Computing Optimal Multi-period 130/30 Portfolios: A Neural Network Framework

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

[1]Cheriton School of Computer Science, University of Waterloo, Waterloo, N2L 3G1, Canada, chendi.ni@uwaterloo.ca
[2]Cheriton School of Computer Science, University of Waterloo, Waterloo, N2L 3G1, Canada, yuying@uwaterloo.ca
[3]Cheriton School of Computer Science, University of Waterloo, Waterloo, N2L 3G1, Canada, paforsyt@uwaterloo.ca

January 31, 2024

Abstract

We propose a neural network-based approach for the portfolio optimization problem under 130/30 constraints. Specifically, we formulate the dynamic benchmark outperforming problem as a multi-period stochastic optimal control problem. To overcome computational challenges due to high dimensionality and non-standard constraints which depend on distinction between long or short positions, we propose a novel relaxed-constraint neural network (RCNN) model to represent a feasible control. Using the proposed RCNN model, the original high dimensional constrained multi-period stochastic optimal control problem is transformed into an unconstrained optimization problem. This makes it feasible to discover the optimal 130/30 strategies directly by solving a DNN using gradient descent, without using dynamic programming. In addition, we prove mathematically that the RCNN model can approximate the optimal 130/30 strategy with an arbitrary precision. Based on the cumulative quadratic shortfall (CS) performance metric, using CRSP market data from 1/1/1926 to 1/1/2023, we computationally demonstrate the advantages of 130/30 strategies over long-only strategies in outperforming various benchmark portfolios, which includes better return and risk profile.

1 Introduction

1.1 Rise and fall of 130/30 strategy

130/30 strategies, or more generally, 1X0/X0 strategies, are known as relaxed constraint (RC) or active extension strategies (Ang et al., 2017). These relaxed constraint strategies constitute a class of investment strategies that allow portfolio managers to simultaneously hold long and short positions in a diverse universe of securities. Among these strategies, the 130/30 strategy, in particular, has gained significant popularity and recognition. In the case of the 130/30 strategy, the portfolio allows an aggregate long position of up to 130% of the total wealth and a total short position of up to 30%. Alternative notable variants of 1X0/X0 strategies encompass distinct structures spanning from 120/20 to 150/50.[1] The primary focus of this article is on the 130/30 strategy. However, the methodological framework proposed herein can readily be applied to other variants within the 1X0/X0 family.

The 130/30 investment strategy represents a promising advancement over traditional long-only portfolios by permitting short positions, enabling a more dynamic and sophisticated approach to portfolio management. This strategy is particularly appealing to investors amidst the current environment of global economic uncertainties and a potential regime switch in the financial market. By embracing short positions, the 130/30 strategy offers investors access to more flexible investment vehicles to adapt and capitalize on both positive and negative market trends, potentially enhancing their ability to navigate market uncertainty and achieve desired investment outcomes.

Shortly before the financial crisis of 2007-2009, the 130/30 strategy gained much attention from the investment community. Numerous studies surrounding the 130/30 strategy emerged amidst its increasing popularity (e.g. Johnson et al. (2007); Gastineau (2008); Lo and Patel (2008); Krusen et al. (2008)). In 2007, Tabb Group, a consultancy firm, projected that the size of 130/30 funds would reach $2 trillion by 2010 (Tabb and Johnson, 2007). Unfortunately, the actual performance of 130/30 funds since their inception was disappointing (Gay, 2012). By the end of 2012, the total identifiable assets of 130/30 funds, across Europe and the United States, amounted to a mere $9 billion, significantly below the optimistic expectation set in 2007 (Johnson, 2013). Since then, 130/30 funds have never regained the level of popularity witnessed in 2007. In a recent report by Morningstar, the concept of 130/30 funds is regarded as a "flopped" invention (Rekenthaler, 2022).

Some argue that the underperformance of 130/30 funds can be attributed to poor timing, as many of these funds were launched just before the onset of the financial crisis of 2007-2009, suffering market downturn alongside other active funds that had a net positive exposure to the market. However, reports indicate that, even when compared to long-only funds, which the 130/30 funds were designed to replace, the 130/30 funds did not demonstrate superior returns (Johnson, 2013). This is perplexing since 130/30 portfolios theoretically offer a larger solution space than long-only portfolios due to the relaxed constraint. Indeed, empirical evidence fails to support the anticipated performance advantage. Practitioners attribute

[1] The structure is capped at 150/50 due to Federal Reserve Board Regulation T (Federal Reserve Board, 1974).

the poor performance of 130/30 funds to the managers' inability to effectively apply their knowledge of long-only strategies to shorting stocks.

A closer examination reveals that many 130/30 funds, particularly those managed quantitatively, adopted a heuristic stock ranking system as a means to manage their 130/30 portfolios. This ranking system involved assessing all stocks within the investment universe and constructing the 130/30 portfolio by taking long positions in the highest-ranked stocks and short positions in the lowest-ranked stocks (Leibowitz et al., 2009; Korhonen and Kunz, 2010). In comparison to computing an optimal portfolio that maximizes a specific performance metric based on an appropriately predefined investment objective, the use of a stock ranking system appears simplistic and lacking in rigor. Fundamentally, this crude approach may be the underlying cause of the poor performance observed in 130/30 funds.

Can the optimal 130/30 strategy significantly outperform the long-only strategy under a suitable performance objective? To answer this question, it is necessary to first develop a computational method to solve the challenging multi-period stochastic optimal control problem under non-standard constraints, which is inherently more complex than that of long-only portfolios, particularly in a high dimensional context. The complexity stems, in part, from the non-standard 130/30 portfolio constraint, where the position bounds are contingent on distinction between long and short positions. This portfolio approach allows flexibility in taking short positions in any asset while imposing a limit on the total long position. In fact, the literature on the topic of 130/30 portfolio optimization, particularly in the context of multi-period settings, remains scarce, even to this day. Consequently, many 130/30 funds had to rely on less rigorous approaches, such as ranking systems, to construct their portfolios. Given the limited knowledge and tools available at the time, it is perhaps not surprising that these crude approaches failed to deliver satisfactory results. Consequently, one of the primary objectives of this paper is to develop an efficient computational method to identify optimal 130/30 strategies and evaluate their performance in comparison to long-only strategies.

1.2 Portfolio optimization under 130/30 constraints

In this article, we propose a neural network portfolio optimization framework to address the problem of constructing an optimal 130/30 portfolio. In particular, we formulate a multi-period stochastic optimal control problem with 130/30 portfolio constraints, in which security prices are assumed to be stochastic. The goal is to compute the parameters of the neural network which approximates the optimal control function that represents the optimal 130/30 strategy.

Existing literature related to 130/30 strategies primarily focuses on empirical studies and is not concerned with solving portfolio optimization problems under 130/30 constraints. On the other hand, literature in the domain of multi-period portfolio optimization either disregards allocation constraints at all (Zhou and Li, 2000; Li and Ng, 2000) or considers simple constraints such as long-only stock positions with unbounded leverage (Li et al., 2002), with minimal attention given to the unique characteristics of 130/30 strategies. This limited research can be attributed to the non-conventional nature of the total long position constraint imposed by the 130/30 rule, which existing methodologies do not address.

Given the scarcity of literature on the optimization of 130/30 portfolios, we aim to bridge this gap by providing a novel portfolio optimization framework that addresses the specific challenges posed by the 130/30 constraints. In particular, we propose to use a neural network model to approximate the optimal 130/30 strategy. On a high level, the idea of approximating the optimal control (allocation strategy) in a multi-period portfolio optimization problem is also considered in Han et al. (2016); Tsang and Wong (2020); Reppen et al. (2023); Li and Forsyth (2019); van Staden et al. (2023); Ni et al. (2022, 2023). However, Han et al. (2016); Tsang and Wong (2020) consider a stacked neural network approach which uses a different subnetwork at every rebalancing time, which is computationally inefficient. On the other hand, Li and Forsyth (2019); Reppen et al. (2023); van Staden et al. (2023) use a single recurrent neural network for all timesteps, in which time is considered a feature for the network model.

In neural network models for portfolio optimization, it is common to incorporate constraints by designing the model in a way that satisfies the constraints automatically. One common approach is to use the softmax activation function in the last layer of the network, ensuring that the output allocation fractions are non-negative and sum up to one. This technique is widely used in both portfolio optimization with long-only constraints and other fields such as classification and probabilistic modeling. By formulating the problem as an unconstrained optimization with appropriate activation functions, gradient-based optimization algorithms, e.g., stochastic gradient descent (SGD) can be applied readily (Buehler et al., 2019).

However, for the 130/30 constraints, which explicitly distinguish long and short positions to limit the total long position of the portfolio to 130%, it is not immediately obvious how to design such a neural network model. The closest work is the proposed methodology in Ni et al. (2023), in which the authors consider the multi-period portfolio optimization problem where the portfolio allows bounded leverage. However, in Ni et al. (2023), it is assumed that the manager can only short a specific pre-determined subset of the universe of securities, which does not pertain to the 130/30 portfolio setting.

To address this, we propose a novel *relaxed-constraint neural network* (RCNN) model that explicitly satisfies the 130/30 constraints by designing the neural network model with appropriate activation functions. Using the proposed RCNN, we convert the unusually constrained stochastic optimization problem into an unconstrained optimization problem, making computing a solution computationally feasible. Furthermore, we mathematically prove that the RCNN can approximate any optimal 130/30 strategy arbitrarily well, implying the capability of the proposed RCNN to yield the optimal 130/30 strategy via unconstrained optimization.

