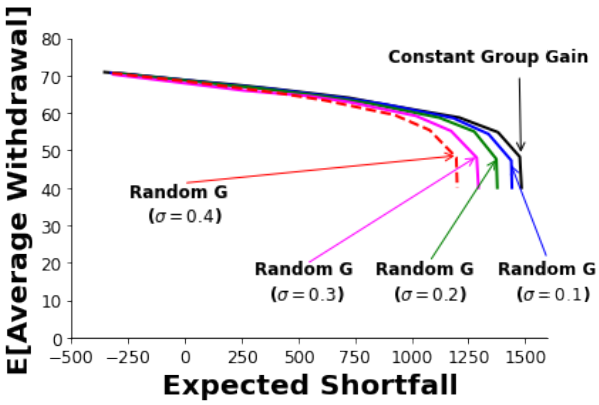


FIGURE 5.9: *Effect of Random group gain G. EW-ES frontiers, computed from the problem (3.33). Note: Scenario in Table 5.2. Comparison of NN training performance results on different random distributions of group gain G.Training data is 2.56×10^5 observations of synthetic data. Parameters for synthetic data based on cap-weighted real CRSP, real 30-day T-bills (see Table 5.1). $q_{min} = 40, q_{\max} = 80$. $\epsilon = -10^{-4}$. Units: thousands of dollars*

We now move towards analyzing how training the NN with the randomness in the group gain G changes the control. For this purpose, we compare the controls produced by high uncertainty in group gain ($\sigma = 0.4$) with the controls produced with a constant group gain ($G = 1$). We choose points on the curves with similar ES and where the two curves do not overlap. For the base case, we choose the point $(856.95, 62.5)$ and for the modified control we choose the point $(916.93, 59.5)$.

We first analyze the control for the allocation control p. We see that compared to the base case ($G = 1$), the modified control (random G) is very similar and only has a couple of minor differences. Firstly, we see that there is a higher investment in stocks in

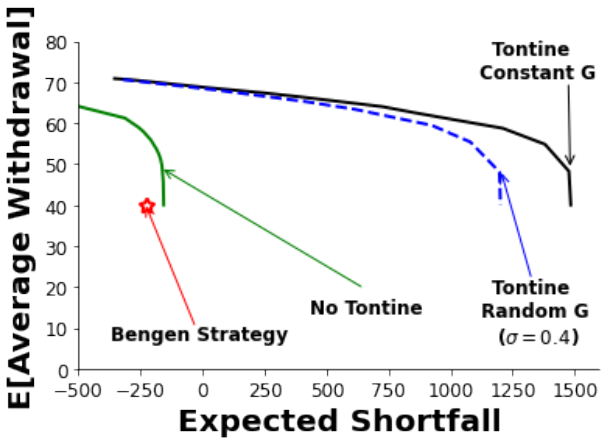


FIGURE 5.10: *Effect of Random group gain G vs no tontine vs Bengen. EW-ES frontiers, computed from the problem (3.33). Note: Scenarios in Table 5.2. No tontine scenario does not consider mortality credits but dynamically allocates and withdraws. Comparison of NN training performance results on different random distributions of the group gain G. Training data is* 2.56×10^5 *observations of synthetic data. Parameters for synthetic data based on cap-weighted real CRSP, real 30-day T-bills (see Table 5.1).* $q_{min} = 40, q_{\max} = 80$. $\epsilon = -10^{-4}$. *Units: thousands of dollars*

the earlier stages of investment (Figures 5.11c and 5.11a) at higher wealth levels. This difference, however, lies in the region beyond the 95th percentile of wealth (Figures 5.12e and 5.12b) and represents improbable situations. We can further see in Figure 5.12d that for the modified control the investment in stocks in early years is actually lower compared to the base case in Figure 5.12a. It must also be noted that in the case of modified control, we start by allocating only 30% in stocks compared to the base case which allocates almost 40% in stocks. This is in line with what we expect. If there is no uncertainty regarding future tontine gains, one may be convinced to take on more risk initially and continue to invest in stocks even when the performance is poor. However, uncertainty in the future would mean that the retiree wants to take less risk since mortality credits in the future can not be relied upon. For both cases, we see that the 95th percentile of allocation rises to around 50% and stays high until later years. However, the 5th percentile of investment decreases quickly after 5 years for both scenarios. While the investment in the base case falls gradually to around 5%, the modified control takes a more aggressive approach in reducing investment into stocks and maintains investment in stocks at 15%.

Another notable difference is that the base case suggests investing 50% in stocks towards the end of the time horizon at higher wealth levels compared to the modified control which

suggests investing 100% in bonds at similar wealth levels (Figures 5.11a and 5.11c). We can also see this in the percentiles in Figures 5.12a and 5.12d that towards the end of the time horizon, the base case increases investment in stocks while the modified control decreases investment in stocks. This could also be explained by the high amount of uncertainty in mortality credits at later stages of the investment horizon. When faced with high uncertainty in the case of modified control, the control suggests low variance bonds. When the retiree has guaranteed high returns due to mortality credits (as in the base case), it would make sense to take on more risk.

We now move to see how the wealth changes by incorporating the uncertainty in the group gain G. We see in Figures 5.12b and 5.12e that the 95th, 50th, and 5th percentiles of wealth follow almost the same trajectory. This is no coincidence since the two controls also had very similar expected shortfalls.

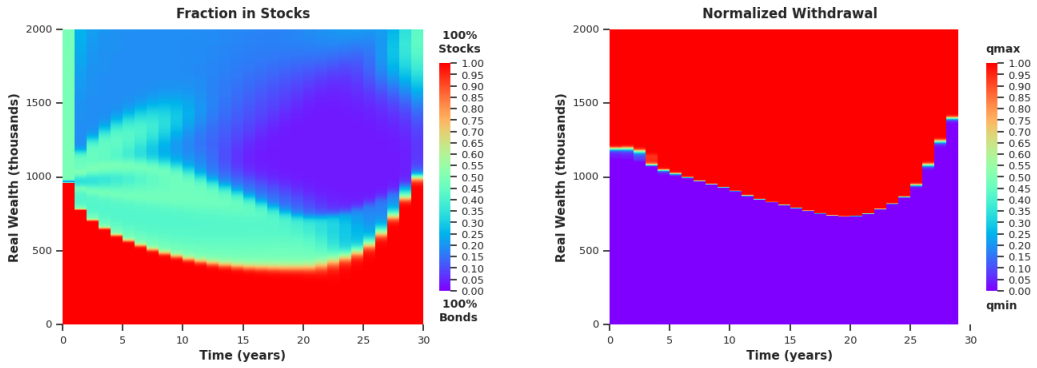
We next move to analyze the withdrawal control q. We see in Figures 5.11b and 5.11d that the two controls behave similarly for the most part with two notable differences. The bang-bang boundary in the modified control is higher than in the base case. This is expected since every withdrawal is a trade-off between current consumption and future wealth, and uncertainties in future wealth would move one to consume less. It must be noted that this uncertainty is not the same associated with uncertainty in other investments. Mortality credits almost always lead to an increase in wealth, so uncertainty in mortality credits signifies that the increase in wealth could be less compared to the base case. In such a scenario, it makes sense that the retiree would prefer to withdraw at the minimum levels and not bank too much on future gains. We see this behavior in the percentile plots for withdrawal in Figures 5.12c and 5.12f. While the behavior at the 5th percentile is the same, we see that the 95th percentile and median levels rise slower to the maximum withdrawals in the modified control compared to the base case.

We see that the control changed significantly when trained with a random G and that the NN framework does well in modifying the suggestion in response to this randomness. This also suggests that using the control from the base case may not represent an optimal strategy when there are other sources of randomness.

5.6 Conclusion

We started this chapter with a discussion of a scenario incorporating tontines in the retirement problem. A member in a tontine pool has a lower risk of running out of funds since they receive an additional return in the form of mortality credits. However, they incur a

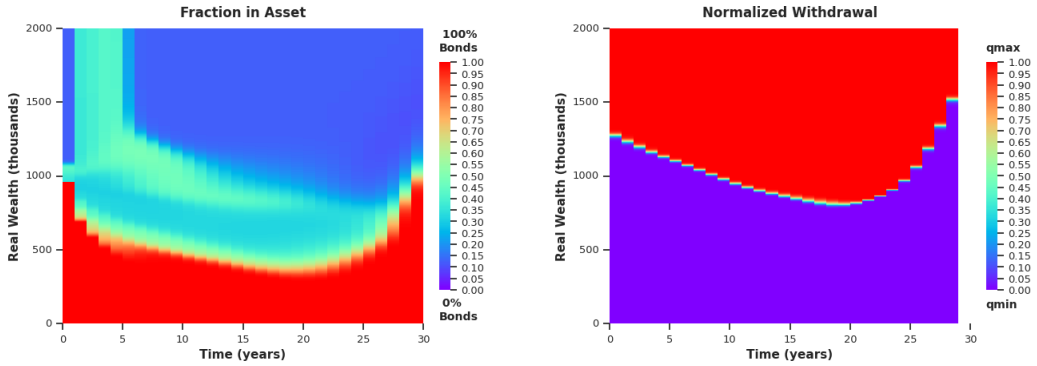
Constant $G = 1$



a *Fraction in stocks, NN Control*

b *Withdrawals, NN Control*

Random $G(\sigma = 0.4)$



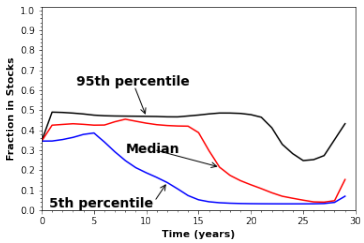
c *Fraction in stocks, NN Control*

d *Withdrawals, NN Control*

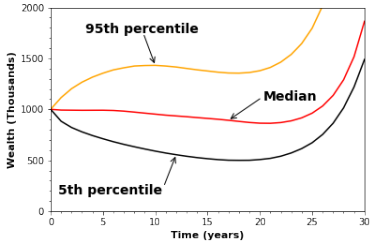
FIGURE 5.11: *Heat map of controls: fraction in stocks and withdrawals, computed from the problem (3.33). Note: problem setup described in Table 5.2. Parameters for synthetic data based on cap-weighted real CRSP, real 30-day T-bills (see Table 5.1). NN model trained on* 2.56×10^5 *observations of synthetic data.* $q_{min} = 40, q_{\max} = 80$, $\kappa = 0.3$ *for the base case and* $\kappa = 0.5$ *for the random* G. $\epsilon = -10^{-4}$. *Normalized withdrawal* $(q - q_{\min})/(q_{\max} - q_{\min})$. *Units: thousands of dollars.*

greater risk of wealth forfeiture in case of death. Forsyth et al. (2022) quantified the value of this risk by comparing performances on both the synthetic and historical market data. We repeat the experiment here but with our proposed NN framework and see the NN very closely approximates the provably convergent HJB PDE framework presented in Forsyth et al. (2022). We further test the robustness of the NN solution by making modifications

