Optimal Multi-period Leverage-Constrained Portfolios: a Neural Network Approach

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

September 30, 2024

Abstract

We present a neural network approach for multi-period portfolio optimization that relaxes the long-only restriction and instead imposes a bound constraint on leverage. We formulate the optimization problem for such a relaxed-constraint portfolio as a multi-period stochastic optimal control problem. We propose a novel relaxed-constraint neural network (RCNN) model to approximate the optimal control. Using our proposed RCNN model transforms the original leverage-constrained optimization problem into an unconstrained one, which makes solving it computationally more feasible. We prove mathematically that the proposed RCNN control model can approximate the optimal relaxed-constraint strategy with arbitrary precision. We further propose to compute the optimal outperforming strategy over a benchmark based on cumulative quadratic shortfall (CS). Using U.S. historical market data from Jan 1926 to Jan 2023, we computationally compare and assess the proposed neural network approach to the optimal leverage-constrained strategy and long-only strategy respectively. We demonstrate that the leverage-constrained optimal strategy can achieve enhanced performance over the long-only strategy in outperforming a benchmark portfolio.

1 Introduction

Traditionally, most mutual fund portfolios operate under a long-only strategy. This means that if a security is perceived as undervalued, it can be included in the portfolio. Conversely, if a security is considered overvalued, to capture potentially additional alpha, investors can only choose to avoid investing in it rather than actively shorting it.

To address these limitations, *relaxed-constraint portfolios*, which permit some chosen level of leverage in contrast to long-only, have emerged (Ang et al., 2017). These portfolios enable managers to short sell securities considered to be overvalued, while maintaining a net exposure to the market of 100%. By shorting some securities and using the proceeds to invest in other securities, this approach introduces leverage into the portfolio. Subject to internal risk mandates and regulatory requirements (Federal Reserve Board, 1974), these portfolios typically cap the total leverage, which can be expressed as imposing an upper bound on the total long positions. For instance, the popular 130/30 portfolio allows investors to hold short positions totalling up to 30% of the portfolio's net wealth (or equivalently the total long position is bounded below 130%) (Lo and Patel, 2008).

While there are clear incentives for adopting the relaxed-constraint portfolios, the literature on the topic of portfolio optimization for such strategies, particularly in the context of multi-period setting, remains scarce. 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 restrictions of relaxed-constraint strategies, which caps the total leverage allowed in the portfolio.

Consequently, many fund managers had to rely on less rigorous approaches, such as ranking systems (Leibowitz et al., 2009; Korhonen and Kunz, 2010), to construct their relaxed-constraint portfolios. These challenges perhaps explain why there is little empirical evidence that relaxed-constraint portfolios brings superior risk and return profiles than long-only portfolios (Johnson, 2013).

Leverage constrained portfolio optimization separates long positions from short positions and impose constraints on the total long position and total short position accordingly. This leads to an optimization problem that a typical method cannot be immediately applied, since it usually assumes a standard formulation expressed by a continuous objective function and equality and inequality constraint functions.

Given the scarcity of literature on the multi-period optimization of relaxed-constraint portfolios, we aim to bridge this gap by providing a novel portfolio optimization framework that addresses the specific challenges posed by this leverage constraint. Particularly, we propose to use a neural network model to approximate the optimal relaxed-constraint 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); Li and Mulvey (2021); van Staden et al. (2023); Ni et al. (2022, 2024).

Notably, Li and Mulvey (2021) use recurrent neural networks to model the upper and lower bounds of asset allocations at each timestep. They demonstrate that this approach allows the multiperiod optimization problem to be solved in polynomial time, rather than exponential time, with respect to the number of rebalancing periods and risky assets, thus addressing the curse of dimensionality issue often encountered in traditional numerical methods (Pun and Wong, 2019; Li et al., 2022).

However, Han et al. (2016); Tsang and Wong (2020); Li and Mulvey (2021) consider a stacked neural network approach which uses a different subnetwork at every rebalancing time, which is still computationally expensive. 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.

When computing a neural network model for portfolio optimization, it is desirable to incorporate constraints by designing the neural network control model in a way that satisfies constraints explicitly, since this leads to a training optimization problem which can be readily solved by a stochastic gradient method. One common approach is to use the softmax activation function in the output layer of the network, ensuring that the output allocation fractions are non-negative and summing 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 function via using appropriate activation functions, gradient-based optimization algorithms like stochastic gradient descent (SGD) can be applied effectively (Buehler et al., 2019).