Since 130/30 funds are considered part of the long-only fund family in practice, and are often evaluated based on their relative performance to a passive benchmark portfolio, we deliberately choose a cumulative quadratic shortfall (CS) objective function that measures the tracking performance of the active portfolio against the benchmark. However, the RCNN is flexible and is immediately applicable to any continuous investment objective function. As long as standard optimization methods can backpropagate through the chosen objective function, our methodology can be applied to a wide range of investment problems with ease.

To demonstrate, in this paper, we utilize the proposed RCNN to computationally investigate the performance of the optimal 130/30 strategy in comparison to the optimal long-only strategies in a four-asset investment universe. Our results demonstrate clear advantages of the 130/30 strategy, showcasing superior returns and improved risk management outcomes. This empirical evidence not only validates the effectiveness of our proposed RCNN approach but also highlights the untapped potential of 130/30 portfolios that is yet to be fully realized.

2 Mathematical formulation

In this section, we mathematically formalize the multi-period portfolio optimization problem with 130/30 constraints.

As mentioned in Section 1.1, 130/30 funds are considered part of the extended family of long-only funds due to the net long exposure to the market, and are thus assessed against a passive benchmark. Therefore, we compare two portfolios: an active portfolio and a benchmark portfolio. We consider a fixed-horizon investment period of $[t_0, T]$. At time $t \in [t_0, T]$, let $W(t), \hat{W}(t)$ denote the wealth values of the active portfolio and the benchmark portfolio respectively. To ensure a fair assessment of the relative performance of the two portfolios, we assume both portfolios start with an equal initial value $w_0 > 0$, i.e. $W(t_0) = \hat{W}(t_0) = w_0 > 0$.

For simplicity, we assume that both the active portfolio and the benchmark portfolio can allocate among the same set of N_a assets. Let vector $\boldsymbol{S}(t) \in \mathbb{R}^{N_a}$ denote prices of the N_a underlying assets at time $t \in [t_0, T]$, where $\boldsymbol{S}(t) = (S_i(t) : i = 1, \cdots, N_a)^\top$. In this article, asset prices $\boldsymbol{S}(t) \in \mathbb{R}^{N_a}$ are assumed to be stochastic. In addition, let vectors $\boldsymbol{p}^{(t)} = (p_i^{(t)} : i = 1, \cdots, N_a)^\top \in \mathbb{R}^{N_a}$ and $\hat{\boldsymbol{p}}^{(t)} = (\hat{p}_i^{(t)} : i = 1, \cdots, N_a)^\top \in \mathbb{R}^{N_a}$ denote the allocation fractions to N_a underlying assets at time $t \in [t_0, T]$, for the active portfolio and the benchmark portfolio respectively. In this article, we consider a passive benchmark portfolio, i.e. $\hat{\boldsymbol{p}}^{(t)} \equiv \hat{\boldsymbol{p}}$, $\forall t \in [0, T]$, where $\hat{\boldsymbol{p}}$ is a constant vector that represents the pre-defined allocation fractions to each asset.

Under an optimal control perspective, the allocation vector $\boldsymbol{p}^{(t)}$ is a stochastic process denoting the control value at time t, which defines the evolution of the portfolio values. In general, the control is a function of the state variables that fully describe the state of the dynamic system. It is shown that under common assumptions on the asset price evolutions, e.g., jump-diffusion processes, the state variables for the optimal control are simply the portfolio values and time (Dang and Forsyth, 2014). While we consider the simple case of the portfolio values and time as state variables in this article, incorporating additional features as state variables poses no technical challenges for the proposed methodology. Mathematically, $\boldsymbol{p}^{(t)} = p(\boldsymbol{X}(t)) = (p_i(\boldsymbol{X}(t)) : i \in \{1, \cdots, N_a\})^\top \in \mathbb{R}^{N_a}$, where $\boldsymbol{X}(t) = \big(t, W(t), \hat{W}(t)\big)^\top \in \mathcal{X} \subseteq \mathbb{R}^3$, and $p_i : \mathcal{X} \mapsto \mathbb{R}$. Our goal is to find the optimal control function $p : \mathcal{X} \mapsto \mathbb{R}^3$, so that the relative performance measure of the active portfolio (following control p) over the benchmark portfolio (following $\hat{\boldsymbol{p}}^{(t)}$) is maximized.

We assume that the active portfolio and the benchmark portfolio follow the same discrete rebalancing schedule denoted by $\mathcal{T} \subseteq [t_0, T]$. Specifically, we consider an equally spaced discrete schedule with N rebalancing events, i.e.

$$\mathcal{T} = \Big\{ t_i : \ i = 0, \cdots, N-1 \Big\}, \tag{2.1}$$

where $t_i = i\Delta t$, and $\Delta t = T/N$.

2.1 Feasible strategies under 130/30 constraints

Here we first mathematically define the constraints for the feasible 130/30 strategies.

Definition 2.1. (Feasible strategies under 130/30 constraints). *A strategy* $p : \mathcal{X} \mapsto \mathbb{R}^{N_a}$ *is a feasible strategy under the 130/30 constraints if and only if*

$$Im(p) \subseteq \mathcal{Z}, \tag{2.2}$$

where $Im(p)$ *denotes the image of* p *and* $\mathcal{Z} \subset \mathbb{R}^{N_a}$ *encodes the 130/30 constraints, i.e. the summation to one constraint and the maximum total long position constraint, as follows,*

$$\mathcal{Z} = \left\{ \boldsymbol{z} \in \mathbb{R}^{N_a} \,\middle|\, \sum_{i=1}^{N_a} z_i = 1, \sum_{i=1}^{N_a} (z_i)^+ \leq p_{max}, \right\}, \tag{2.3}$$

where $(z_i)^+ = \max(z_i, 0)$ *is the positive part of* z_i*, and* p_{max} *is a constant.*

Furthermore, the set of all feasible strategies is denoted by $\mathcal{A}$*.*

Remark 2.1. (Choice of p_{max}). Since we focus on the 130/30 strategies in this article, we choose $p_{max} = 1.3$ as the maximum total long position of the portfolio (consequently, the total magnitude of fractions in the short position is upper bounded by 30%). However, the methodology can be readily applied to other 1X0/X0 strategies to accommodate different values of p_{max}. For example, for portfolios with the 150/50 constraints, p_{max} will be 1.5.

2.2 Stochastic optimal control problem

In this article, we consider a registered investment fund operating as a limited-liability legal entity (Carney, 1998). This structure is commonly found among investment funds in the United States (Fung and Hsieh, 1999; McCrary, 2004). Limited liability is a crucial characteristic of these funds that restricts investors' liability to the amount they have invested in the fund (Easterbrook and Fischel, 1985). Consequently, investors are protected from personal liability for the fund's debts or obligations beyond their initial investment.

Due to the nature of the active portfolio considered here, which permits both long and short positions, there is a theoretical possibility for the value of the portfolio to become negative. In such circumstances, the fund would initiate a bankruptcy process, resulting in the settlement of outstanding liabilities and the cessation of future trading activities. From a mathematical perspective, the portfolio value remains at zero throughout the remainder of the investment horizon. In addition, for simplicity, we assume no subsequent cash injections beyond the initial investment. Consequently, the evolution of the portfolio values, from the perspective of an investor in a limited-liability fund, can be described as follows:

$$\begin{cases} W(t_{j+1}) = \begin{cases} \Big(\sum\limits_{i=1}^{N_a} p_i(\boldsymbol{X}(t_j)) \cdot \frac{S_i(t_{j+1})-S_i(t_j)}{S_i(t_j)}\Big) W(t_j), & \text{if } W(t_j) > 0, \\ 0, & \text{if } W(t_j) \leq 0, \end{cases} \qquad \forall j \in \{0, \cdots, N-1\}, \\ \hat{W}(t_{j+1}) = \Big(\sum\limits_{i=1}^{N_a} \hat{p}_i \cdot \frac{S_i(t_{j+1})-S_i(t_j)}{S_i(t_j)}\Big) \hat{W}(t_j), \end{cases} \tag{2.4}$$

Let $\mathcal{W}_p = \{W(t), t \in \mathcal{T}\}$ and $\hat{\mathcal{W}}_{\hat{p}} = \{\hat{W}(t), t \in \mathcal{T}\}$ represent trajectories of the portfolio values for the active portfolio and the benchmark portfolio respectively, following the dynamics specified in equation (2.4). We denote an investment metric by $F(\mathcal{W}_p, \hat{\mathcal{W}}_{\hat{p}}) \in \mathbb{R}$, which quantifies the relative performance of the active portfolio in relation to the benchmark portfolio based on their respective value trajectories.

Note that the value trajectories $\mathcal{W}_p$, $\hat{\mathcal{W}}_{\hat{p}}$, and the performance metric $F(\mathcal{W}_p, \hat{\mathcal{W}}_{\hat{p}})$ are also stochastic, since asset prices $\boldsymbol{S}(t) \in \mathbb{R}^{N_a}$ are stochastic. Therefore, when investment managers aim to optimize an investment metric, the evaluation typically is the expectation of the chosen random performance metric.

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

$$\text{(Stochastic optimization problem):} \quad \inf_{p \in \mathcal{A}} \mathbb{E}_p^{(t_0,w_0)}\big[F(\mathcal{W}_p, \hat{\mathcal{W}}_{\hat{p}})\big]. \tag{2.5}$$

Solving the constrained stochastic optimal control problem (2.5) is challenging due to the presence of the intricate feasibility constraint (2.2) represented by the set $\mathcal{A}$. In the subsequent section, we propose a neural network methodology that circumvents the complexity in handling this constraint through the introduction of a specially designed relaxed-constraint neural network (RCNN) model.

3 Relaxed-constraint neural network (RCNN)

In this section, we propose a neural network for solving the stochastic optimization problem (2.5) with 130/30 constraints. Given the non-standard nature of these constraints, which involve discerning between long and short positions and imposing an upper limit on total long positions, the key idea is to approximate the optimal control function by designing a neural network function that automatically satisfies the feasibility constraint (2.2).