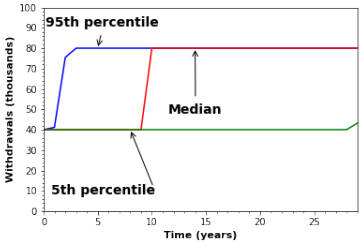
Constant $G = 1$



a *Percentiles fraction in stocks, NN Control,* $\epsilon = -10^{-4}$

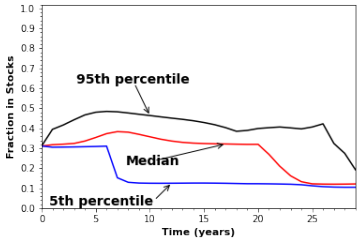


b *Percentiles wealth, NN Control,* $\epsilon = -10^{-4}$

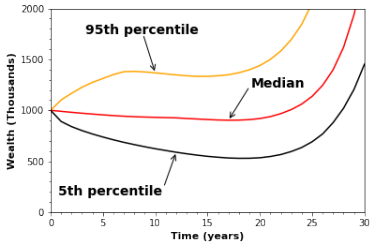


c *Percentiles withdrawals, NN Control,* $\epsilon = -10^{-4}$

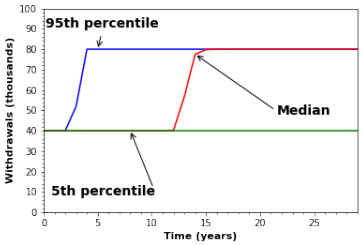
Random $G(\sigma = 0.4)$



d *Percentiles fraction in stocks, NN Control,* $\epsilon = -10^{-4}$



e *Percentiles wealth, NN Control,* $\epsilon = -10^{-4}$



f *Percentiles withdrawals, NN Control,* $\epsilon = -10^{-4}$

FIGURE 5.12: *Scenario in Table 5.2. NN controls computed from the problem (3.33). Parameters based on the real CRSP index, and real 30-day T-bills (see Table 5.1). NN model trained on* 2.56×10^5 *observations of synthetic data.* $q_{min} = 40, q_{\max} = 80$, $\kappa = 0.3$ *for the base case and* $\kappa = 0.5$ *for the random* G. *Units: thousands of dollars.*

to the retirement scenarios and using historical data sets for training. This builds upon the results in Chapter 4 where we were able to approximate results for a simple decumulation scenario. Lastly, we extend the NN framework to incorporate another systemic uncertainty in the state space and observed how the control changes in response. It should be noted that this increase in problem complexity (and state space) did not require us to change the structure of the two Neural Networks. This lends us confidence that the NN framework can be used in more complicated scenarios where the HJB PDE framework may not be very efficient. We consider one such scenario in the next chapter where we introduce more assets for investment.

Chapter 6

Multi-Asset Tontine

This chapter builds upon the scenario that was set up in Chapter 5. So far we have established that a dynamic allocation and decumulation strategy outperforms any constant allocation strategy such as that suggested by Bengen (1994). We further showed that participation in a tontine pool significantly reduces the risk of running out of money during retirement, while increasing the amount of withdrawals. In all these scenarios, we considered two assets that were available to the retiree; a bond index and a stock index. An advantage of the NN method is that it is feasible to explore scenarios with a large number of assets. Therefore, in this chapter, we will consider examples with several assets.

6.1 Scenario

We begin this chapter by recalling the retirement scenario we have been considering in this thesis. We discuss a 65-year-old retiree with a substantial amount of money from a DC retirement plan. It is necessary to manage money effectively to sustain retirement. Each year, the retiree withdraws money within certain limits. The portfolio's value fluctuates based on returns and mortality credits. If the value becomes negative, assets are sold, trading stops, and debt accrues. The retiree's goal is to maximize withdrawals while avoiding running out of savings. It is, of course, desirable to avoid insolvency.

This scenario differs from previous chapters in two main ways: the investor has more assets available, and the assets span different national markets. We now move to a discussion on how this changes our formulation of the problem.

6.2 Problem Formulation

We now extend the problem described in Equation (3.33) by adding more assets. Let the number of assets available to the investor be N_a. We denote the real (inflation-adjusted) amount (in CAD) invested in asset i at time t by $A_i(t)$. Then the amounts invested in all assets at time t can be represented by the set $\mathbf{A}(t)$ where:

$$\mathbf{A}(t) = \{A_i(t) : i = 1, 2 \dots N_a\} \ . \tag{6.1}$$

Without loss of generality, we let the first asset, $A_1(t)$ denote the amount in risk-free bonds at time t, which will also serve as the benchmark for our borrowing rate. From Equation (6.1), it easily follows that the total wealth $W(t)$ of the investor is given by:

$$W(t) = \sum_{i=1}^{N_a} A_i(t) \ . \tag{6.2}$$

We recall that the finite set of discrete withdrawal/rebalancing times $\mathcal{T}$ is represented as:

$$\mathcal{T} = \{t_0 = 0 < t_1 < t_2 < \ldots < t_M = T\} \ . \tag{6.3}$$

At each rebalancing time $t_i \in \mathcal{T}$, the investor first withdraws an amount q_i, and then rebalances the portfolio. At time T, the portfolio is liquidated. We assume that there are no taxes and no transaction costs. These assumptions are justified since retirement accounts are generally tax-exempt and trading is infrequent. We continue to use the shorthand notation for a time-dependent function $f(t)$:

$$f(t_i^+) \equiv \lim_{\epsilon \to 0^+} f(t_i + \epsilon) \ , \qquad f(t_i^-) \equiv \lim_{\epsilon \to 0^+} f(t_i - \epsilon) \ \ . \tag{6.4}$$

We assume that the dynamics surrounding tontines stay the same as described in Section 3.1. In particular, a tontine fee $\mathbb{T}^f$ is charged and the mortality credits earned at time t_i can be represented by:

$$\mathbb{T}_i^g \ = \ G_i\left(\frac{\mathfrak{q}_{i-1}^j}{1 - \mathfrak{q}_{i-1}^j}\right) \tag{6.5}$$

and G_i is the group gain for period (t_{i-1}, t_i) as discussed in Section 3.1.3.

Let $X(t) = \{A_i(t) : i = 1, 2 \dots N_a\}$, $t \in [0,T]$ denote the multidimensional controlled underlying process. Following typical notation, let $x = \{a_i(t) : i = 1, 2 \dots N_a\}$ denote the the realized state of the system.

At each rebalancing time t_i, the investor receives mortality credits, and withdraws the amount $q_i(\cdot)$, determined by the control at time t_i; that is, $q_i(\cdot) = q_i(X(t_i^-)) = q(X(t_i^-), t_i)$. This control is used to evolve the investment portfolio from W_t^- to W_t^+

$$W(t_i^+) = W(t_i^-)(1 + \mathbb{T}_i^g) \exp(-(\Delta t)_i \mathbb{T}^f) - q_i \ , \qquad t_i \in \mathcal{T} \ . \tag{6.6}$$

With some notational abuse, we define $W(t_i^-)$ as the wealth after tontine gains $\mathbb{T}_i^g$ and fee $\mathbb{T}^f$, and at the instance before withdrawal q_i:

$$W(t_i^-) = \left(\sum_{k=1}^{N_a} A_k(t_i^-) \right) (1 + \mathbb{T}_i^g) \exp(-(\Delta t)_i \mathbb{T}^f), \qquad t_i \in \mathcal{T} \ . \tag{6.7}$$

The withdrawal and allocation controls are formally functions of the state before withdrawal, $X(t_i^-)$. However, it is useful to note that the allocation control is specifically a function of the state after withdrawal. This is simply due to the fact that rebalancing occurs after the withdrawal. Let $p_k(\cdot)$ represent the fraction of wealth in asset k, after rebalancing

$$A_k(t_i^+) \quad = \quad p_k(X(t_i^+), t_i^+) W(t_i^+) \ . \tag{6.8}$$

The full underlying control $\mathbf{p}(X(t_i), t_i)$ can then be described as

$$\mathbf{p}(X(t_i^+), t_i^+) \quad = \quad \{p_k(X(t_i^+), t_i^+) : k = 1, 2, \dots N_a\} \ . \tag{6.9}$$

Without of loss generality, we let p_1 denote the proportion of wealth invested in the risk-free bonds, which also serve as the benchmark for our borrowing rate. As formulated in Section 3.1, assuming no transaction costs, the control depends on wealth only, i.e., $p_k(\cdot, i) = p_k(X(t_i^+), t_i) = p_k(W_i^+, t_i))$. Therefore, we make another notational adjustment for the sake of simplicity and consider $q_i(\cdot)$ to be a function of wealth before withdrawal, W_i^-, and $\mathbf{p}(\cdot, i)$ to be a function of wealth after withdrawal, W_i^+.