However, for a leverage-constrained portfolio, which limits the total long (and short) position, it is not immediately clear how to design such a neural network model which explicitly satisfies the required constraints. The closest work is the proposed methodology in Ni et al. (2024), in which the authors consider the multi-period portfolio optimization problem where the portfolio also allows bounded leverage. However, in Ni et al. (2024), it is assumed that the manager can only short a specific pre-determined subset of the universe of securities, whereas in this work we allow the manager to short any security in the entire portfolio universe.

To address this, we propose a novel *relaxed-constraint neural network* (RCNN) control model that specifically satisfies the relaxed-constraint portfolio restrictions. By designing the neural network model with appropriate activation functions, we convert the leverage constrained stochastic optimization problem into an unconstrained optimization problem, which is more computationally feasible to solve. Furthermore, we mathematically prove that the RCNN control model can approximate any optimal relaxed-constraint strategy arbitrarily well, implying that solving the unconstrained optimization problem can yield sufficiently accurate approximation to the optimal relaxed-constraint strategy.

In practice, relaxed-constraint portfolios are considered as part of the long-only portfolio family and are typically evaluated based on their relative performance over a passive benchmark portfolio. To achieve benchmark outperformance, we choose a cumulative quadratic shortfall (CS) objective function that measures the tracking difference of the active portfolio against a benchmark portfolio.

We emphasize that the RCNN is flexible and applicable to diverse investment objective functions. As long as standard optimization methods can backpropagate through the chosen objective function, our proposed approach can be applied to a wide range of investment problems with ease.

Using the proposed neural network approach and based on historical market data, we assess and compare performance of the optimal relaxed-constraint portfolio with to the optimal long-only portfolio under the same investment scenario and the CS objective. Our computational results demonstrate clear advantages of the relaxed-constraint strategy, showcasing superior returns and improved risk management outcomes, which empirically validates the effectiveness of our proposed RCNN approach.

The main contributions of this article are summarized below.

(i) We propose a novel relaxed-constraint neural network (RCNN) control model, so that the otherwise challenging constrained multi-period optimization problem for the relaxed-constrained portfolio can be computationally solved by applying an algorithm for unconstrained optimization.

(ii) We mathematically prove that the proposed RCNN control model is capable of approximating any relaxed-constraint strategy arbitrarily well. This proof serves as a theoretical foundation, validating the efficacy of our proposed methodology.

(iii) While the proposed neural network approach is computationally flexible and applicable to any general continuous objective function, we propose to compute the optimal outperforming strategy to overcome a benchmark based on cumulative quadratic shortfall (CS) under a leverage constraint, which is relaxed over long-only constraint.

(iv) Through computational assessment based on the U.S. market data from Jan 1926 to Jan 2023, we provide evidence of the advantages of relaxed-constraint portfolios over traditional long-only portfolios. Our findings are contrary to the commonly held view that relaxed-constraint portfolios yield few benefits for investors over long-only portfolios.

Subsequently, in §2, we first mathematically formulate a general multi-period stochastic optimal control problem for optimal leveraged portfolio under a relaxed-constraint. In §3, we describe the proposed RCNN control model for handling leverage constraints. We establish a universal approximation theorem for the proposed RCNN in §4. In §5, we motivate our choice of the cumulative quadratic shortfall (CS) objective function to achieve benchmark outperformance. In addition, using the proposed neural network approach, we present computational comparison and assessment of the optimal strategies based on market data. Finally concluding remarks are given in §6.

2 Mathematical formulation

In this section, we mathematically formulate the multi-period portfolio optimization problem for relaxed-constraint portfolios.

Relaxed-constraint portfolios are considered as part of the extended family of long-only portfolios and are thus often assessed against a passive benchmark (Ang et al., 2017). Therefore, we consider two portfolios: an actively managed portfolio and a benchmark portfolio.