In other words, we want to design a neural network $f_{\boldsymbol{\theta}} : \mathcal{X} \mapsto \mathbb{R}^{N_a}$, where $\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}$ represents the parameters of the neural network (i.e., weights and biases), that approximates the control function p as follows:

$$p(\boldsymbol{X}(t)) \simeq f_{\boldsymbol{\theta}}(\boldsymbol{X}(t)), \tag{3.1}$$

and

$$f_{\boldsymbol{\theta}}(\boldsymbol{X}(t)) \in \mathcal{Z},\ \forall \boldsymbol{X}(t) \in \mathcal{X}, \tag{3.2}$$

where $\mathcal{Z}$ is defined in (2.3).

Then, the output of the neural network $f_{\boldsymbol{\theta}}$ is automatically a feasible 130/30 strategy, i.e., $f_{\boldsymbol{\theta}} \in \mathcal{A}$, where $\mathcal{A}$ is the set of 130/30 strategies described in Definition 2.1 with $p_{max} = 1.3$. Consequently, the original constrained optimization problem (2.5) can be transformed into the following unconstrained optimization problem, which is computationally solvable using standard optimization methods for unconstrained minimization problems:

$$\text{(Unconstrained optimization problem):} \quad \inf_{\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}} \mathbb{E}_{f_{\boldsymbol{\theta}}}^{(t_0,w_0)}\big[F(\mathcal{W}_{\boldsymbol{\theta}}, \hat{\mathcal{W}}_{\hat{p}})\big]. \tag{3.3}$$

Here $\mathcal{W}_{\boldsymbol{\theta}}$ is the wealth trajectory of the active portfolio with control following the neural network parameterized by $\boldsymbol{\theta}$.

3.1 Model design

To achieve 130/30 constraint explicitly, our proposed RCNN applies a tailored activation function to the output of a fully connected feed-forward network, which is described below following notations used in Lu and Lu (2020).

Definition 3.1. (Fully connected feedforward neural network $\tilde{f}_{\boldsymbol{\theta}}$). *A fully connected feed-forward neural network (FNN) maps an input vector $\boldsymbol{x} \in \mathbb{R}^{d_0}$ to an output vector $\boldsymbol{h} \in \mathbb{R}^{d_{K+1}}$, and contains K hidden layers of sizes $d_1, \cdots, d_K$. The neural network is parameterized by the weight matrices $\boldsymbol{\theta}^{(k)} \in \mathbb{R}^{d_{k-1} \times d_k}$ and bias vectors $\boldsymbol{\theta}_b^{(k)} \in \mathbb{R}^{d_k}$, for $k = 1, \cdots, K+1$. Then, the output $\boldsymbol{h}$ is derived from the input $\boldsymbol{x}$ iteratively as follows.*

$$\begin{cases} \boldsymbol{x}^{(0)} = \boldsymbol{x}, \\ \boldsymbol{x}^{(k)} = \sigma\Big(\big(\boldsymbol{\theta}^{(k)}\big)^\top \cdot \boldsymbol{x}^{(k-1)} + \boldsymbol{\theta}_b^{(k)}\Big), 1 \le k \le K, \\ \boldsymbol{h} = \big(\boldsymbol{\theta}^{(K+1)}\big)^\top \cdot \boldsymbol{x}^{(K)} + \boldsymbol{\theta}_b^{(K+1)}. \end{cases} \tag{3.4}$$

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

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

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

We propose the following relaxed-constraint activation function, which is applied to the output of FNN $\tilde{f}_{\boldsymbol{\theta}}$.

Definition 3.2. (Relaxed-constraint activation function). *Let* $\boldsymbol{h} = (h_1, \cdots, h_{N_a-1})^\top \in \mathbb{R}^{N_a-1}$ *be any output of a FNN* $\tilde{f}_{\boldsymbol{\theta}}$. *Given a constant* $\alpha \in \mathbb{R}$, $\alpha \neq \frac{1}{2}$, *define the "bounded mapping",* $\phi_1 : \mathbb{R}^{N_a-1} \mapsto \mathbb{R}^{N_a-1}$, *as follows,*

$$\phi_1(\boldsymbol{h}) = (1-\alpha) + (2\alpha - 1) \cdot \sigma(\boldsymbol{h}). \tag{3.5}$$

Here $\sigma : \mathbb{R}^{N_a-1} \mapsto \mathbb{R}^{N_a-1}$ *is the pointwise sigmoid function, i.e.* $[\sigma(\boldsymbol{h})]_i = \sigma(h_i)$. ϕ_1 *maps the unbounded real vector space* $\mathbb{R}^{N_a-1}$ *into the bounded open set of* $(1-\alpha, \alpha)^{N_a-1}$, *if* $\alpha > \frac{1}{2}$, *or* $(\alpha, 1-\alpha)^{N_a-1}$, *if* $\alpha < \frac{1}{2}$.[2]

Next, for any $\boldsymbol{u} = (u_1, \cdots, u_{N_a-1})^\top \in \mathbb{R}^{N_a-1}$, *define the "extension mapping",* $\phi_2 : \mathbb{R}^{N_a-1} \mapsto \mathbb{R}^{N_a}$, *as*

$$\phi_2(\boldsymbol{u}) = \left(\boldsymbol{u}, 1 - \sum_{i=1}^{N_a-1} u_i\right)^\top. \tag{3.6}$$

ϕ_2 *extends a vector from* $\mathbb{R}^{N_a-1}$ *into a vector in* $\mathbb{R}^{N_a}$, *which has the property that all entries of this vector sum up to one.*

Furthermore, for any $\boldsymbol{v} = (v_1, \cdots, v_{N_a})^\top \in \mathbb{R}^{N_a}$, *and a constant* $p_{max} > 1$, *define the "scaling mapping",* $\phi_3 : \mathbb{R}^{N_a} \mapsto \mathbb{R}^{N_a}$, *as*

$$\phi_3(\boldsymbol{v}) = \begin{cases} \boldsymbol{v}, & \text{if } \sum_{i=1}^{N_a} (v_i)^+ \leq p_{max}, \\ \boldsymbol{v}^+ \cdot \frac{p_{max}}{\sum_{i=1}^{N_a} (v_i)^+} + \boldsymbol{v}^- \cdot \frac{1-p_{max}}{1-\sum_{i=1}^{N_a} (v_i)^+}, & \text{if } \sum_{i=1}^{N_a} (v_i)^+ > p_{max}. \end{cases} \tag{3.7}$$

Here $(v_i)^+ = \max(v_i, 0), \forall i \in \{1, \cdots, N_a\}$. $\boldsymbol{v}^+ = \big(\max(v_1, 0), \cdots, \max(v_{N_a}, 0)\big)^\top \in \mathbb{R}^{N_a}$ *and* $\boldsymbol{v}^- = \big(\min(v_1, 0), \cdots, \min(v_{N_a}, 0)\big)^\top \in \mathbb{R}^{N_a}$ *are the positive and negative parts of vector* $\boldsymbol{v}$. ϕ_3 *scales any vector in* $\mathbb{R}^{N_a}$ *so that the sum of all positive entries of the scaled vector is less than or equal to the constant* p_{max}.

Finally, define the "relaxed-constraint activation function", $\phi : \mathbb{R}^{N_a-1} \mapsto \mathbb{R}^{N_a}$, *as*

$$\phi = \phi_3 \circ \phi_2 \circ \phi_1. \tag{3.8}$$

In other words, the relaxed-constraint activation function ϕ *is a composition of* ϕ_3, ϕ_2 *and* ϕ_1.

Finally, we define the relaxed-constraint neural network (RCNN) as follows.

[2]Obviously, if $\alpha = \frac{1}{2}$, then ϕ_1 becomes a trivial constant mapping.

Definition 3.3. (Relaxed-constraint neural network). *Let $\mathcal{X} \subset \mathbb{R}^{d_0}$ be the state space (domain of the feature state variables). Given hyperparameters $\big(K, (d_1, \cdots, d_K)^\top\big)$ representing the number of hidden layers and their sizes, and given weight and bias parameter $\boldsymbol{\theta}$, let $\tilde{f}_{\boldsymbol{\theta}} : \mathcal{X} \mapsto \mathbb{R}^{N_a - 1}$ be the fully connected feedforward neural network (FNN) function parameterized by $\boldsymbol{\theta}$ as defined in Definition 3.1. Let $\phi : \mathbb{R}^{N_a-1} \mapsto \mathbb{R}^{N_a}$ be the relaxed-constraint activation function defined in Definition 3.2. Then, we define the relaxed-constraint neural network (RCNN) function, $f_{\boldsymbol{\theta}} : \mathcal{X} \mapsto \mathbb{R}^{N_a}$, as*

$$f_{\boldsymbol{\theta}} = \phi \circ \tilde{f}_{\boldsymbol{\theta}}. \tag{3.9}$$

We first establish the following Lemma to show that the RCNN function defined in Definition 3.3 is a feasible strategy that satisfies the 130/30 constraints.