We assume instantaneous rebalancing, with the implication that the control at time t_i is described by a pair $(q_i(\cdot), \mathbf{p}(\cdot, i)) \in \mathcal{Z}(W_i^-, W_i^+, t_i)$, where $\mathcal{Z}(W_i^-, W_i^+, t_i)$ represents

the set of admissible control values for t_i. The constraints on the allocation control are no shorting, no leverage (except in an insolvent state). There are minimum and maximum values for the withdrawal. In the normal course of events, the no-shorting and no-leverage constraints imply that wealth is always positive. However, due to minimum withdrawals at rebalancing times, it is possible for insolvency to occur. In this case, no stock holdings are permitted, and debt accumulates at the borrowing rate. Any subsequent withdrawals are restricted to the minimum amounts. Any non-zero stock positions are liquidated at terminal time. We can mathematically state these constraints by imposing suitable bounds on the value of the controls as follows:

$$\mathcal{Z}_q(W_i^-,t_i) = \begin{cases} [q_{\min}, q_{\max}] \ ; \ t_i \in \mathcal{T} \ ; \ W_i^- > q_{\max} \\ [q_{\min}, W_i^-] \ ; \ t_i \in \mathcal{T} \ ; \ q_{\min} < W_i^- < q_{\max} \\ \{q_{\min}\} \ ; \ t_i \in \mathcal{T} \ ; \ W_i^- < q_{\min} \end{cases} , \tag{6.10}$$

$$\mathcal{Z}_{\mathbf{p}}(W_i^+,t_i) = \left\{ \mathbf{p} \in \mathbb{R}^{N_a} \middle| \begin{cases} \sum_{k=1}^{N_a} p_k = 1 & t_i \neq t_M \\ p_k \in [0,1] | k = \{1, \dots N_a\} & W_i^+ > 0 \ ; \ t_i \in \mathcal{T} \ ; \ t_i \neq t_M \\ \mathbf{p} = \{1, 0, \dots 0\}, & W_i^+ \leq 0 \ ; \ t_i \in \mathcal{T} \ ; \ t_i \neq t_M \\ \mathbf{p} = \{1, 0, \dots 0\} & t_i = t_M \end{cases} \right. \tag{6.11}$$

$$\mathcal{Z}(W_i^-, W_i^+, t_i) = \mathcal{Z}_q(W_i^-,t_i) \times \mathcal{Z}_{\mathbf{p}}(W_i^+,t_i) \ . \tag{6.12}$$

Note that p_1 is the fraction in risk-free bonds, which also serve as the benchmark for our borrowing rate. While the controls for both withdrawal and allocation are formally a function of wealth and time before withdrawal (W_i^-,t_i), but for implementation purposes, it will be helpful to write the allocation as a function of wealth and time after withdrawal (W_i^+,t_i).

The admissible control set $\mathcal{A}$ can be written as

$$\mathcal{A} = \left\{ (q_i, \mathbf{p}(\cdot,i))_{0 \leq i \leq M} : (q_i, \mathbf{p}(\cdot,i)) \in \mathcal{Z}(W_i^-, W_i^+, t_i) \right\} \tag{6.13}$$

An admissible control $\mathcal{P} \in \mathcal{A}$, can be written as

$$\mathcal{P} = \{ (q_i(\cdot), \mathbf{p}(\cdot,i)) \ : \ i = 0, \dots, M \} \ . \tag{6.14}$$

It will sometimes be necessary to refer to the tail of the control sequence at $[t_n, t_{n+1}, \dots, t_M]$, which we define as

$$\mathcal{P}_n = \{(q_n(\cdot), \mathbf{p}(\cdot, n)) \ldots, (q_M(\cdot), \mathbf{p}(\cdot, M))\} \ . \tag{6.15}$$

Since we only use the NN framework to solve for the above optimal control, we now define the NN approximations to the controls q and $\mathbf{p}$:

$$\begin{gathered}
\hat{q}(W_i^-, t_i^-, \boldsymbol{\theta}_q) \simeq q_i(W_i^-), \quad i = 0, \ldots, M-1 \\
\hat{\mathbf{p}}(W_i^+, t_i^+, \boldsymbol{\theta}_p) \simeq \mathbf{p}(W_i^+, t_i^+), \quad i = 0, \ldots, M-1 \\
\hat{\mathcal{P}} = \{(\hat{q}(\cdot), \hat{\mathbf{p}}(\cdot))\} \simeq \mathcal{P}
\end{gathered}$$

We continue to use the same NN structure for the control q as we did in the previous chapters, i.e. two inputs representing W_i^- and t_i with sigmoid activation functions for hidden layers and a modified sigmoid function for the output layer as described in Section 3.3. For our control $\mathbf{p}$, we maintain the same structure for the NN but add more nodes to the hidden layers. We continue to use the softmax activation function for the output layer to ensure that $p_i \in [0,1] \forall i$. We give additional details in Appendix D. We now seek to find an optimal control $\hat{\mathcal{P}}^*$ using our new NN.

6.2.1 Objective Function

We continue to use the same objective function from Equation (3.32) using a weighted sum of EW and ES :

$$\mathrm{EW}(X_0^-, t_0^-) + \kappa \mathrm{ES}_\alpha(X_0^-, t_0^-) \ . \tag{6.16}$$

With the new definitions outlined in the previous section, we now provide the NN objective function that will be used to find the optimal control. Incorporating the above information, we transform Equation (3.46) to:

$$\hat{V}_{NN}(\boldsymbol{\theta_q},\boldsymbol{\theta_p},W^*,X(t_0^-),t_0^-) =$$
$$\frac{1}{N}\sum_{j=1}^{N}\left[\sum_{i=0}^{M}\hat{q}((W_i)^j,t_i;\boldsymbol{\theta}_q)+\kappa\left(W^*+\frac{1}{\alpha}\min((W_T)^j-W^*,0)\right)+\epsilon(W_T)^j\middle|X(t_0^-)=(W_0^-,t_0^-)\right]$$
$$\text{subject to}\begin{cases}(R_t)^j \text{ drawn from the } j^{th} \text{ sample of returns;}\quad t\notin\mathcal{T}\\(W_i^+)^j=(W_i^-)^j-\hat{q}\left((W_{t_i}^-)^j,t_i,\boldsymbol{\theta_q}\right)\ ;\ (X_i^+)^j=(t_i^+,W_i^+)^j\\(W_i^-)^j=(\sum_{k=1}^{N_a}(A_k)^j)(1+\mathbb{T}_i^g)\exp(-(\Delta t)_i\mathbb{T}^f)\\(A_k^+)^j=(W_i^+)^j)\hat{p}_k\left((W_{t_i}^-)^j,t_i,\boldsymbol{\theta_p}\right)\\(\hat{q}_i(\cdot),\hat{\mathbf{p}}(\cdot,i))\in\mathcal{Z}\left((W_i^-)^j,(W_i^+)^j,t_i\right)\\i=0,\ldots,M\ ;\ t_i\in\mathcal{T}\end{cases},\quad(6.17)$$

where $\boldsymbol{\theta_q}$ and $\boldsymbol{\theta_p}$ represent the parameters of the two controls.

6.3 Data

The computational analysis in this chapter utilizes monthly returns data from the Canadian Financial Markets Research Centre[1] spanning the period from November 1950 to December 2022. The specific indices employed include the Canadian Treasury Bill Index and long-term Government of Canada bond index for bonds[2] and Canadian equal-weighted equity index, the Canadian value-weighted equity index, the US value-weighted equity index, and US equal-weighted equity index for stocks[3]. These indices are expressed in real terms in Canadian dollars with USD/CAD conversion based on the average exchange rate data on the last day of the month from the Bank of Canada[4]. We reduce the returns for equal-weighted indices for both US and Canadian markets by 50bps[5] due to higher costs associated with these indices. By employing real indices, we emphasize the importance of considering inflation when investors plan for retirement and aim to achieve real wealth

[1] https://www.cfmrc.org/

[2] Long term bond index represents the return on a long term GoC bond with an assumed term to maturity of 17 years purchased at the end of last month (at par, or $P_{t-1} = 100$) and sold at the end of this month.

[3] US indexes from CRSP (https://www.crsp.org/)

[4] https://www.bankofcanada.ca/rates/exchange/daily-exchange-rates-lookup/

[5] Management expense fee based on the SPDR Global Dow ETF (https://www.etf.com/DGT)

objectives rather than nominal ones. We use the Canadian CPI data provided by Statistics Canada[6] to adjust these indexes to real terms.

In this chapter, we only consider historical bootstrapped data for training and evaluation purposes. While it is certainly possible to use simulated data like we did in earlier chapters, we avoid that use here given that we have established the NN framework's independence from the data source.

6.4 Results

We briefly describe the investment specification that is used to generate the optimal control for the NN in Table 6.1.

Investment horizon T (years)	30
Equity market indices	U.S\Canada Cap-weighted\equal-weighted index (real)
Bond index	Canadian 30-day\10-year Treasury index (real)
Initial portfolio value W_0	1000
Cash withdrawal times	$t = 0,1,\ldots,29$
Withdrawal range	$[40, 80]$
Equity fraction range	$[0,1]$
Borrowing spread μ_c^b	0.02
Rebalancing interval (years)	1
Tontine Fees $\mathbb{T}^f$	50 bps per year
Tontine Gain $\mathbb{T}^g$	Equation (3.10)
Stabilization ϵ	-10^{-4}

TABLE 6.1: *Problem setup and input data. Monetary units: thousands of dollars.*

We now present and compare the performance of the control when the investor invests in more than two assets. We compare the performance against the base case of two assets (T-bills and US value-weighted index) as discussed in Chapter 5. We also provide the performance of a constant proportion strategy where equal proportions are invested into each asset and a constant withdrawal of 4% of initial investment is made each year. We see in Figure 6.1 that the control's performance improves by adding multiple assets compared

[6]https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=1810000401)

to the base case. This is in line with our expectation, as these assets span across multiple markets, and hence provide better diversification.

We take a more detailed look by analyzing one point on the EW-ES frontier. In particular, we take the point corresponding to (ES =46.25, EW =77.16) ($\kappa = 0.03$) since it has a low positive expected shortfall and thus corresponds to high withdrawals, in line with the retiree's preferences. We see that the allocation control suggests a portfolio selection that is primarily dominated by three assets: Canadian T-bills, the Canadian equal-weighted index, and the US value-weighted index. We show the control allocations for these in Figure 6.3. It is easy to see from the heat maps that the NN suggests investing in Canadian 30-day T-bills at higher levels of wealth (Figure 6.3a). This is because, at higher levels of wealth, the 30-day T-bills represent a safe low-risk option. This does not come as a surprise since it is well-known that short-term T-bills provide a better hedge against inflation compared to long-term bonds (Spierdijk and Umar, 2015). We see in Figure 6.2a that the median level of investment in 30-day T-bills rises as we progress in time and can be more than 90% towards the end of our investment period. This is perhaps because we see high levels of wealth at both the median and 95th percentile (Figure 6.2d) and thus a low-risk asset is preferred. We also see that the 5th percentile of investment also increases towards the end of the time horizon.