We consider a fixed investment horizon $[t_0, T]$. At any 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 $\boldsymbol{S}(t) = (S_i(t) : i = 1, \cdots, N_a)^\top \in \mathbb{R}^{N_a}$ denote prices of the N_a underlying assets at time $t \in [t_0, T]$. In addition, let vectors $p^{(t)} = (p_i^{(t)} : i = 1, \cdots, N_a)^\top \in \mathbb{R}^{N_a}$ and $\hat{p}^{(t)} = (\hat{p}_i^{(t)} : i = 1, \cdots, N_a)^\top \in \mathbb{R}^{N_a}$ denote the allocation fractions to the N_a underlying assets at time $t \in [t_0, T]$ respectively, for the active portfolio and the benchmark portfolio. In this article, we consider a passive benchmark portfolio with constant allocation, i.e., $\hat{p}^{(t)} \equiv \hat{p}^{(0)}, \ \forall t \in [0, T]$, where $\hat{p}^{(0)}$ is a constant vector that represents the pre-defined allocation fractions to respective assets.

From a stochastic optimal control perspective, the allocation vector $p^{(t)}$ is regarded as the control value at time t, which determines the outcome of the system, i.e., the evolution of the portfolio values, for a given realization of the environment. The control vector $p^{(t)}$ is assumed to be a function of the state variables that fully describe the state of the dynamic system at time t. It is shown that under common assumptions of the asset prices, such as jump-diffusion processes, the state variables are simply the portfolio values and time (Dang and Forsyth, 2014). While we consider the case of the portfolio values and time as state variables in this article, incorporating additional factors as state variables poses no technical challenges for the proposed methodology. Mathematically, $p^{(t)} = p(\boldsymbol{X}(t)) = (p_i(\boldsymbol{X}(t)) : i \in \{1, \cdots, N_a\})^\top \in \mathbb{R}^{N_a}$, 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 so that some chosen relative performance measure of the active portfolio over the benchmark portfolio is maximized.

In addition, 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 relaxed-constraint strategies

In practice, a permissible relaxed-constraint portfolio needs to satisfy some specific constraints, e.g., a bound on leverage. In this section, we mathematically define the feasible relaxed-constraint strategies.

Definition 2.1. (Feasible relaxed-constraint strategies). *A strategy $p : \mathcal{X} \mapsto \mathbb{R}^{N_a}$ is a feasible relaxed-constraint strategy if and only if*

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

where $\mathcal{Z} \subset \mathbb{R}^{N_a}$ encodes the portfolio 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 $p_{max} \geq 1$ is a given constant.

Furthermore, $\mathcal{A}$ denotes the set of all feasible strategies, i.e., $\mathcal{A} = \{p : Im(p) \subseteq \mathcal{Z}\ \}$.

Remark 2.1. (Financial meaning of p_{max}). p_{max} is the maximum total long position of the portfolio. For example, the 130/30 portfolios use $p_{max} = 1.3$. Note that setting p_{max} is the same as setting a limit on the total leverage of the portfolio, since the leverage is calculated based on the amount of debt (short position) raised in the portfolio. In particular, if $p_{max} = 1$, no leverage is allowed.

2.2 Stochastic optimal control problem

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

For an active portfolio which allows for both long and short positions, there is a theoretical possibility for the value of the portfolio to become negative. In such circumstances, the fund would initiate a bankruptcy process, resulting in the settlement of outstanding liabilities and the cessation of future trading activities. From a mathematical perspective, the portfolio value remains at zero throughout the remainder of the investment horizon. In addition, for simplicity, we do not consider subsequent cash injections after the initial investment. Consequently, the evolution of the portfolio values can be described as follows from the perspective of an investor in the limited-liability fund:

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

Let sets $\mathcal{W}_p = \{W(t), t \in \mathcal{T}\}$ and $\hat{\mathcal{W}}_{\hat{p}} = \{\hat{W}(t), t \in \mathcal{T}\}$ represent the trajectories of the portfolio values for the active portfolio and the benchmark portfolio respectively, following the dynamics specified in equation (2.4). We introduce an investment performance metric denoted by $F(\mathcal{W}_p, \hat{\mathcal{W}}_{\hat{p}}) \in \mathbb{R}$, which quantifies the relative performance of the active portfolio in relation to the benchmark portfolio based on their respective value trajectories. In this article, we assume the asset prices $\boldsymbol{S}(t) \in \mathbb{R}^{N_a}$ are stochastic. Consequently, the value trajectories $\mathcal{W}_p$, $\hat{\mathcal{W}}_{\hat{p}}$, and the performance metric $F(\mathcal{W}_p, \hat{\mathcal{W}}_{\hat{p}})$ are also stochastic.