Lemma 3.1. *(Feasibility of RCNN function). For any hyperparameters $\big(K, (d_1, \cdots, d_K)^\top\big)$ (i.e. number of hidden layers and their sizes), and parameter $\boldsymbol{\theta}$, let $f_{\boldsymbol{\theta}}$ be the corresponding RCNN function defined in Definition 3.3. Then, for any $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^{d_0}$, $f_{\boldsymbol{\theta}}(\boldsymbol{x}) \in \mathcal{Z}$, where $\mathcal{Z}$ is the feasibility region defined in (2.3), i.e., $f_{\boldsymbol{\theta}}$ is a feasible strategy under the 130/30 constraints, as described in Definition 2.1.*

Proof. For any $\boldsymbol{x} \in \mathcal{X}$, let $\boldsymbol{h} = \tilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $\boldsymbol{y} = f_{\boldsymbol{\theta}}(\boldsymbol{x})$, where $\tilde{f}_{\boldsymbol{\theta}}$ is the FNN in Definition 3.1. We show that $\boldsymbol{y} \in \mathcal{Z}$. Let $\boldsymbol{y} = (y_1, \cdots, y_{N_a})^\top$. This is equivalent to establishing

$$\begin{cases} \sum_{i=1}^{N_a} y_i = 1, \\ \sum_{i=1}^{N_a} (y_i)^+ \leq p_{max}. \end{cases} \tag{3.10}$$

Let ϕ_1, ϕ_2 and ϕ_3 be the bounded mapping, extension mapping, and scaling mapping in Definition 3.2. Let $\boldsymbol{v} = \phi_2(\phi_1(\boldsymbol{h})) \in \mathbb{R}^{N_a}$ and $\boldsymbol{y} = \phi_3(\boldsymbol{v})$. Note that $\sum_{i=1}^{N_a} v_i = 1$ follows from definition of ϕ_2.

If $\sum_{i=1}^{N_a} (v_i)^+ \leq p_{max}$, then $\boldsymbol{y} = \phi_3(\boldsymbol{v}) = \boldsymbol{v} \in \mathcal{Z}$. On the other hand, if $\sum_{i=1}^{N_a} (v_i)^+ > p_{max}$, then

$$\boldsymbol{y} = \phi_3(\boldsymbol{v}) = (\boldsymbol{v})^+ \cdot \frac{p_{max}}{\sum_{i=1}^{N_a} (v_i)^+} + (\boldsymbol{v})^- \cdot \frac{1 - p_{max}}{1 - \sum_{i=1}^{N_a} (v_i)^+}. \tag{3.11}$$

Note that $\frac{p_{max}}{\sum_{i=1}^{N_a} (v_i)^+} > 0$, $\frac{1-p_{max}}{1-\sum_{i=1}^{N_a} (v_i)^+} > 0$. Thus,

$$\begin{cases} (\boldsymbol{y})^+ = (\boldsymbol{v})^+ \cdot \frac{p_{max}}{\sum_{i=1}^{N_a} (v_i)^+}, \\ (\boldsymbol{y})^- = (\boldsymbol{v})^- \cdot \frac{1-p_{max}}{1-\sum_{i=1}^{N_a} (v_i)^+}. \end{cases} \tag{3.12}$$

Then, we have

$$\begin{cases} \sum_{i=1}^{N_a} (y_i)^+ = \frac{p_{max}}{\sum_{i=1}^{N_a} (v_i)^+} \cdot \big(\sum_{i=1}^{N_a} (v_i)^+\big) = p_{max} \leq p_{max}, \\ \sum_{i=1}^{N_a} (y_i)^- = \frac{1-p_{max}}{1-\sum_{i=1}^{N_a} (v_i)^+} \cdot \big(\sum_{i=1}^{N_a} (v_i)^-\big) = \frac{1-p_{max}}{\sum_{i=1}^{N_a} (v_i)^-} \cdot (\sum_{i=1}^{N_a} (v_i)^-) = 1 - p_{max}, \end{cases} \tag{3.13}$$

and

$$\sum_{i=1}^{N_a} y_i = \sum_{i=1}^{N_a} (y_i)^+ + \sum_{i=1}^{N_a} (y_i)^- = 1. \tag{3.14}$$

Therefore, both conditions in (3.10) are satisfied, thus concluding the proof. □

Remark 3.1. (Intuition behind the RCNN design). As shown in Definition 3.3, the RCNN function is constructed by applying the relaxed-constraint activation function ϕ in Definition 3.2 to the output of a FNN in Definition 3.1. The FNN provides universal approximation by connecting sufficient hidden layers and nodes via the sigmoid activation functions. The relaxed-constraint activation function ϕ, on the other hand, guarantees that the RCNN function satisfies the 130/30 constraints. In particular, recall the three mappings ϕ_1, ϕ_2 and ϕ_3 in Definition 3.2. ϕ_1 maps the output of the FNN (which is scattered over $\mathbb{R}^{N_a-1}$) into a bounded region of $(1-\alpha, \alpha)^{N_a-1}$, if $\alpha > \frac{1}{2}$, or $(\alpha, 1-\alpha)^{N_a-1}$, if $\alpha < \frac{1}{2}$. In the context of asset allocation, the output of ϕ_1 represents an initial estimate of the allocation fraction for the first $N_a - 1$ assets. Since the total long position is upper bounded by $p_{max} > 1$, the allocation fraction for each asset lies in $[1 - p_{max}, p_{max}]$. Therefore, we choose α to be slightly larger than p_{max} (in numerical experiments, we use $\alpha = p_{max} + \epsilon$ where $\epsilon = 10^{-6}$ is a small constant), so that $(1-\alpha, \alpha)^{N_a-1}$ tightly covers $[1 - p_{max}, p_{max}]^{N_a-1}$. As we will show in the following lemma, choosing $\alpha > p_{max}$ guarantees the existence of a right inverse of ϕ, which is critical to ensuring the RCNN function can approximate the optimal 130/30 strategy accurately. Subsequently, ϕ_2 guarantees that the summation to one constraint is satisfied, and ϕ_3 guarantees that the maximum total long position constraint is satisfied while preserving the summation to one property obtained from ϕ_2. It is worth noting that without ϕ_1, the RCNN function is still a feasible 130/30 strategy. However, we find that applying ϕ_1 allows faster convergence in the training of the neural network, since it narrows down roughly into the correct range.

As discussed in Remark 3.1, we show that ϕ has a right inverse if $\alpha > p_{max}$.

Lemma 3.2. *(Existence of right-inverse of ϕ). Given a relaxed constraint activation function $\phi : \mathbb{R}^{N_a-1} \mapsto \mathbb{R}^{N_a}$ as defined in Definition 3.2. Let p_{max} be the maximum total long position in (2.3). If $\alpha > p_{max}$, then there exists a function $\overrightarrow{\phi} : Im(\phi) \mapsto \mathbb{R}^{N_a-1}$, such that*

$$\phi(\overrightarrow{\phi}(\boldsymbol{y})) = \boldsymbol{y}, \forall \boldsymbol{y} \in Im(\phi),$$

i.e., $\overrightarrow{\phi}$ is the right-inverse of ϕ.

Proof. Let $\boldsymbol{y} = (y_1, \cdots, y_{N_a})^\top \in Im(\phi) \subset \mathbb{R}^{N_a}$. According to Lemma 3.1,

$$Im(\phi) \subseteq \mathcal{Z}. \tag{3.15}$$

From (3.10), $y_i \in [1 - p_{max}, p_{max}], \forall i \in \{1, \cdots, N_a\}$. Then,

$$\frac{y_i - 1 + \alpha}{2\alpha - 1} \in \left[\frac{\alpha - p_{max}}{2\alpha - 1}, \frac{\alpha + p_{max} - 1}{2\alpha - 1}\right] \subset \left(\frac{0}{2\alpha - 1}, \frac{2\alpha - 1}{2\alpha - 1}\right) = (0, 1). \tag{3.16}$$

We can then define $\overrightarrow{\phi}: Im(\phi) \mapsto \mathbb{R}^{N_a-1}$ as

$$\overrightarrow{\phi}(\boldsymbol{y}) = \Big(\sigma^{-1}\big(\frac{y_1 - 1 + \alpha}{2\alpha - 1}\big), \cdots, \sigma^{-1}\big(\frac{y_{N_a-1} - 1 + \alpha}{2\alpha - 1}\big)\Big)^\top, \tag{3.17}$$

where $\sigma^{-1}(\cdot)$ is the inverse function of the sigmoid function.

Then, it can be easily verified that $\overrightarrow{\phi}$ is a right-inverse of ϕ, i.e. for all $\boldsymbol{y} \in Im(\phi)$

$$\phi\big(\overrightarrow{\phi}(\boldsymbol{y})\big) = \boldsymbol{y}. \tag{3.18}$$

□

Denote the wealth trajectory corresponding to control $f_{\boldsymbol{\theta}}$ by $\mathcal{W}_{\boldsymbol{\theta}}$. Then, the original optimization problem (2.5) constrained by the 130/30 rule is converted into the following unconstrained optimization problem:

$$\text{(Unconstrained parameterized problem):} \quad \inf_{\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}} \mathbb{E}_{f_{\boldsymbol{\theta}}}^{(t_0, w_0)}\big[F(\mathcal{W}_{\boldsymbol{\theta}}, \hat{\mathcal{W}}_{\hat{p}})\big]. \tag{3.19}$$

However, an essential question remains unanswered: for the optimal 130/30 strategy p^*, can we find a selection of the hyperparameters $\Big(K, (d_1, \cdots, d_K)^\top\Big)$ and parameter $\boldsymbol{\theta}$ so that the corresponding RCNN function $f_{\boldsymbol{\theta}}$ can be arbitrarily close to p^*? If the answer is affirmative, it assures that solving the unconstrained problem (3.19) can yield an accurate approximation of the optimal 130/30 strategy. To address this crucial question, we establish an approximation theorem in the following section, which provides a formal proof of the existence of such approximations. This theorem will justify the theoretical effectiveness of the proposed neural network model for approximating the optimal 130/30 strategy.

4 Universal approximation theorem for RCNN

Before we prove the approximation theorem for the RCNN, we first describe some standard assumptions necessary for establishing the mathematical results.