Similarly, we see that the high-risk high-return option preferred by the control at low wealth levels is the Canadian equal-weighted index (Figure 6.3b). In fact, the 95th percentile of investment into the Canadian equal-weighted index goes as high as 60%, and even the 5th percentile has an investment level of 10%. This can be explained by the high returns that this index has enjoyed in the past (Figure 6.4). We remind the reader that the retiree receives high mortality credits during the terminal years and thus can afford to take on the additional risk brought about by investing in the Canadian equal-weighted index in the earlier years. Interestingly, we see that at levels of wealth around $500,000 to $900,000, the control suggests investing into both the Canadian equal-weighted and the US value-weighted index. This split is almost equal (Figures 6.3b and 6.3c) and represents diversification by the control. Historically, we can see that investments in equal-weight indexes have been a good idea (Figure 6.4) and the US equal-weighted index outperforms the US value-weighted index, but the control chooses not to invest in it. However, observing the index performance over the last ten years (Figure 6.5) shows us that value-weighted indexes have outperformed equal-weighted indexes, with the US value-weighted index outperforming the other three indexes. The last ten years also represent a period of poor performance of the Canadian equal-weighted index. We can not comment on whether things have changed permanently or if this period is just an outlier[7]. What we do see,

[7]The most expensive words in the English language are "This time it's different".

however, is that of the correlations between returns of the different indexes in Table 6.2, the returns of the US value-weighted index and the Canadian equal-weighted index have the lowest correlation among all combinations of indexes. The control suggests reducing investment in both stock indexes during the end of the time horizon. This may be because the point we are concerned with represents very low positive ES, and thus it is preferable to take on less risk. Risky investments could make the retiree insolvent, and thus lower the withdrawals. However, since mortality credits can be as high as 28% during the terminal years, it is safer to invest only in bonds and rely on these mortality credits to increase wealth. We see an increase in wealth levels during the terminal years at the 95th, 50th, and 5th percentile levels of wealth (Figure 6.2d). This increase in wealth allows the retiree to withdraw at the highest level during the terminal years (Figure 6.2e).

We further look at the withdrawal control suggested by the NN, presented in Figure 6.3d. Similar to observations in the previous chapters, we see that the control suggests a bang-bang withdrawal strategy, with the threshold for maximum withdrawals being lowered as we progress forward in time. This is in line with our expectations as the opportunity cost of withdrawals is lower in later years. We look at the percentiles for withdrawals (Figure 6.2e), suggesting that the retiree withdraws the maximum allowed amount throughout the 30-year period in more than 50% of paths. This represents a considerable improvement over the two asset case we considered in Chapter 5 where this was not possible even at the 95th percentile (Figure 5.4c). Even at the 5th percentile of withdrawals, the retiree can withdraw the maximum amount for the last eight years.

We see that investing in multiple assets considerably improves performance compared to investing in only two assets (Figure 6.1). We can see that the NN is able to diversify some of the risk and invests in assets that have a lower correlation. It is also important to see how the NN diversification performs in historical backtests that are common among practitioners.

For this particular scenario, we split the historical dataset into halves. We use the first half of this data to generate the training dataset (1950:11 - 1985:12), and the second half to generate the testing data set (1986:01 - 2022:12). We use the historical bootstrapping technique (blocksize = 2 years) as described in Section 2.4 to ensure that we have a sufficiently large number of sample paths from both the data sets. We show the results in Figure 6.6.

It can be seen that the performance of the control is very similar for the two datasets, and this gives us confidence that the NN framework is not overfitting to the data and suggests a good diversification strategy. The degradation in performance on the test data set is not significant compared to the training data set.

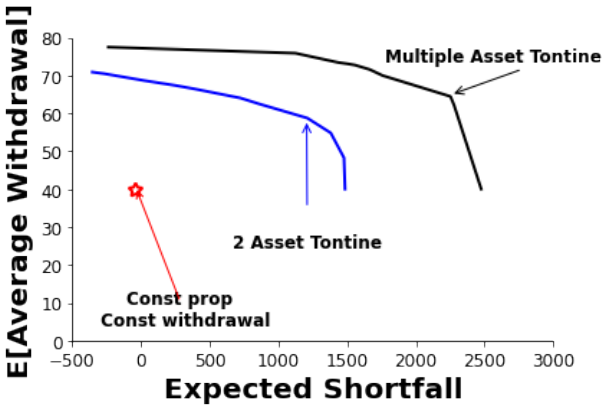
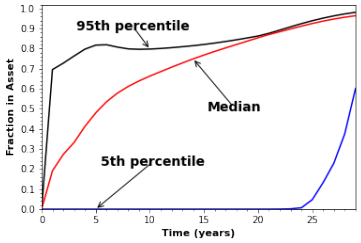


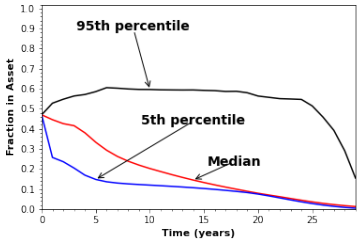
FIGURE 6.1: *Comparison of performance of multiple asset tontine vs two asset tontine vs constant proportion strategy. EW-ES frontiers, computed from the problem (3.33). Note: Tontine scenario in Table 6.1. Two asset tontine scenario uses the same specification as the multi-asset scenario except the retiree has access to only Canadian T-bill and US cap-weighted index. Training data is* 2.56×10^5 *observations of historical bootstrapped data.* $q_{min} = 40, q_{\max} = 80$. $\epsilon = -10^{-6}$. *Units: thousands of Canadian dollars*

	Cvw	USvw	Cew	USew
Cvw	1	0.738299	0.870859	0.716551
USvw	0.738299	1	0.598338	0.852433
Cew	0.870859	0.598338	1	0.718511
USew	0.716551	0.852433	0.718511	1

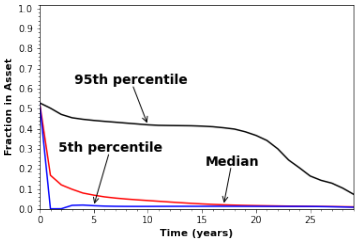
TABLE 6.2: *Comparison of correlation between returns of different Canadian and US indexes. All returns are calculated in real Canadian Dollars. Returns from 1950:11 to 2022:12. Rows and columns indicate the following indexes: Cew: Canadian equal-weighted index; USew: US equal-weighted index; Cvw: Canadian value-weighted index; USvw: US value-weighted index.*



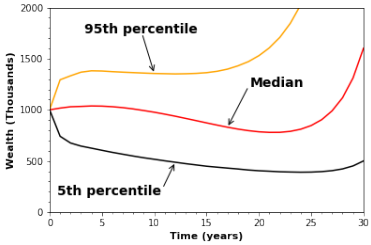
a *Percentiles fraction in 30-day Canadian T-bills, NN Control,* $\epsilon = -10^{-6}$



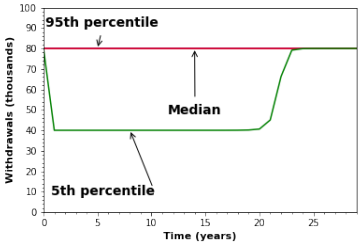
b *Percentiles fraction in Canadian equal-weighted stock index, NN Control,* $\epsilon = -10^{-6}$



c *Percentiles fraction in US value-weighted stock index, NN Control,* $\epsilon = -10^{-6}$

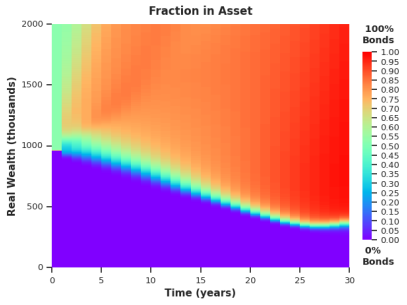


d *Percentiles wealth, NN Control,* $\epsilon = -10^{-6}$

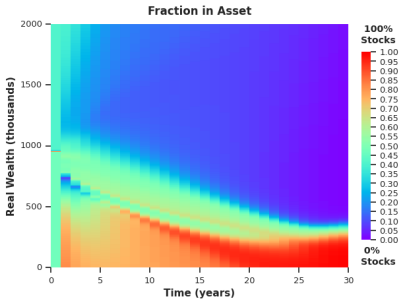


e *Percentiles withdrawals, NN Control,* $\epsilon = -10^{-6}$

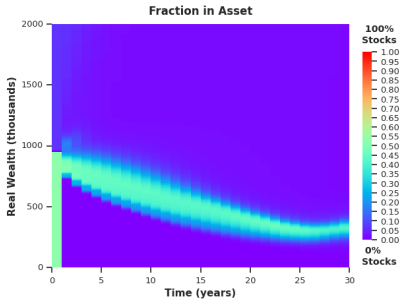
FIGURE 6.2: *Scenario in Table 6.1. NN controls computed from the problem (6.17). No investment in long term bonds, the US equal-weighted index, and the Canadian value-weighted index. NN model trained on* 2.56×10^5 *observations of historical bootstrapped data.* $q_{min} = 40, q_{\max} = 80$, $\kappa = 0.03$. *Units: thousands of dollars.*



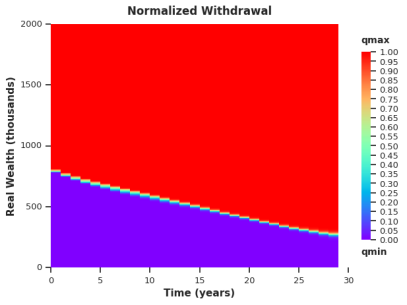
a *Percentiles fraction in 30-day Canadian T-bills, NN Control,* $\epsilon = -10^{-6}$



b *Heatmap of fraction in Canadian equal-weighted stock index, NN Control,* $\epsilon = -10^{-6}$



c *Heatmap of fraction in US value-weighted stock index, NN Control,* $\epsilon = -10^{-6}$



d *Heatmap of normalized withdrawals, NN Control,* $\epsilon = -10^{-6}$

FIGURE 6.3: *Scenario in Table 6.1. NN controls computed from the problem (6.17). NN model trained on* 2.56×10^5 *observations of synthetic data.* $q_{min} = 40, q_{\max} = 80$, $\kappa = 0.05$. *Units: thousands of dollars.*