When investment managers aim to optimize an investment performance, the assessment commonly involves evaluating the expectation of a random metric. Let $\mathbb{E}_p^{(t_0,w_0)}[F(\mathcal{W}_p, \hat{\mathcal{W}}_{\hat{p}})]$ denote the expectation of the performance metric F, given a specific initial (cash injection) value $w_0 = W(0) = \hat{W}(0)$ at time $t_0 = 0$. The expectation is evaluated on random wealth trajectory following an admissible investment strategy $p \in \mathcal{A}$ and the benchmark investment strategy $\hat{p}$. Since we assume the benchmark strategy to be predetermined and known, we keep the benchmark strategy $\hat{p}$ notationally implicit for simplicity. Subsequently, we try to solve the following stochastic optimization (SO) problem:

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

The choice of $F(\cdot)$ depends on specifying investment goals appropriate performance assessment metrics. One of the advantages of our proposed approach is its applicability to any function F (ideally continuously differentiable) and computational feasibility for high dimensional problems, even under some constraints.

Solving the constrained stochastic optimal control problem (2.5) is challenging when the feasible set $\mathcal{A}$ corresponds to the intricate leverage constraint (2.2)&(2.3). Subsequently we first focus on addressing this challenge for a general optimal relaxed-constraint problem (2.5) by proposing a neural network approach that circumvents the complexity of handling this constraint through introduction of a specially designed relaxed-constraint neural network (RCNN) model. In §5, we motivate the cumulative quadratic shortfall as a suitable choice of the objective function in outperforming a benchmark and assess computationally performance of the corresponding optimal strategy.

3 Relaxed-constraint neural network (RCNN)

In this section, we describe proposed neural network approach for solving the stochastic optimization problem (2.5) for relaxed-constraint portfolios described in (2.2) & (2.3). In order to efficiently handle these nonstandard constraints, our key idea is to approximate the optimal control function using a neural network activation function that automatically satisfies the feasibility constraint (2.2).

Specifically, 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,

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

and this neural network itself is a feasible relaxed-constraint strategy, i.e., $f_{\boldsymbol{\theta}} \in \mathcal{A}$, where $\mathcal{A}$ is the set of relaxed-constraint strategies described in Definition 2.1. Using such a neural network, the original constrained optimization problem (2.5) can be converted to the following unconstrained optimization problem, which can readily be solved computationally using optimization methods for unconstrained optimization,

$$\text{(Unconstrained optimization problem):} \quad \inf_{\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}} \mathbb{E}_{f_{\boldsymbol{\theta}}}^{(t_0, w_0)} \big[F(\mathcal{W}_{\boldsymbol{\theta}}, \hat{\mathcal{W}}_{\hat{p}}) \big]. \tag{3.2}$$

Here $\mathcal{W}_{\boldsymbol{\theta}}$ is the wealth trajectory of the active portfolio with control following the neural network approximation function $f_{\boldsymbol{\theta}}(\boldsymbol{X}(t))$ parameterized by $\boldsymbol{\theta}$.

To design such a neural network model, we first define the commonly used fully connected feedforward neural network (Lu and Lu, 2020) as follows.

Definition 3.1. (Fully connected feedforward neural network $\tilde{f}_{\boldsymbol{\theta}}$). *A fully connected feedforward neural network (FNN) maps an input vector $\boldsymbol{x} \in \mathbb{R}^{d_0}$ to an output vector $\boldsymbol{h} \in \mathbb{R}^{d_{K+1}}$, where FNN 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.*

$$\begin{cases} \boldsymbol{x}^{(0)} = \boldsymbol{x}, \\ \boldsymbol{x}^{(k)} = \sigma\Big(\big(\boldsymbol{\theta}^{(k)}\big)^\top \cdot \boldsymbol{x}^{(k-1)} + \boldsymbol{\theta}_b^{(k)} \Big), 1 \le k \le K, \\ \boldsymbol{h} = \big(\boldsymbol{\theta}^{(K+1)}\big)^\top \cdot \boldsymbol{x}^{(K)} + \boldsymbol{\theta}_b^{(K+1)}. \end{cases} \tag{3.3}$$

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 assemble all weight matrices and bias vectors into a single parameter vector $\boldsymbol{\theta} = (\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}_b^{(1)}, \cdots, \boldsymbol{\theta}^{(K+1)}, \boldsymbol{\theta}_b^{(K+1)})^\top \in \mathbb{R}^{N_{\boldsymbol{\theta}}}$, where $N_{\boldsymbol{\theta}} = \sum_{k=1}^{K+1} (d_{k-1} \cdot d_k + d_k)$. Furthermore, we use the 2-tuple $\Big(K, (d_1, \cdots, d_K)^\top\Big)$ to denote the hyperparameters, i.e., the number of hidden layers and the sizes of each hidden layer.