Assumption 4.1. *(Assumption on state space and optimal control).*

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

(ii) The optimal control $p^* : \mathcal{X} \mapsto \mathcal{Z}$ is continuous.

Remark 4.1. (Remark on Assumption 4.1). In our particular problem of outperforming a benchmark portfolio, the state variable vector is $X(t) = (t, W(t), \hat{W}(t))^\top \in \mathcal{X}$ where $t \in [0, T]$. In this case, assumption (i) is equivalent to the assumption that the wealth of the active portfolio and benchmark portfolio is bounded, i.e. $\mathcal{X} = [0, T] \times [0, w_{max}] \times [0, \hat{w}_{max}]$, where w_{max} and $\hat{w}_{max}$ are the respective upper bounds for the portfolio values. Assumption (ii) assumes that the optimal control is a continuous function, which is an intuitive and common assumption.

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

Lemma 4.1. *Let $\mathcal{X} \subset \mathbb{R}^l$ be a compact set, and $\mathcal{Y} \subset \mathbb{R}^m$. Let $\rho : \mathbb{R}^n \mapsto \mathcal{Y}$ satisfy the following:*

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

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

Then, for any continuous $g : \mathcal{X} \mapsto \mathcal{Y}$, and any $\epsilon > 0$, there exists a choice of hyperparameters $\left(K, (d_1, \cdots, d_K)^\top\right)$ and parameter $\boldsymbol{\theta}$, such that the corresponding FNN $\tilde{f}_{\boldsymbol{\theta}} : \mathcal{X} \mapsto \mathbb{R}^n$ described in Definition 3.1 satisfies

$$\sup_{\boldsymbol{x} \in \mathcal{X}} \|\rho\left(\tilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x})\right) - g(\boldsymbol{x})\| < \epsilon, \forall \boldsymbol{x} \in \mathcal{X}. \tag{4.1}$$

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

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

Intuitively, the second assumption of Lemma 4.1 allows the use of an activation function (such as the softmax function) which outputs an open set, as long as this open set is dense in $\mathcal{Y}$ (which can be a closed set). The two assumptions ensure the existence of a continuous mapping whose image almost covers $\mathcal{Y}$.

Next, we present the approximation theorem for the RCNN.

Theorem 4.1. *(Approximation of optimal 130/30 strategy). Assume the constant α in Definition 3.2 satisfies $\alpha > p_{max}$. Given the optimal control p^* and following Assumption 4.1, $\forall \epsilon > 0$, there exists $\left(K, (d_1, \cdots, d_K)^\top\right)$, and $\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}$ such that the corresponding RCNN $f_{\boldsymbol{\theta}}$ described in Definition (3.3) satisfies the following:*

$$\sup_{x \in \mathcal{X}} \|f_{\boldsymbol{\theta}}(x) - p^*(x)\| < \epsilon. \tag{4.2}$$

Proof. Let ϕ be the relaxed constraint activation function in Definition 3.2. According to Lemma 3.1,

$$Im(\phi) \subseteq \mathcal{Z}. \tag{4.3}$$

Furthermore, $\forall \boldsymbol{z} = (z_1, \cdots, z_{N_a})^\top \in \mathcal{Z}$, $z_i \in [1 - p_{max}, p_{max}], \forall i \in \{1, \cdots, N_a\}$. For the same reason as (3.16), $\overrightarrow{\phi}(\boldsymbol{z})$ is well-defined and $\phi(\overrightarrow{\phi}(\boldsymbol{z})) = \boldsymbol{z}$. Therefore,

$$\mathcal{Z} \subseteq Im(\phi). \tag{4.4}$$

Combine (4.4) with (4.3), we know that $Im(\phi) = \mathcal{Z}$, and thus $Im(\phi)$ is dense in $\mathcal{Z}$. In addition, ϕ is continuous and has a right-inverse, according to Lemma 3.2.

Then, following Lemma 4.1, we know that there exists $\Big(K, (d_1, \cdots, d_K)^\top\Big)$, and $\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}$, such that the corresponding FNN $\tilde{f}_{\boldsymbol{\theta}}$ (Definition 3.1) and RCNN $f_{\boldsymbol{\theta}} = \phi \circ \tilde{f}_{\boldsymbol{\theta}}$ satisfy

$$\sup_{x \in \mathcal{X}} \| f_{\boldsymbol{\theta}}(x) - p^*(x) \| = \sup_{x \in \mathcal{X}} \| \phi\big(\tilde{f}_{\boldsymbol{\theta}}(x)\big) - p^*(x) \| < \epsilon. \tag{4.5}$$

□

Remark 4.2. (Implication of Theorem 4.1). Theorem 4.1 provides valuable insight that, for any feasible control that satisfies the 130/30 constraints, there exists a selection of hyper-parameters and weight parameter values that enables the corresponding RCNN to approximate the control arbitrarily well. Intuitively, when RCNN is sufficiently large in terms of the number of hidden nodes, solving the unconstrained optimization problem (3.19) results in a solution that closely approximates the optimal control p^*.

5 Computational performance evaluation

In this section, we computationally assess the performance of the RCNN model based on market data directly. We compute the optimal parameters $\boldsymbol{\theta}^*$ of the RCNN model based on problem (3.19) and assess the performance of the corresponding optimal strategy. Subsequently, we first provide a brief overview of how we compute the parameters.

For the numerical experiments, we approximate the expectation in (3.19) by utilizing a finite set of samples from a training set $\boldsymbol{Y} = Y^{(j)}, j = 1, \cdots, N_d$, where N_d denotes the number of samples. Here, $Y^{(j)}$ represents a sample of a time series comprising joint observations of asset returns $\big\{R_i(t), i \in \{1, \cdots, N_a\}\big\}$, observed at $t \in \mathcal{T}$.[3] Mathematically, the approximation of problem (3.19) can be formulated as follows:

$$\inf_{\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}} \left\{ \frac{1}{N_d} \sum_{j=1}^{N_d} F\left(\mathcal{W}_{\boldsymbol{\theta}}^{(j)}, \hat{\mathcal{W}}_{\hat{p}}^{(j)} \right) \right\}. \tag{5.1}$$

Here, $\mathcal{W}_{\boldsymbol{\theta}}^{(j)} = \big(W_{\boldsymbol{\theta}}^{(j)}(t_0), \cdots, W_{\boldsymbol{\theta}}^{(j)}(t_N)\big)$ represents the wealth trajectory of the active portfolio, which follows the RCNN model parameterized by $\boldsymbol{\theta}$. $\hat{\mathcal{W}}_{\hat{p}}^{(j)} = \big(\hat{W}^{(j)}(t_0), \cdots, \hat{W}^{(j)}(t_N)\big)$ denotes the wealth trajectory of the benchmark portfolio, following the benchmark strategy $\hat{p}$. Both portfolios are evaluated based on the j-th time series sample, $Y^{(j)}$.

We adopt a shallow neural network structure, specifically, with a single hidden layer consisting of 10 hidden nodes ($K = 1$ and $d_1 = 10$). The feature input to the RCNN network is a 3-tuple vector $(t, W_{\boldsymbol{\theta}}(t), \hat{W}(t))^\top$. Here, at time $t \in [t_0, T]$, $W_{\boldsymbol{\theta}}(t)$ represents the wealth of the active portfolio governed by the RCNN model parameterized by $\boldsymbol{\theta}$, while $\hat{W}(t)$ represents the wealth of the benchmark portfolio.

An important computational advantage of the proposed neural network framework is its capability to directly compute model parameters using gradient descent-based methods. In

[3] It should be noted that the corresponding set of asset prices can be easily inferred from the set of asset returns, and vice versa.

essence, the model functions as a recurrent neural network (RNN), and the procedure for calculating the gradient for a specific path in a sample set $\boldsymbol{Y}$ is outlined in Appendix §A.

Subsequently, the optimal parameter $\boldsymbol{\theta}^*$ can be determined numerically by solving problem (5.1) using gradient-based optimization algorithms such as SGD or ADAM (Kingma and Ba, 2014). This process is commonly referred to as the "training" of the neural network model, and the set $\boldsymbol{Y}$ is commonly known as the training dataset (Goodfellow et al., 2016). Once the model is trained, we evaluate the performance of the model on a distinct set of data samples $\boldsymbol{Y}_{test}$, often referred to as the testing dataset.

5.1 Bootstrap resampled data

We use the U.S. monthly data from the Center for Research in Security Prices (CRSP)[4] from January 1926 to January 2023 as the original dataset for computational assessment. In particular, we obtain the real historical returns of the equal-weighted/cap-weighted U.S. stock indexes and 10-year/30-day treasury indexes by adjusting for the CPI index.

Conventional approaches in mathematical finance often involve fitting synthetic models, such as stochastic process models, to the original historical asset price data and subsequently resampling from these fitted models (Merton, 1976; Kou, 2002). While synthetic models offer the advantage of often providing closed-form solutions, they also present certain disadvantages. Firstly, accurate estimation of model parameters is often challenging and requires a substantial historical data period (Black, 1993; Brigo et al., 2008). Secondly, the assumptions for synthetic models may not always align sufficiently well with the characteristics of the real-world financial markets. Consequently, the validity of synthetic models can be up to debate.

To avoid these limitations, we employ the stationary block bootstrap resampling technique to generate the training and testing datasets. In essence, the block bootstrap resampling method randomly selects blocks from the underlying historical time series data and combines them to form a new time series path. In contrast to synthetic models, the bootstrap resampling method avoids imposing assumptions regarding the underlying data-generating model and can be a relatively unbiased approach to resampling data from historical time series.

The stationary block bootstrap resampling method, originally proposed by Politis and Romano (1994), preserves the stationarity of the original time series data by employing random block sizes. The pseudo-code for the algorithm can be found in Appendix B. In our study, we adopt an expected block size of 6 months and resample 20,000 paths for both the training and testing datasets from the real historical returns.