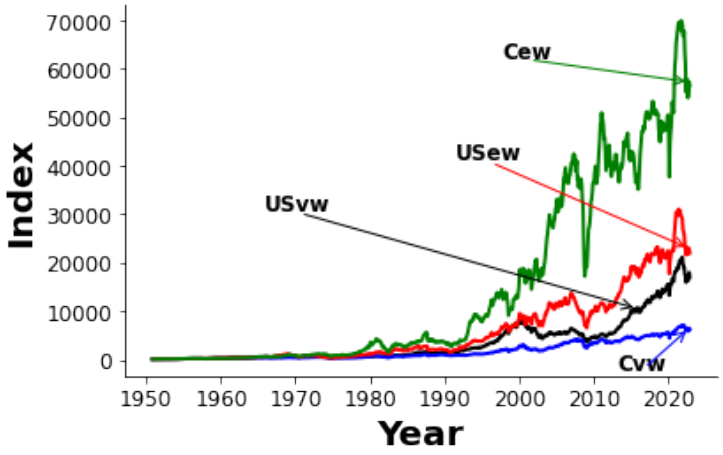


FIGURE 6.4: *Comparison of performance of different Canadian and US indexes. All indexes are calculated from real returns in Canadian Dollars. Base month 1950:11. Labels indicate the following indexes: Cew: Canadian equal-weighted index; USew: US equal-weighted index; Cvw: Canadian value-weighted index; USvw: US value-weighted index.*

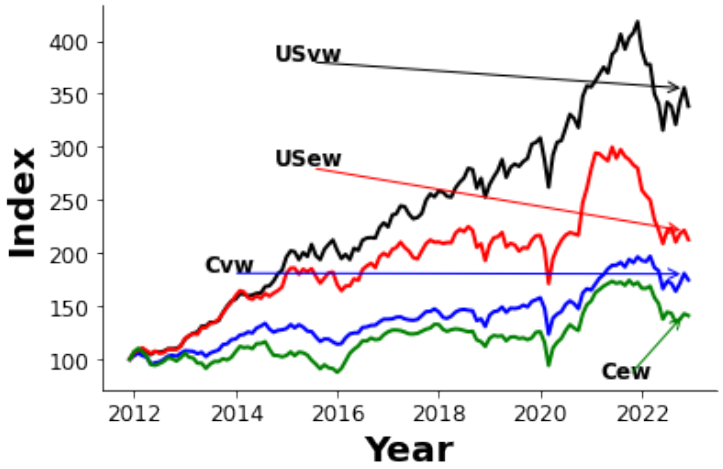


FIGURE 6.5: *Comparison of performance of different Canadian and US indexes. All indexes are calculated from real returns in Canadian Dollars. Base month 2011:12. Labels indicate the following indexes: Cew: Canadian equal-weighted index; USew: US equal-weighted index; Cvw: Canadian value-weighted index; USvw: US value-weighted index.*

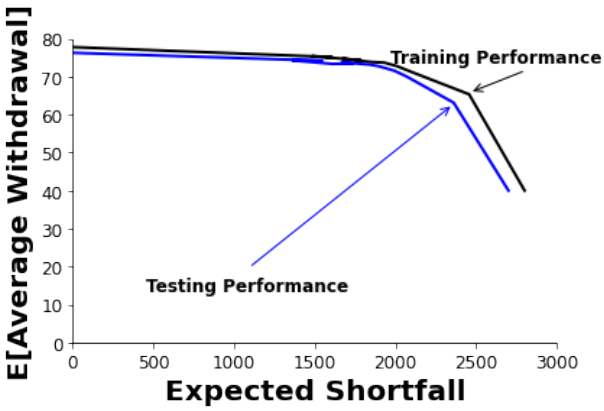


FIGURE 6.6: *Comparison of performance on training vs test data set. Note: Scenario in Table 6.1. Training data is* 2.56×10^5 *observations of historical bootstrapped data from 1950:10 to 1985:12. Test data is* 2.56×10^5 *observations of historical bootstrapped data from 1986:01 to 2022:12.* $q_{min} = 40, q_{\max} = 80$. $\epsilon = -10^{-6}$. *Units: thousands of Canadian dollars*

Chapter 7

Conclusion

In this thesis, we proposed a novel neural network (NN) architecture to efficiently and accurately compute the optimal decumulation strategy for retirees with DC pension plans. The stochastic-constrained optimal control problem is solved based on a single standard unconstrained optimization, without using dynamic programming.

We began by highlighting the increasing prevalence of DC pension plans over traditional DB pension plans, and outlining the critical decumulation problem that faces DC plan investors. There is extensive literature on devising strategies for this problem. In particular, we examine a Hamilton-Jacobi-Bellman (HJB) Partial Differential Equation (PDE) based approach that can be shown to converge to an optimal solution for a dynamic withdrawal/allocation strategy. This provides an attractive balance of risk management and withdrawal efficiency for retirees. In this paper, we seek to build upon this approach by developing a new, more versatile framework using NNs to solve the decumulation problem.

We conduct computational investigations to demonstrate the accuracy and robustness of the proposed NN solution in multiple scenarios, utilizing the unique opportunity to compare NN solutions with the HJB results as a ground truth. Of particular noteworthiness is that the continuous function approximation from the NN framework is able to approximate a bang-bang control with high accuracy. We extend our experiments to establish the robustness of our approach, testing the NN control's performance on both synthetic and historical data sets, as well as on modifications to the investment scenario.

We demonstrate that the proposed NN framework produced solution accurately approximates the ground truth solution. We also note the following advantages of the proposed NN framework:

(i) The NN method is data-driven, and does not require postulating and calibrating a parametric model for market processes.

(ii) The NN method directly estimates the low dimensional control by solving a single unconstrained optimization problem, avoiding the problems associated with dynamic programming methods, which require estimating high dimensional conditional expectations (see Van Staden et al. (2023)).

(iii) The NN formulation maintains its simple structure (discussed in Section 3.3.2), immediately extendable to problems with more frequent rebalancing and/or withdrawal events. In fact, the problem presented in (3.33) requires each control NN to have only two hidden layers for 30 rebalancing and withdrawal periods.

(iv) The approximated control maintains continuity in time and/or space, provided it exists, or otherwise provides a smooth approximation. Continuity of the allocation control p is an important practical consideration for any investment policy.

Due to the ill-posedness of the stochastic optimal control problem in the region of high wealth near the end of the decumulation horizon, we observe that the NN allocation can appear to be very different from the HJB PDE solution. We note, however, that both strategies yield indistinguishable performance when assessed with the expected withdrawal and ES reward-risk criteria. In other words, these differences hardly affect the objective function value, a weighted reward and risk value. In the region of high wealth level near the end of time horizon, the retiree is free to choose whether to invest 100% in stocks or 100% in bonds, since this has a negligible effect on the objective function value (or reward-risk consideration).[1]

We further demonstrate the flexibility of the NN framework to incorporate more state variables without a change in the NN structure. However, when the NN structure does have to change, for example when a higher number of assets are available to the retiree, we notice that the increase in problem complexity is lower compared to the increase in complexity of methods involving DP.

To conclude, the advantages of the NN framework make it a more versatile method, compared to the solution of the HJB PDE. We expect that the NN approach can handle problems of higher complexity, e.g., involving taxes. We leave the extension of this methodology to future work.

[1]This can be termed the *Warren Buffet* effect. Buffet is the fifth richest human being in the world. He is 92 years old. Buffet can choose any allocation strategy, and will never run out of cash.

References

A. Anarkulova, S. Cederburg, and M. S. O'Doherty. Stocks for the long run? evidence from a broad sample of developed markets. *Journal of Financial Economics*, 143(1): 409–433, 2022.

W. Bengen. Determining withdrawal rates using historical data. *Journal of Financial Planning*, 7:171–180, 1994.

T. Bernhardt and C. Donnelly. Pension decumulation strategies: A state of the art report. Technical Report, Risk Insight Lab, Heriot Watt University, 2018.

T. Bjork and A. Murgoci. A general theory of Markovian time inconsistent stochastic control problems. SSRN 1694759, 2010.

T. Bjork and A. Murgoci. A theory of Markovian time inconsisent stochastic control in discrete time. *Finance and Stochastics*, 18:545–592, 2014.

E. B Boukherouaa, K. AlAjmi, J. Deodoro, A. Farias, and R. Ravikumar. Powering the digital economy: Opportunities and risks of artificial intelligence in finance. *IMF Departmental Papers*, 2021(024):A001, 2021. doi:10.5089/9781589063952.087.A001. URL https://www.elibrary.imf.org/view/journals/087/2021/024/article-A001-en.xml.

H. Buehler, L. Gonon, J. Teichmann, and B. Wood. Deep hedging. *Quantitative Finance*, 19(8):1271–1291, 2019.

P. Cogneau and V. Zakamouline. Block bootstrap methods and the choice of stocks for the long run. *Quantitative Finance*, 13(9):1443–1457, 2013.

R. Cont and C. Mancini. Nonparametric tests for pathwise properties of semimartingales. *Bernoulli*, 17:781–813, 2011.

M. Dahl. Stochastic mortality in life insurance: market reserves and mortality-linked insurance contracts. *Insurance: mathematics and economics*, 35(1):113–136, 2004.

D-M. Dang and P. A. Forsyth. Continuous time mean-variance optimal portfolio allocation under jump diffusion: a numerical impulse control approach. *Numerical Methods for Partial Differential Equations*, 30:664–698, 2014.

D-M. Dang and P. A. Forsyth. Better than pre-commitment mean-variance portfolio allocation strategies: a semi-self-financing Hamilton-Jacobi-Bellman equation approach. *European Journal of Operational Research*, 250:827–841, 2016.

H. Dichtl, W. Drobetz, and M. Wambach. Testing rebalancing strategies for stock-bond portfolios across different asset allocations. *Applied Economics*, 48(9):772–788, 2016.

P. A. Forsyth. Multi-period mean CVAR asset allocation: Is it advantageous to be time consistent? *SIAM Journal on Financial Mathematics*, 11:2:358–384, 2020.

P. A. Forsyth. A stochastic control approach to defined contribution plan decumulation: "the nastiest, hardest problem in finance". *North American Actuarial Journal*, 26(2): 227–251, 2022.

P. A. Forsyth and K. R. Vetzal. Dynamic mean variance asset allocation: Tests for robustness. *International Journal of Financial Engineering*, 4:1750021:1–1750021:37, 2017. DOI: 10.1142/S2424786317500219.

P. A. Forsyth and K. R. Vetzal. Optimal asset allocation for retirement savings: deterministic vs. time consistent adaptive strategies. *Applied Mathematical Finance*, 26:1:1–37, 2019.

P. A Forsyth, K. R. Vetzal, and G. Westmacott. Optimal performance of a tontine overlay subject to withdrawal constraints. *arXiv 2211.10509*, 2022.

P.A. Forsyth and G. Labahn. $\epsilon-$Monotone Fourier methods for optimal stochastic control in finance. *Journal of Computational Finance*, 22:4:25–71, 2019.