Finally, we note that the use of bootstrap resampling for testing investment strategies is widely adopted by practitioners (Alizadeh and Nomikos, 2007; Cogneau and Zakamouline,

[4]©2023 Center for Research in Security Prices (CRSP), The University of Chicago Booth School of Business. Wharton Research Data Services (WRDS) was used in preparing this article. This service and the data available thereon constitute valuable intellectual property and trade secrets of WRDS and/or its third-party suppliers.

2013; Dichtl et al., 2016; Scott and Cavaglia, 2017; Shahzad et al., 2019; Cavaglia et al., 2022; Simonian and Martirosyan, 2022) as well as academics (Anarkulova et al., 2022).

5.2 Choice of performance metric

A commonly used metric for evaluating the relative performance of an active portfolio compared to a benchmark portfolio is the information ratio (IR). In a dynamic context, the IR of the active portfolio over the interval $[0, T]$ is defined as follows:

$$IR_p^{(t_0,w_0)} = \frac{\mathbb{E}_p^{(t_0,w_0)}\Big[W(T) - \hat{W}(T)\Big]}{Stdev_p^{(t_0,w_0)}\Big[W(T) - \hat{W}(T)\Big]}. \tag{5.2}$$

where $W(T)$ and $\hat{W}(T)$ represent the terminal value of the active and benchmark portfolios, respectively. The IR measures the volatility-adjusted relative performance of the active portfolio at the terminal time T. However, it does not capture the intermediate tracking performance of the portfolio, which is a crucial aspect of evaluating the performance of an active portfolio.

To address this limitation, van Staden et al. (2024) introduce the cumulative quadratic tracking difference (CD) metric:

$$(CD): \quad F(\mathcal{W}_p, \hat{\mathcal{W}}_{\hat{p}}) = \sum_{t \in \mathcal{T} \cup \{T\}} \Big(W(t) - e^{\beta t}\hat{W}(t)\Big)^2 \Delta t. \tag{5.3}$$

In the CD metric, the parameter β represents a predefined outperformance target. The CD metric quantifies how closely the value of the active portfolio tracks an *elevated target* $e^{\beta t}\hat{W}(t)$, a concept proposed by Ni et al. (2022). Unlike (5.2), the CD objective captures the intermediate tracking performance of the portfolio, and the parameter β provides a clear performance goal.

However, the CD metric (5.3) minimizes the relative performance between the active portfolio and the elevated target quadratically. In practice, the outperformance of the active portfolio over the elevated target is desirable. Therefore, instead of the CD metric, we use the following cumulative quadratic shortfall (CS) metric in the numerical experiments.

$$(CS): \quad F(\mathcal{W}_p, \hat{\mathcal{W}}_{\hat{p}}) = \sum_{t \in \mathcal{T} \cup \{T\}} \Big(\min\big(W(t) - e^{\beta t}\hat{W}(t), 0\big)\Big)^2 \Delta t. \tag{5.4}$$

Consequently, we investigate the following optimization problem in numerical experiments:

$$\inf_{\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}} \left\{ \frac{1}{N_d} \sum_{j=1}^{N_d} \sum_{t \in \mathcal{T} \cup \{T\}} \Big(\min\big(W_{\boldsymbol{\theta}}^{(j)}(t) - e^{\beta t}\hat{W}^{(j)}(t), 0\big)\Big)^2 \Delta t \right\}. \tag{5.5}$$

It is of course possible to use various techniques to generate training data, see e.g. Yoon et al. (2019). However, we focus on block bootstrap resampling in this work, due to its wide acceptance in the practitioner community. Finally, it is worth mentioning that our proposed neural network approach is agnostic to the choice of the objective function and can be applied to a broad range of performance metrics.

5.3 Performance on resampled market data

In this section, we present a case study focusing on optimal asset allocation for outperforming a benchmark portfolio, imposing the 130/30 constraints. The objective of the study is to assess the potential advantages offered by a 130/30 portfolio compared to long-only portfolios. Specifically, we analyze and compare the performance of the optimal 130/30 portfolio, as determined by the RCNN, with the optimal long-only portfolio generated by the state-of-the-art model for long-only constraints (Li and Forsyth, 2019; Ni et al., 2022), using the same investment specifications, which are outlined in Table 5.1. Essentially, both the active portfolio and the benchmark portfolio commence with an initial wealth of 100 at time $t_0 = 0$. Monthly rebalancing is implemented for both portfolios over a 10-year investment horizon. Our empirical case allows the investment manager to allocate among four distinct investment assets: the equal-weighted stock index, the cap-weighted stock index, the 30-day U.S. T-bill index, and the 10-year U.S. T-bond index.

As in Ni et al. (2023), we choose the benchmark portfolio to be a constant weight portfolio with 70% in the equal-weighted stock and 30% in the 30-day T-bill index. By varying the outperformance target rates (β) across the range of 0.5% to 5% (incremental with a 0.5% step size), we obtain the corresponding optimal portfolios through the cumulative quadratic shortfall (CS) objective.

Investment horizon T (years)	10
Underlying assets	CRSP cap-weighted/equal-weighted index (real)
	CRSP 30-day/10-year U.S. treasury index (real)
Index samples for bootstrap	1926/01 to 2023/01
Initial portfolio wealth	100
Rebalancing frequency	Monthly
Cash injections	0
Benchmark portfolio	70% equal-weighted index/30% 30-day T-bill
Performance metric	Cumulative quadratic shortfall (CS)
Outperformance target rate β	0.5% - 5%, incremental by 0.5%

Table 5.1: Investment scenario.

In Table 5.2, we compare the performance of computed optimal 130/30 portfolios with that of the optimal long-only portfolios in outperforming the same elevated target across time on test data sets. In particular, the outperformance across time is reflected in the value of the CS objective, which measures the cumulative quadratic shortfall with respect to elevated targets defined by the target rate β.

β	0.5%	1%	1.5%	2%	2.5%	3%	3.5%	4%	4.5%	5%
130/30	275	402	622	985	1491	2434	3427	4782	6518	8709
Long-only	280	432	708	1150	1841	2855	4547	6430	9041	12594

Table 5.2: Cumulative quadratic shortfall (CS) objective function values from the trained models evaluated on $\boldsymbol{Y}_{test}$ for various β (a lower value is better.)

As we can observe from Table 5.2, even though the optimal long-only portfolio is obtained under the same investment specifications and optimized under the same objective function, it achieves significantly worse performance compared to the optimal 130/30 portfolio.

In particular, the gap between the CS objective function values widens as the target outperformance rate β increases, indicating that the long-only portfolio is further restricted by the long-only constraints as the outperformance target becomes more ambitious. This phenomenon is further demonstrated in Table 5.3, in which we list the median annual return of both portfolios.

β	0.5%	1%	1.5%	2%	2.5%	3%	3.5%	4%	4.5%	5%
130/30	7.2%	7.6%	8.1%	8.5%	8.9%	9.4%	9.7%	10.0%	10.2%	10.5%
Long-only	7.2%	7.6%	8.1%	8.5%	8.8%	8.9%	8.9%	8.9%	8.9%	8.9%

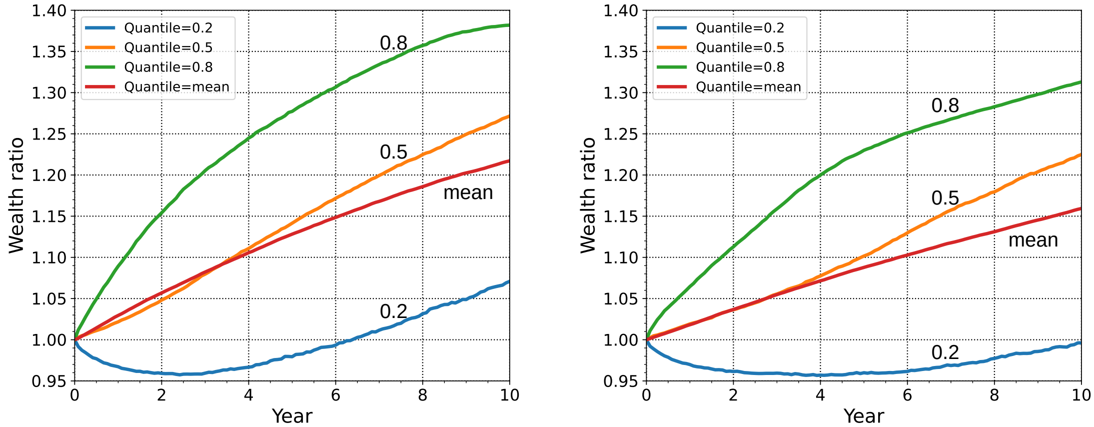
Table 5.3: Median annualized returns from the trained models evaluated on $\boldsymbol{Y}_{test}$ for various β. The benchmark portfolio has a median annualized return of 6.7%.

As we can see from Table 5.3, when β is modest, e.g., $\beta < 3\%$, the long-only portfolio shows similar median returns as the 130/30 portfolio (despite the fact that the objective function value is slightly worse). However, as β becomes more ambitious, e.g., $\beta \geq 3\%$, the long-only portfolio trails behind the 130/30 portfolio. Specifically, we can see that the median return of the long-only portfolio stagnates for $\beta \geq 3\%$.

We now examine in greater detail the performance comparison for $\beta = 3\%$. We plot the quantiles of the wealth ratio $\frac{W(t)}{\hat{W}(t)}$, which measures the relative performance of the active portfolio with respect to the benchmark portfolio throughout the investment horizon.