R. K. Fullmer and M. J. Sabin. Individual tontine accounts. *Fullmer, RK, & Sabin, MJ (2019). Individual Tontine Accounts. Journal of Accounting and Finance*, 19(8), 2018.

J. Han and E. Weinan. Deep learning approximation for stochastic control problems. *CoRR*, abs/1611.07422, 2016. URL http://arxiv.org/abs/1611.07422.

S. Homer and R. Sylla. *A History of Interest Rates*. Wiley, New York, 2005.

C. m. Huré, H. Pham, A. Bachouch, and N. Langrené. Deep neural networks algorithms for stochastic control problems on finite horizon: Convergence analysis. *SIAM Journal on Numerical Analysis*, 59(1):525–557, 2021. doi:10.1137/20m1316640. URL https://doi.org/10.1137%2F20m1316640.

V. Ismailov. A three layer neural network can represent any multivariate function, 2022.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv*, page arXiv:1412.6980, 2014.

S. G. Kou. A jump-diffusion model for option pricing. *Management Science*, 48:1086–1101, 2002.

S. G. Kou and H. Wang. Option pricing under a double exponential jump diffusion model. *Management Science*, 50:1178–1192, 2004.

M. Laurière, O. Pironneau, et al. Performance of a markovian neural network versus dynamic programming on a fishing control problem. *arXiv preprint arXiv:2109.06856*, 2021.

Y. Lin, R. MacMinn, and R. Tian. De-risking defined benefit plans. *Insurance: Mathematics and Economics*, 63:52–65, 2015.

B.-J. MacDonald, B. Jones, R. J. Morrison, R. L. Brown, and M. Hardy. Research and reality: A literature review on drawing down retirement financial savings. *North American Actuarial Journal*, 17:181–215, 2013.

R. MacMinn, P. Brockett, J. Wang, Y. Lin, and R. Tian. The securitization of longevity risk and its implications for retirement security. In O. S. Mitchell, R. Maurer, and P. Brett Hammond, editors, *Recreating Sustainable Retirement*, pages 134–160. Oxford University Press, Oxford, 2014.

C. Mancini. Non-parametric threshold estimation models with stochastic diffusion coefficient and jumps. *Scandinavian Journal of Statistics*, 36:270–296, 2009.

R. Marler and Jasbir Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26:369–395, 04 2004. doi:10.1007/s00158-003-0368-6.

C. Ni, Y. Li, P. Forsyth, and R. Carroll. Optimal asset allocation for outperforming a stochastic benchmark target. *Quantitative Finance*, 22(9):1595–1626, 2022.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

A. Patton, D. Politis, and H. White. Correction to: automatic block-length selection for the dependent bootstrap. *Econometric Reviews*, 28:372–375, 2009.

W. D. Pfau. An overview of retirement income planning. *Journal of Financial Counseling and Planning*, 29:1:114:120, 2018.

S. Pfeiffer, J. R. Salter, and H. E. Evensky. Increasing the sustainable withdrawal rate using the standby reverse mortgage. *Journal of Financial Planning*, 26:12:55–62, 2013.

D. Politis and J. Romano. The stationary bootstrap. *Journal of the American Statistical Association*, 89:1303–1313, 1994.

D. Politis and H. White. Automatic block-length selection for the dependent bootstrap. *Econometric Reviews*, 23:53–70, 2004.

W. B. Powell. *From Reinforcement Learning to Optimal Control: A Unified Framework for Sequential Decisions*, pages 29–74. Springer International Publishing, Cham, 2021. ISBN 978-3-030-60990-0. doi:10.1007/978-3-030-60990-0_3. URL https://doi.org/10.1007/978-3-030-60990-0_3.

B. Ritholz. Tackling the 'nastiest, hardest problem in finance'. www.bloomberg.com/view/articles/2017-06-05/tackling-the-nastiest-hardest-problem-in-finance, 2017.

R. T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–42, 2000.

S. Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

J. Scott, W. Sharpe, and J. Watson. The 4% rule – at what price? *Journal of Investment Management*, 7(3):31–48, 2009.

L. Scott and S. Cavaglia. A wealth management perspective on factor premia and the value of downside protection. *Journal of Portfolio Management*, 43(3):33–41, 2017.

H. M. Shefrin and R. H. Thaler. The behavioral life-cycle hypothesis. *Economic Inquiry*, 26(4):609–643, 1988. doi:https://doi.org/10.1111/j.1465-7295.1988.tb01520.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1465-7295.1988.tb01520.x.

J. Simonian and A. Martirosyan. Sharpe parity redux. *The Journal of Portfolio Management*, 48(4):183–193, 2022.

L. Spierdijk and Z. Umar. Stocks, bonds, t-bills and inflation hedging: From great moderation to great recession. *Journal of Economics and Business*, 79:1–37, 2015.

M. Strub, D. Li, and X. Cui. An enhanced mean-variance framework for robo-advising applications. SSRN 3302111, 2019.

C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu. A survey on deep transfer learning. *arXiv 1808.01974*, 2018.

P. Tankov and R. Cont. *Financial Modelling with Jump Processes*. Chapman and Hall/CRC, New York, 2009.

K. H. Tsang and H. Y. Wong. Deep-learning solution to portfolio selection with serially-dependent returns. *SSRN*, 10.2139, 2020. URL https://ssrn.com/abstract=3396246.

U.S. Bureau of Labor Statistics. Employee benefits survey: Latest numbers, 2022. https://www.bls.gov/ebs/latest-numbers.htm.

P. Van Staden, P. Forsyth, and Y. Li. Beating a benchmark: dynamic programming may not be the right numerical approach. *SIAM Journal on Financial Mathematics*, 14:2: 407–451, 2023.

E. Vigna. On efficiency of mean-variance based portfolio selection in defined contribution pension schemes. *Quantitative Finance*, 14:237–258, 2014.

E. Vigna. Tail optimality and preferences consistency for intertemporal optimization problems. Working paper no. 502 , Collegio Carlo Alberto, Università Degli Studi di Torino, 2017.

M. B. Waring and L. B. Siegel. The only spending rule article you will ever need. *Financial Analysts Journal*, 71(1):91–107, 2015.

R. Williams and C. Kawashima. Beyond the 4% rule: How much can you spend in retirement? *Charles Schwab Brokerage*, 2023. URL https://www.schwab.com/learn/story/beyond-4-rule-how-much-can-you-spend-retirement. https://www.schwab.com/learn/story/beyond-4-rule-how-much-can-you-spend-retirement.

APPENDICES

Appendix A

Induced Time Consistent Policy

In this chapter of the appendix, we review the concept of time consistency and relate its relevance to the $PCEE_{t_0}(\kappa)$ problem, (3.33).

Consider the optimal control $\mathcal{P}^*$ for problem (3.33),

$$(\mathcal{P}^*)^{t_0}(X(t_i^-), t_i) \ ; \ i = 0, \ldots, M \ . \tag{A.1}$$

Equation (A.1) can be interpreted as the optimal control for any time $t_i \geq t_0$, as a function of the state variables $X(t)$, as computed at t_0.

Now consider if we were to solve the problem (3.33) starting at a later time $t_k, k > 0$. This optimal control starting at t_k is denoted by:

$$(\mathcal{P}^*)^{t_k}(X(t_i^-), t_i) \ ; \ i = k, \ldots, M\} \ . \tag{A.2}$$

In general, the solution of (3.33) computed at t_k is not equivalent to the solution computed t_0:

$$(\mathcal{P}^*)^{t_k}(X(t_i^-), t_i) \neq (\mathcal{P}^*)^{t_0}(X(t_i^-), t_i) \ ; \ i \geq k > 0. \tag{A.3}$$

This non-equivalence makes problem (3.33) *time inconsistent*, implying that the investor will be motivated to diverge from the control determined at time t_0 at later times. This type of control is considered a *pre-commitment* control since the investor would need to commit to following the strategy at all times following t_0. Some authors describe pre-commitment controls as non-implementable because of the incentive to diverge from the initial control.

In our case, however, the pre-commitment control from (3.33) can be shown to be identical to the time consistent control for an alternative version of the objective function. By holding W' fixed at the optimal value (at time zero), we can define the time consistent equivalent problem (TCEQ). We define the optimal value of W' as [1]

$$\mathcal{W}^*(s,b) \quad = \quad \arg\max_{W'} \sup_{\mathcal{P}_0 \in \mathcal{A}} \left\{ E_{\mathcal{P}_0}^{X_0^-, t_0^-} \left[\sum_{i=0}^{M} q_i \; + \; \kappa \left(W' + \frac{1}{\alpha} \min(W_T - W', 0) \right) \middle| X(t_0^-) = (s,b) \right] \right\} . \tag{A.4}$$

With a given initial wealth of W_0^-, this gives the following result from Forsyth (2020): [Pre-commitment strategy equivalence to a time consistent policy for an alternative objective function]

The pre-commitment EW-ES strategy found by solving $J\left(s,b,t_0^-\right)$ from (3.33), with fixed $W' = \mathcal{W}^*$ from Equation (A.4), is identical to the time consistent strategy for the equivalent problem TCEQ (which has fixed $\mathcal{W}^*(0,W_0^-)$), with the following value function:

$$\begin{aligned} &\left(TCEQ_{t_n}\left(\kappa/\alpha\right)\right): \\ &\tilde{J}\left(s,b,t_n^-\right) \quad = \quad \sup_{\mathcal{P}_n \in \mathcal{A}} \left\{ E_{\mathcal{P}_n}^{X_n^-, t_n^-} \left[\sum_{i=n}^{M} q_i \; + \; \frac{\kappa}{\alpha} \min(W_T - \mathcal{W}^*(0, W_0^-), 0) \middle| X(t_n^-) = (s,b) \right] \right\}. \end{aligned} \tag{A.5}$$

Proof. This follows similar steps as in Forsyth (2020), proof of Proposition (6.2). □

With fixed W', $TCEQ_{t_n}(\kappa/\alpha)$ uses a target-based shortfall as its measure of risk, which is trivially time consistent. W' has the convenient interpretation of a disaster level of final wealth, as specified at time zero. Since the optimal controls for $PCEE_{t_0}(\kappa)$ and $TCEQ_{t_n}(\kappa/\alpha)$ are identical, we regard $TCEQ_{t_n}(\kappa/\alpha)$ as the induced time consistent strategy Strub et al. (2019) for problem EW-ES. The retiree has no motivation to diverge from the induced time consistent strategy, determined at time zero. Hence this policy is implementable.