From Figure 5.1, it is evident that the 130/30 portfolio outperforms the long-only portfolio across various quantiles. The wealth ratios of the 130/30 portfolio consistently exceed those of the long-only portfolio, indicating superior performance.

Unsurprisingly, the enhanced performance of the 130/30 portfolio can be attributed to its relaxed portfolio constraints. We plot the median allocation fractions of the 130/30 portfolio and long-only portfolio in Figure 5.2. We can see from Figure 5.2a that the optimal 130/30 portfolio strategically leverages its position by exceeding 100% exposure to the equal-weighted stock index in the first half of the investment period. Interestingly, the 130/30 portfolio longs the equal-weighted stock and the long-term bond, and shorts the cap-weighted stock and the short-term bond, creating long/short patterns within both asset classes (i.e. stock and bond). On the other hand, as observed from Figure 5.2b, the long-only portfolio



(a) 130/30 portfolio (b) Long-only portfolio

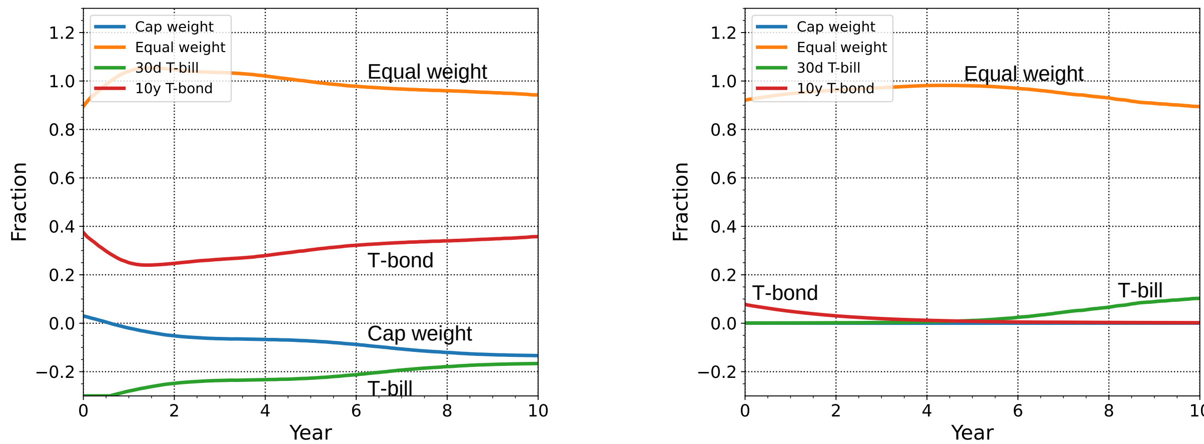
Figure 5.1: Quantiles of wealth ratio over the investment horizon $[0, T]$. $\beta = 3\%$. Outperformance is indicated when the wealth ratio is greater than one. The 130/30 portfolio follows the RCNN trained on $\boldsymbol{Y}$. The long-only portfolio follows the neural network model from (Li and Forsyth, 2019; Ni et al., 2022) trained on $\boldsymbol{Y}$. Results in the plots are testing results evaluated on $\boldsymbol{Y}_{test}$.

is obviously restricted by the long-only constraint. It yields an almost trivial strategy that has a close to 100% allocation to the equal-weighted stock index throughout the investment horizon.

We note that, when $\beta = 3\%$, the optimal long-only portfolio is already allocating almost 100% allocation to the equal-weighted stock index, the riskiest asset with the highest expected return. There is little room for the long-only portfolio to take more risks due to the long-only constraint to meet the more aggressive β values. On the other hand, we can see that the median return of the optimal 130/30 portfolio increases alongside β.

We remark that there is no free lunch, and the 130/30 strategy achieves superior results with some compromises. In particular, if we examine the extreme tail statistics such as the 1% CVaR[5] of the terminal wealth (i.e. the mean of the lowest 1% of the terminal wealth), we can see that the 130/30 portfolios have slightly worse results than the long-only portfolios, as shown in Table 5.4. This is because the 130/30 portfolios are leveraged and exposed to greater market risk and thus perform worse in consistent bear market scenarios.

[5]Note that we define CVaR in terms of the final wealth, not losses. Hence a larger CVaR is better.



(a) 130/30 portfolio (b) Long-only portfolio

Figure 5.2: Median allocation fractions over the investment horizon $[0, T]$. $\beta = 3\%$. The 130/30 portfolio follows the RCNN trained on $\boldsymbol{Y}$. The long-only portfolio follows the neural network model from (Li and Forsyth, 2019; Ni et al., 2022) trained on $\boldsymbol{Y}$. Results in the plots are testing results evaluated on $\boldsymbol{Y}_{test}$.

β	0.5%	1%	1.5%	2%	2.5%	3%	3.5%	4%	4.5%	5%
130/30	44	38	32	29	27	25	24	23	22	22
Long-only	44	39	36	35	34	33	32	32	32	32

Table 5.4: 1%-CVaR of terminal wealth (higher is better). CVaR here is the mean of the worst one percent of the outcomes. The results are based on the performance of trained models evaluated on $\boldsymbol{Y}_{test}$.

However, Table 5.4 might be regarded as an overly pessimistic risk comparison between the 130/30 strategy and the long-only strategy. Note that the initial portfolio value for both strategies is 100. Consider the case for $\beta = 5\%$ in Table 5.4. In this case, the 1% CVaR for the 130/30 strategy is 22, while it is 32 for the long-only policy. The long-only policy has a higher (better) CVaR, but both strategies yield extremely poor results, i.e. losses of 70-80% of the original investment after ten years. It is perhaps more instructive to examine the 20th quantile of the wealth ratio in Figure 5.1. We can observe that the optimal 130/30 portfolio exhibits better wealth ratios compared to the optimal long-only portfolio, in particular over the 2-10 year time period. This behavior is not surprising. In terms of the risk measure (5.4), the 130/30 strategy outperforms the long-only strategy, since the 130/30 policy has a larger admissible set of controls. This is reflected in the behavior of the 20th quantile in Figure 5.1. This suggests that the 130/30 portfolio is capable of mitigating downside risks in most scenarios. Of course, if the investor is really concerned primarily with left tail risk (relative to the benchmark) it is possible to construct an objective function based on a relative CVaR, but this is beyond the scope of this work.

Overall, the numerical experiments illustrate the enhanced performance of the 130/30 portfolio over the long-only portfolio using the across-time outperformance-based investment objective. The 130/30 portfolio not only achieves more ambitious returns but also demonstrates good risk management. This can be attributed to the broader range of portfolio strategies available within the 130/30 structure, which allows for more flexibility and potential for generating excess return and risk control.

Additionally, we note that the optimization algorithm automatically discovers which assets to long and which assets to short, based on achieving the best performance measure.

6 Conclusion

In this article, we introduced a neural network-based solution for the portfolio optimization problem under 130/30 constraints. By formulating the benchmark outperforming asset allocation problem as a constrained multi-period stochastic optimal control problem, we proposed a novel relaxed-constraint neural network (RCNN) model to compute the optimal control. By solving this constrained optimal control problem directly based on data samples without dynamic programming, we are able to compute optimal solutions to the high dimensional 130/30 allocation problems via a gradient-type method for unconstrained optimization.

By designing a tailored activation function that outputs feasible strategies, the proposed RCNN explicitly handles the original optimization problem with non-standard constraints that differentiate between long and short positions while imposing an upper limit on total long positions, thus converting it into an unconstrained problem that can be solved efficiently. We provided mathematical proof demonstrating that the RCNN can accurately approximate the leverage-constrained long/short strategy.

Based on directly resampled market returns, we compared the optimal 130/30 portfolio following the RCNN with the optimal long-only portfolio, with the objective of outperforming a benchmark portfolio. The results consistently indicate that the optimal 130/30 portfolio achieves enhanced performance over the long-only portfolio in terms of the chosen investment objective.

We believe the methodology developed in this article can be applied to investment problems of widespread interest, such as finding optimal portfolios of factor ETFs (Glushkov, 2015). In addition, in the future, it may be worthwhile to consider other types of securities such as options in the portfolio (Andersson and Oosterlee, 2023), which may yield better performance in practice.

A Gradient Calculation

Here we provide details of gradient computation, assuming a given training set. Consider the j-th sample path $Y^{(j)}$.

$$\nabla_{\boldsymbol{\theta}} F\left(\mathcal{W}_{\boldsymbol{\theta}}^{(j)}, \hat{\mathcal{W}}_{\hat{p}}^{(j)}\right) = \sum_{i=1}^{N} \frac{\partial F}{\partial W_{\boldsymbol{\theta}}^{(j)}(t_i)} \nabla_{\boldsymbol{\theta}} W_{\boldsymbol{\theta}}^{(j)}(t_i). \tag{A.1}$$

Let $\boldsymbol{R}(t_i) = (R_1(t_i), \cdots, R_1(t_i))^\top \in \mathbb{R}^{N_a}$ denote the return vector at t_i, then, the wealth dynamics for the value of the active portfolio described in (2.4) can be summarized as

$$W_{\boldsymbol{\theta}}^{(j)}(t_i) = f_{\boldsymbol{\theta}}\big(W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}), \hat{W}^{(j)}(t_{i-1}), t_{i-1}\big)^\top \big(1 + \boldsymbol{R}(t_i)\big) W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}) \mathbf{1}_{W_{\boldsymbol{\theta}}^{(j)}(t_{i-1})>0}, \tag{A.2}$$

where $f_{\boldsymbol{\theta}}$ is the RCNN parameterized by $\boldsymbol{\theta}$, and $\mathbf{1}_{W_{\boldsymbol{\theta}}^{(j)}(t_{i-1})>0}$ is a scalar indicator function.

Note that $\nabla_{\boldsymbol{\theta}} W_{\boldsymbol{\theta}}^{(j)}(t_0) = 0$, since the initial portfolio value is a constant value. Then, for any $i \in \{1, \cdots, N\}$, the gradients $\nabla_{\boldsymbol{\theta}} W_{\boldsymbol{\theta}}^{(j)}(t_i)$ in (A.1) can be obtained recursively using the chain rule, i.e.,

$$\nabla_{\boldsymbol{\theta}} W_{\boldsymbol{\theta}}^{(j)}(t_i) = \nabla_{\boldsymbol{\theta}} \Big(f_{\boldsymbol{\theta}}\big(W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}), \hat{W}^{(j)}(t_{i-1}), t_{i-1}\big)^\top \big(1 + \boldsymbol{R}(t_i)\big) W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}) \mathbf{1}_{W_{\boldsymbol{\theta}}^{(j)}(t_{i-1})>0} \Big) \tag{A.3}$$

$$\begin{aligned} &= \nabla_{\boldsymbol{\theta}} \Big(f_{\boldsymbol{\theta}}\big(W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}), \hat{W}^{(j)}(t_{i-1}), t_{i-1}\big)^\top \big(1 + \boldsymbol{R}(t_i)\big) \Big) W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}) \mathbf{1}_{W_{\boldsymbol{\theta}}^{(j)}(t_{i-1})>0} \\ &+ \Big(f_{\boldsymbol{\theta}}\big(W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}), \hat{W}^{(j)}(t_{i-1}), t_{i-1}\big)^\top \big(1 + \boldsymbol{R}(t_i)\big) \mathbf{1}_{W_{\boldsymbol{\theta}}^{(j)}(t_{i-1})>0} \Big) \nabla_{\boldsymbol{\theta}} W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}) \end{aligned} \tag{A.4}$$

$$\begin{aligned} &= \Big(\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}\big(W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}), \hat{W}^{(j)}(t_{i-1}), t_{i-1}\big) \Big) \big(1 + \boldsymbol{R}(t_i)\big) W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}) \mathbf{1}_{W_{\boldsymbol{\theta}}^{(j)}(t_{i-1})>0} \\ &+ \Big(\frac{\partial f_{\boldsymbol{\theta}}\big(W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}), \hat{W}^{(j)}(t_{i-1}), t_{i-1}}{\partial W_{\boldsymbol{\theta}}^{(j)}(t_{i-1})} \Big)^\top \big(1 + \boldsymbol{R}(t_i)\big) \mathbf{1}_{W_{\boldsymbol{\theta}}^{(j)}(t_{i-1})>0} \nabla_{\boldsymbol{\theta}} W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}) \\ &+ \Big(f_{\boldsymbol{\theta}}\big(W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}), \hat{W}^{(j)}(t_{i-1}), t_{i-1}\big)^\top \big(1 + \boldsymbol{R}(t_i)\big) \mathbf{1}_{W_{\boldsymbol{\theta}}^{(j)}(t_{i-1})>0} \Big) \nabla_{\boldsymbol{\theta}} W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}) \end{aligned} \tag{A.5}$$

B Stationary block bootstrap algorithm

Algorithm B.1 presents the pseudocode for the stationary block bootstrap. See Ni et al. (2022) for more discussion.

Algorithm B.1: Pseudocode for stationary block bootstrap

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

References

Alizadeh, A. H. and N. K. Nomikos (2007). Investment timing and trading strategies in the sale and purchase market for ships. *Transportation Research Part B: Methodological 41*(1), 126–143.

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

Andersson, K. and C. W. Oosterlee (2023). D-tipo: Deep time-inconsistent portfolio optimization with stocks and options. *arXiv preprint arXiv:2308.10556*.

Ang, I., A. Michalka, and A. Ross (2017). Understanding relaxed constraint equity strategies. Technical report, Working Paper, AQR Capital.

Black, F. (1993). Estimating expected return. *Financial Analysts Journal 49*(5), 36–38.

Brigo, D., A. Dalessandro, M. Neugebauer, and F. Triki (2008). A stochastic processes toolkit for risk management. *arXiv preprint arXiv:0812.4210*.

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

Carney, W. J. (1998). Limited liability. *Encyclopedia of law and economics*. Available at SSRN 50563.

Cavaglia, S., L. Scott, K. Blay, and S. Hixon (2022). Multi-asset class factor premia: A strategic asset allocation perspective. *The Journal of Portfolio Management 48*(4), 14–32.

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

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

Dichtl, H., W. Drobetz, and L. Kryzanowski (2016). Timing the stock market: Does it really make no sense? *Journal of Behavioral and Experimental Finance 10*, 88–104.

Easterbrook, F. H. and D. R. Fischel (1985). Limited liability and the corporation. *52 Unversity of Chicago Law Review*, 89.

Federal Reserve Board (1974). Electronic code of federal regulations. `https://www.ecfr.gov/current/title-12/chapter-II/subchapter-A/part-220`. [Online; accessed 21-Oct-2023].

Fung, W. and D. A. Hsieh (1999). A primer on hedge funds. *Journal of empirical finance 6*(3), 309–331.

Gastineau, G. L. (2008). The short side of 130/30 investing for the conservative portfolio manager. *Journal of Portfolio Management 34*(2), 39.

Gay, C. (2012). Leverage, volatility, and the curious case of 130/30 funds. `https://finance.yahoo.com/news/leverage-volatility-curious-case-130-160221697.html`. [online, Yahoo Finance].

Glushkov, D. (2015). How smart are 'smart beta' ETFs? analysis of relative performance and factor exposure. *Available at SSRN 2594941*.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep learning*. MIT press.

Han, J. et al. (2016). Deep learning approximation for stochastic control problems. *arXiv preprint arXiv:1611.07422*.

Johnson, G., S. Ericson, and V. Srimurthy (2007). An empirical analysis of 130/30 strategies. *Journal of Alternative Investments 10*(2), 31–42.

Johnson, S. (2013). The decline, fall and afterlife of 130/30. `https://www.ft.com/content/fdbf6284-b724-11e2-841e-00144feabdc0`. [online, Financial Times].

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Korhonen, J. A. O. and D. Kunz (2010). 130/30 investment strategies–is the active extension value adding? Master Thesis, Lund Unversity School of Economics and Management.

Kou, S. G. (2002). A jump-diffusion model for option pricing. *Management science 48*(8), 1086–1101.

Kratsios, A. and I. Bilokopytov (2020). Non-euclidean universal approximation. *Advances in Neural Information Processing Systems 33*, 10635–10646.

Krusen, C., F. Weber, and R. A. Weigand (2008). 130/30 funds: The evolution of active equity investing. *Special Issues 2008*(1), 176–185.

Leibowitz, M. L., S. Emrich, and A. Bova (2009). *Modern portfolio management: active long/short 130/30 equity strategies*. John Wiley & Sons.

Li, D. and W.-L. Ng (2000). Optimal dynamic portfolio selection: Multiperiod mean-variance formulation. *Mathematical finance 10*(3), 387–406.

Li, X., X. Y. Zhou, and A. E. Lim (2002). Dynamic mean-variance portfolio selection with no-shorting constraints. *SIAM Journal on Control and Optimization 40*(5), 1540–1555.

Li, Y. and P. A. Forsyth (2019). A data-driven neural network approach to optimal asset allocation for target based defined contribution pension plans. *Insurance: Mathematics and Economics 86*, 189–204.

Lo, A. W. and P. N. Patel (2008). 130/30: The new long-only. *Special Issues 2008*(1), 186–211.

Lu, Y. and J. Lu (2020). A universal approximation theorem of deep neural networks for expressing probability distributions. *Advances in neural information processing systems 33*, 3094–3105.

McCrary, S. A. (2004). *Hedge fund course*. John Wiley & Sons.

Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of financial economics 3*(1-2), 125–144.

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

Ni, C., Y. Li, and P. A. Forsyth (2023). Optimal asset allocation in a high inflation regime: a leverage-feasible neural network approach. *arXiv preprint arXiv:2304.05297*.

Politis, D. N. and J. P. Romano (1994). The stationary bootstrap. *Journal of the American Statistical Association 89*(428), 1303–1313.

Rekenthaler, J. (2022). Nobody likes index funds, except investors. Morningstar Inc. https://www.morningstar.com/articles/1096069/nobody-likes-index-funds-except-investors.

Reppen, A. M., H. M. Soner, and V. Tissot-Daguette (2023). Deep stochastic optimization in finance. *Digital Finance 5*(1), 91–111.

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

Shahzad, S. J. H., E. Bouri, D. Roubaud, L. Kristoufek, and B. Lucey (2019). Is bitcoin a better safe-haven investment than gold and commodities? *International Review of Financial Analysis 63*, 322–330.

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

Tabb, L. and J. Johnson (2007). Alternative investments 2007: The quest for alpha. https://research.tabbgroup.com/report/v05-008-alternative-investments-2007-quest-alpha. [Technical report, Tabb Group].

Tsang, K. H. and H. Y. Wong (2020). Deep-learning solution to portfolio selection with serially dependent returns. *SIAM Journal on Financial Mathematics 11*(2), 593–619.

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

van Staden, P. M., P. A. Forsyth, and Y. Li (2024). Across-time risk-aware strategies for outperforming a benchmark. *European Journal of Operational Research 313:2*, 776–800.

Yoon, J., D. Jarrett, and M. Van der Schaar (2019). Time-series generative adversarial networks. *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*. https://proceedings.neurips.cc/paper/2019/file/c9efe5f26cd17ba6216bbe2a7d26d490-Paper.pdf.

Zhou, X. Y. and D. Li (2000). Continuous-time mean-variance portfolio selection: A stochastic LQ framework. *Applied Mathematics and Optimization 42*, 19–33.