For more detailed analysis concerning the subtle distinctions involved in pre-commitment, time consistent, and induced time consistent strategies, please consult Bjork and Murgoci (2010; 2014); Vigna (2014; 2017); Strub et al. (2019); Forsyth (2020); **?**.

[1] The arg max is well defined since $\sup_{\mathcal{P}}\{\cdot\}$ is a continuous function of W'.

Appendix B

PIDE Between Rebalancing Times

Applying Ito's Lemma for jump processes Tankov and Cont (2009), using Equations (3.13) and (3.14) in Equation (3.41) gives

$$V_t + \frac{(\sigma^s)^2 s^2}{2} V_{ss} + (\mu^s - \lambda_\xi^s \gamma_\xi^s) s V_s + \lambda_\xi^s \int_{-\infty}^{+\infty} V(e^x s, b, t) f^s(x)\, dx + \frac{(\sigma^b)^2 b^2}{2} V_{bb}$$
$$+ (\mu^b + \mu_c^b \mathbf{1}_{\{b<0\}} - \lambda_\xi^b \gamma_\xi^b) b V_b + \lambda_\xi^b \int_{-\infty}^{+\infty} V(s, e^x b, t) f^b(x)\, dx - (\lambda_\xi^s + \lambda_\xi^b) V + \rho_{sb} \sigma^s \sigma^b s b V_{sb} = 0 \ ,$$
$$s \geq 0 \ . \tag{B.1}$$

where the density functions $f^s(x), f^b(x)$ are as given in equation (3.11).

Appendix C

Computational Details: Hamilton-Jacobi-Bellman (HJB) PDE Framework

For a detailed description of the numerical algorithm used to solve the HJB equation framework described in Section 3.2, we refer the reader to Forsyth (2022). We summarize the method here.

First, we solve the auxiliary problem (3.35), with fixed values of W', κ and α. The state space in $s > 0$ and $b > 0$ is discretized using evenly spaced nodes in log space to create a grid to represent cases. A separate grid is created in a similar fashion to represent cases where wealth is negative. The Fourier methods discussed in Forsyth and Labahn (2019) are used to solve the PIDE representing market dynamics between rebalancing times. Both controls for withdrawal and allocation are discretized using equally spaced grids. The optimization problem (3.37) is solved first for the allocation control by exhaustive search, storing the optimal for each discretized wealth node. The withdrawal control in (3.38) can then be solved in a similar fashion, using the previously stored allocation control to evaluate the right-hand side of (3.38). Linear interpolation is used where necessary. The stored controls are used to advance the solution in (3.40).

Since the numerical method just described assumes a constant W', an outer optimization step to find the optimal W' (candidate Value-at-Risk) is necessary. Given an approximate solution to (3.35) at $t = 0$, the full solution to $PCEE_{t_0}(\kappa)$ (3.33) is determined using Equation (3.42). A coarse grid is used at first for an exhaustive search. The coarse grid solution is used as an initial guess for a univariate optimization technique on finer grids.

Appendix D

Computational Details: NN Framework

D.1 NN Optimization

The NN framework, as described in Section 3.3 and illustrated in Figure 3.1, was implemented using the PyTorch library Paszke et al. (2019). The withdrawal network $\hat{q}$, and allocation network $\hat{p}$ were both implemented with 2 hidden layers of 10 nodes each, with biases for problems with two assets. For the multi-asset case, we used 2 hidden layers with 14 nodes each, with biases. Stochastic Gradient Descent Ruder (2016) was used in conjunction with the Adaptive Momentum optimization algorithm to train the NN framework Kingma and Ba (2014). The NN parameters and auxiliary training parameter W' were trained with different initial learning rates. The same decay parameters and learning rate schedule were used. Weight decay (ℓ_2 penalty) was also employed to make training more stable. The training loop utilizes the auto-differentiation capabilities of the PyTorch library. Hyper-parameters used for NN training for experiments in Chapters 4 and 5 are given in Table D.1. For the experiment in **??**, the hyper-parameters can be found in Table D.2

The training loop tracks the minimum loss function value as training progresses and selects the model that had given the optimal loss function value based on the entire training dataset by the end of the specified number of training epochs.

D.2 Transfer learning between different κ points

For high values of κ, the objective function is weighted more towards optimizing ES (lower risk). In these cases, optimal controls are more difficult to compute. This is because the ES measure used (CVAR) is only affected by the sample paths below the 5^{th} percentile of terminal wealth, which are quite sparse. To overcome these training difficulties, we employ transfer learning Tan et al. (2018) to improve training for the more difficult points on the efficient frontier. We begin training the model for the lowest κ from a random initialization ('cold-start'), and then initialize the models for each increasing κ with the model for the previous κ. Through numerical experiments, we found this method made training far more stable and less likely to terminate in local minima for higher values of κ.

D.3 Running minimum tracking

The training loop tracks the minimum loss function value as training progresses and selects the model that had given the optimal loss function value based on the entire training dataset by the end of the specified number of training epochs.

D.4 Standardization

To improve learning for the neural network, we normalize the input wealth using means and standard deviations of wealth samples from a reference strategy. We use the constant withdrawal and allocation strategy defined in Forsyth (2022) as the reference strategy with 2.56×10^5 simulated paths. Let W_t^b denote the wealth vector at time t based on simulations. Then $\bar{W}_t^b$ and $\sigma(W_t^b)$ denote the associated average wealth and standard deviation. Then we normalize the feature input to the neural network in the following way:

$$\tilde{W}_t = \frac{W_t - \bar{W}_t^b}{\sigma(W_t^b)}$$

For the purpose of training the neural network, the values $\bar{W}_t^b$ and $\sigma(W_t^b)$ are just constants, and we can use any reasonable values. This input feature normalization is done for both withdrawal and allocation NNs.

In Section 4.3, we show in out-of-sample and out-of-distribution tests that $\bar{W}_t^b$ and $\sigma(W_t^b)$ do not need to be related to the testing data as long as these are reasonable values.

NN framework hyper-parameter	Value
Hidden layers per network	2
# of nodes per hidden layer	10
Nodes have biases	True
# of iterations (#itn)	50,000
SGD mini-batch size	1,000
# of training paths	2.56×10^5
Optimizer	Adaptive Momentum
Initial Adam learning rate for $(\boldsymbol{\theta}_q, \boldsymbol{\theta}_p)$	0.05
Initial Adam learning rate for W'	0.04
Adam learning rate decay schedule	$[0.70 \times \#\text{itn}, 0.97 \times \#\text{itn}]$, $\gamma = 0.20$
Adam β_1	0.9
Adam β_2	0.998
Adam weight decay (ℓ_2 Penalty)	0.0001
Transfer Learning between κ points	True
Take running minimum as result	True

TABLE D.1: *Hyper-parameters used in training the NN framework for numerical experiments presented in Chapters 4 and 5.*

In Section 3.3, when referring to W as part of the input to the NN functions $\hat{q}$ and $\hat{p}$, we use the standardized $\tilde{W}$ for computation.

NN framework hyper-parameter	Value
Hidden layers per network	2
# of nodes per hidden layer	14
Nodes have biases	True
# of iterations (#itn)	50,000
SGD mini-batch size	1,000
# of training paths	2.56×10^5
Optimizer	Adaptive Momentum
Initial Adam learning rate for $(\boldsymbol{\theta}_q, \boldsymbol{\theta}_p)$	0.05
Initial Adam learning rate for W'	0.05
Adam learning rate decay schedule	$[0.70 \times \#\text{itn}, 0.97 \times \#\text{itn}]$, $\gamma = 0.20$
Adam β_1	0.9
Adam β_2	0.998
Adam weight decay (ℓ_2 Penalty)	0.0001
Transfer Learning between κ points	True
Take running minimum as result	True

TABLE D.2: *Hyper-parameters used in training the NN framework for numerical experiments presented in Chapter* **??**.

Appendix E

Optimal expected block sizes for block resampling

Tables E.1 and E.2 shows our estimates of the optimal block size using the algorithm in Politis and White (2004); Patton et al. (2009) using market data described in Sections 4.1 and 5.2.

Optimal expected block size for bootstrap resampling historical data

Data	Optimal expected block size $\hat{b}$ (months)
CRSP US 10-year treasury	4.2
CPI adjusted CRSP US Total Market Index	3.1

TABLE E.1: *Optimal expected blocksize* $\hat{b} = 1/v$*, from Patton et al. (2009). Range of historical data is between 1926:1 and 2019:12. The blocksize is a draw from a geometric distribution with* $Pr(b = k) = (1 - v)^{k-1}v$.

Optimal expected block size for bootstrap resampling historical data

Data	Optimal expected block size $\hat{b}$ (months)
Real 30-day T-bill index	50.6
Real CRSP Value-weighted index	3.42

TABLE E.2: *Optimal expected blocksize* $\hat{b} = 1/v$*, from Patton et al. (2009). Range of historical data is between 1926:1 and 2020:12. The blocksize is a draw from a geometric distribution with* $Pr(b = k) = (1 - v)^{k-1}v$.

Appendix F

Convergence Test: HJB Equation

F.1 Simple decumulation problem

Table F.1 shows a detailed convergence test for a single point on the (EW, ES) frontier, using the PIDE method. The controls are computed using the HJB PDE, and stored, which are then used in MC simulations. These results are used to verify the PDE solution, and also generate various statistics of interest.

HJB Method in §3.2					MC Simulation	
Grid $n_x \times n_b$	ES (5%)	$E[\sum_i q_i]/(M+1)$	Value Function	W'	ES (5%)	$E[\sum_i q_i]/(M+1)$
512×512	-51.302	52.056	1.562430e+3	50.10	-45.936	52.07
1024×1024	-46.239	52.049	1.567299e+3	52.47	-45.102	52.05
2048×2048	-42.594	51.976	1.568671e+3	58.00	-42.623	51.97
4096×4096	-40.879	51.932	1.569025e+3	61.08	-41.250	51.93

TABLE F.1: *HJB convergence analysis. CPI adjusted CRSP US Total Market Index and CRSP US 10-year treasury. Investment setup up in Table 4.2. Calibrated jump model in Table 4.1.* 2.56×10^6 *MC simulations.* $\kappa = 1.0, \alpha = .05$. *Discretization grid in Section 3.2.* n_x*: # of nodes in* $\log s$. n_b *: # of nodes* $\log b$. *Monetary units:* USD *in thousands.* $(M+1)$*: # of withdrawals.* M *: # of rebalancing dates. Minimum withdrawal: 35. Maximum withdrawal: 60. HJB method in Section 3.2.*

F.2 Tontine

Table F.2 shows a detailed convergence test for a single point on the (EW, ES) frontier, using the PIDE method (Forsyth et al., 2022). The controls are computed using the HJB PDE, and stored, which are then used in MC simulations. These results are used to verify the PDE solution, and also generate various statistics of interest.

HJB Method in §3.2					MC Simulation	
Grid $n_x \times n_b$	ES (5%)	$E[\sum_i q_i]/(M+1)$	Value Function	W'	ES (5%)	$E[\sum_i q_i]/M$
512×512	108.13	67.99	2059.60	123.26	68.04	
1024×1024	158.88	67.79	2063.19	164.45	67.81	
2048×2048	201.88	67.56	2064.27	203.87	67.56	
4096×4096	206.56	67.54	2064.54	207.70	67.54	

TABLE F.2: *HJB convergence analysis. Real stock index: deflated real capitalization weighted CRSP, real bond index: deflated 30-day T-bills. Investment setup up in Table 5.2. Calibrated jump model in Table 5.1.* 2.56×10^6 *MC simulations.* $\kappa = 0.185, \alpha = .05$. *Discretization grid in Section 3.2.* n_x*: # of nodes in* $\log s$. n_b *: # of nodes* $\log b$. *Monetary units:* USD *in thousands.* M*: # of withdrawals.* M *: # of rebalancing dates.* $q_{\min} = 40.0$. $q_{\max} = 80$. *HJB method in Section 3.2.*

Appendix G

Detailed efficient frontier comparisons

G.1 Simple decumulation problem

Table G.1 shows the detailed efficient frontier, computed using the HJB equation method, using the 2048×2048 grid. Table G.2 shows the efficient frontier computed from the NN framework. This should be compared to Table G.1. Table G.3 compares the objective function values, at various points on the efficient frontier, for the HJB and NN frameworks.

G.2 Tontine

Table G.4 shows the detailed efficient frontier, computed using the HJB equation method, using the 2048×2048 grid by Forsyth et al. (2022). Table G.5 shows the efficient frontier computed from the NN framework. This should be compared to Table G.1. Table G.6 compares the objective function values, at various points on the efficient frontier, for the HJB and NN frameworks.

Efficient Frontier Details: HJB Framework

κ	ES (5%)	$E[\sum_i q_i]/(M+1)$	$Median[W_T]$
0.05	-596.00	57.14	124.36
0.2	-334.29	56.17	92.99
0.5	-148.99	54.25	111.20
1.0	-42.62	51.97	227.84
1.5	-8.05	50.63	298.20
3.0	17.42	48.95	380.36
5.0	24.09	48.12	414.60
50.0	30.60	45.70	519.03
∞	31.00	35.00	1003.47

TABLE G.1: *Details of training performance efficient frontier in Figure 4.3 for HJB optimal strategies based on calibrated jump model. Investment setup in Table 4.2. CPI adjusted CRSP US Total Market Index and CRSP US 10-year treasury. Jump model parameters from Table 4.1. Monetary units:* USD *in thousands.* 2.56×10^6 *MC simulations. Control is computed using HJB method in §3.2 with (*2048×2048 *grid) stored, subsequently used in MC simulations. Minimum withdrawal: 35. Maximum withdrawal: 60.* $(M+1)$: *# of withdrawals.* M : *# of rebalancing dates.* $\epsilon = 10^{-6}$.

Detailed Efficient Frontier: NN Framework

κ	ES (5%)	$E[\sum_i q_i]/(M+1)$	$Median[W_T]$
0.05	-599.81	57.15	106.23
0.2	-333.01	56.14	78.59
0.5	-160.14	54.40	105.05
1	-43.02	51.95	227.79
1.5	-8.57	50.62	302.17
3	16.01	48.99	374.43
5	23.20	48.13	425.13
50	29.88	45.72	493.41
∞	29.90	35.00	947.60

TABLE G.2: *Details of training performance efficient frontier in Figure 4.3 for NN optimal strategies based on calibrated jump model. Investment setup in Table 4.2. CPI adjusted CRSP US Total Market Index and CRSP US 10-year treasury. Jump model parameters from Table 4.1. Monetary units:* USD *in thousands.* 2.56×10^5 *MC simulations. Control is computed using NN in Section 3.3. Minimum withdrawal: 35. Maximum withdrawal: 60.* $(M+1)$: *# of withdrawals.* M : *# of rebalancing dates.* $\epsilon = 10^{-6}$.

Objective Function Value Comparison: HJB Framework vs. NN Framework

κ	HJB equation	NN	% difference
0.05	1741.54	1741.71	0.01%
0.2	1674.41	1673.81	-0.04%
0.5	1607.26	1606.44	-0.05%
1	1568.45	1567.34	-0.07%
1.5	1557.46	1556.22	-0.08%
3	1569.71	1566.86	-0.18%
5	1612.16	1607.86	-0.27%
50	2946.70	2911.10	-1.21%

TABLE G.3: *Comparison in objective function values between HJB equation and NN framework model for various κ values. Objective function values for both frameworks computed according to $PCEE_{t0}(\kappa)$ (higher is better). Investment setup in Table 4.2. CPI adjusted CRSP US Total Market Index and CRSP US 10-year treasury. Jump model parameters from Table 4.1. HJB solution statistics based on 2.56×10^6 MC simulations. HJB control is computed as in Section 3.2, (2048×2048 grid) stored, and then used in the MC simulations. NN Training performance statistics based on 2.56×10^5 MC simulations. Control is computed using the NN framework in Section 3.3. Minimum withdrawal: 35. Maximum withdrawal: 60. $(M+1)$: # of withdrawals. M : # of rebalancing dates. $\epsilon = 10^{-6}$.*

Efficient Frontier Details: HJB Framework

κ	$E[\sum_i i]/T$	ES (5%)	$Median[W_T]$	W^*
0.15	70.06	-309.569	189.48	.490
0.170	70.04	-270.13	185.19	.489
.180	68.51	46.77	599.42	385.28
.185	67.56	203.87	820.65	585.97
.20	66.41	384.76	1058.40	802.40
0.25	63.85	732.34	1517.04	1220.33
0.30	62.22	912.29	1754.40	1439.83
0.50	58.48	1209.40	2120.59	1802.19
1.0	54.81	1372.46	2327.42	2021.22
10.0	48.96	1457.52	2484.58	2151.79
∞	40	1460.76	2885.85	2173.04

TABLE G.4: *EW-ES synthetic market results for optimal strategies, assuming the scenario given in Table 5.2. Tontine gains assumed. Stock index: real capitalization weighted CRSP stocks; bond index: real 30-day T-bills. Parameters from Table 5.1. Units: thousands of dollars. Statistics based on* 2.56×10^6 *Monte Carlo simulation runs. Control is computed using the Algorithm in Section 3.2, stored, and then used in the Monte Carlo simulations.* $q_{\min} = 40$, $q_{\max} = 80$ *(annually).* $T = 30$ *years,* $\epsilon = -10^{-4}$.

Efficient Frontier Details: NN Framework

κ	$E[\sum_i i]/T$	ES (5%)
0.15	70.87	-354.77
0.170	70.46	-269.38
.180	68.61	26.78
.185	67.35	249.24
.20	66.40	393.55
0.25	64.03	723.25
0.30	62.50	856.95
0.50	58.73	1210.28
1.0	54.81	1380.55
10.0	48.24	1476.72
∞	40	1484.25

TABLE G.5: *EW-ES synthetic market results for optimal strategies, assuming the scenario given in Table 5.2. Tontine gains assumed. Stock index: real capitalization weighted CRSP stocks; bond index: real 30-day T-bills. Parameters from Table 5.1. Units: thousands of dollars. Statistics based on* 2.56×10^5 *Monte Carlo simulation runs. Control is computed using the Algorithm in Section 3.3 on simulated data.* $q_{\min} = 40$, $q_{\max} = 80$ *(annually).* $T = 30$ *years,* $\epsilon = -10^{-6}$.

Objective Function Value Comparison: HJB Framework vs. NN Framework

κ	HJB equation	NN	% difference
0.15	2055.36	2072.8845	0.85%
0.17	2055.27	2068.0054	0.62%
0.18	2063.71	2063.1204	-0.03%
0.185	2064.51	2066.6094	0.10%
0.2	2069.25	2070.71	0.07%
0.25	2098.58	2101.5925	0.14%
0.3	2140.28	2132.085	-0.38%
0.5	2359.10	2367.04	0.34%
1	3016.76	3024.85	0.27%
10	16044	16214.4	1.06%

TABLE G.6: *Comparison in objective function values between HJB equation and NN framework model for various κ values. Objective function values for both frameworks are computed according to $PCEE_{t0}(\kappa)$ (higher is better). Investment setup in Table 5.2. CPI adjusted CRSP US Total Market Index and CRSP US 10-year treasury. Jump model parameters from Table 5.1. HJB solution statistics based on 2.56×10^6 MC simulations. HJB control is computed as in Section 3.2, (2048×2048 grid) stored, and then used in the MC simulations. NN Training performance statistics based on 2.56×10^5 MC simulations. Control is computed using the NN framework in Section 3.3. Minimum withdrawal: 40. Maximum withdrawal: 80. $M : \#$ of withdrawals. $M : \#$ of rebalancing dates. $\epsilon = 10^{-6}$.*

Appendix H

Detailed benchmark strategy comparison

Table H.1 shows the results of the benchmark strategy, computed using MC simulations.

Constant withdrawal and constant allocation

Equity fraction p	$E[\sum_i i]/T$	ES (5%)	$Median[W_T]$
0.0	40	-302.57	-150.56
0.1	40	-238.62	-6.82
0.2	40	-245.48	168.10
0.3	40	-280.27	386.05
0.4	40	-330.37	649.58
0.5	40	-391.61	958.33
0.6	40	-461.54	1312.17
0.7	40	-538.04	1706.49
0.8	40	-619.31	2135.24

TABLE H.1: *Constant weight, constant withdrawals, synthetic market results. No tontine gains. Stock index: real capitalization weighted CRSP stocks; bond index: real 30-day T-bills. Parameters from Table 5.1. Scenario in Table 5.2. Units: thousands of dollars. Statistics based on* 2.56×10^6 *Monte Carlo simulation runs.* $T = 30$ *years.*