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

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

To achieve feasibility explicitly, we propose the following relaxed-constraint activation function which is applied to the feedforward neural network $\tilde{f}_{\boldsymbol{\theta}}$.

Definition 3.2. (Relaxed-constraint activation function). *We define the "relaxed-constraint activation function", $\phi : \mathbb{R}^{N_a - 1} \mapsto \mathbb{R}^{N_a}$, as*

$$\phi = \phi_3 \circ \phi_2 \circ \phi_1, \tag{3.4}$$

i.e., the relaxed-constraint activation function ϕ is a composition of ϕ_3, ϕ_2 and ϕ_1, where ϕ_1, ϕ_2 and ϕ_3 are defined as follows:

bounded mapping $\phi_1 : \mathbb{R}^{N_a - 1} \mapsto \mathbb{R}^{N_a - 1}$. *Given a constant $\alpha \in \mathbb{R}$, $\alpha \neq \frac{1}{2}$, for any $\boldsymbol{h} = (h_1, \cdots, h_{N_a - 1})^\top \in \mathbb{R}^{N_a - 1}$,*

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

Here $\sigma : \mathbb{R}^{N_a-1} \mapsto \mathbb{R}^{N_a-1}$ is the pointwise sigmoid function, i.e. $[\sigma(\boldsymbol{h})]_i = \sigma(h_i)$. In essence, 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 $\alpha < \frac{1}{2}$.[1]

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

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

In other words, 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.

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

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

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 $\boldsymbol{v}$. Namely, 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, we define the relaxed-constraint neural network (RCNN) as follows.

Definition 3.3. (Relaxed-constraint neural network). *Let $\mathcal{X} \subset \mathbb{R}^3$ be the state space. Given hyperparameters $\big(K, (d_1, \cdots, d_K)^\top\big)$ (i.e. number of hidden layers and their sizes), and parameter $\boldsymbol{\theta}$, let $\tilde{f}_{\boldsymbol{\theta}} : \mathcal{X} \mapsto \mathbb{R}^{N_a-1}$ be the fully connected feedforward neural network (FNN) function parameterized by $\boldsymbol{\theta}$ as defined in Definition 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*

$$f_{\boldsymbol{\theta}} = \phi \circ \tilde{f}_{\boldsymbol{\theta}}. \tag{3.8}$$

We first establish the following lemma to show that the RCNN function defined in Definition 3.3 is a feasible strategy that satisfies the constraints defined in Definition 2.1.

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, $f_{\boldsymbol{\theta}}$ is a feasible strategy under the relaxed constraints, as described in Definition 2.1.*

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

Proof. According to the relaxed constraints in Definition 2.1, it is sufficient to show that

$$Im(f_{\boldsymbol{\theta}}) \subseteq \mathcal{Z}, \tag{3.9}$$

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

Let $\boldsymbol{y} = (y_1, \cdots, y_{N_a})^\top = f_{\boldsymbol{\theta}}(\boldsymbol{x})$, $\forall \boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^3$, where $\tilde{f}_{\boldsymbol{\theta}}$ is the FNN in Definition 3.1. To show (3.9), it is sufficient to show that

$$\boldsymbol{y} \in \mathcal{Z}, \tag{3.10}$$

which is equivalent to

$$\begin{cases} \sum_{i=1}^{N_a} y_i = 1, \\ \sum_{i=1}^{N_a} (y_i)^+ \leq p_{max}. \end{cases} \tag{3.11}$$

Let ϕ_1, ϕ_2 and ϕ_3 be the bounded mapping, extension mapping, and scaling mapping in Definition 3.2. Let $\boldsymbol{h} = \tilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $\boldsymbol{v} = \phi_2(\phi_1(\boldsymbol{h})) \in \mathbb{R}^{N_a}$. Then $\boldsymbol{y} = \phi_3(\boldsymbol{v})$. Note that $\sum_{i=1}^{N_a} v_i = 1$ (due to 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

$$\boldsymbol{y} = \phi_3(\boldsymbol{v}) = (\boldsymbol{v})^+ \cdot \frac{p_{max}}{\sum_{i=1}^{N_a} (v_i)^+} + (\boldsymbol{v})^- \cdot \frac{1 - p_{max}}{1 - \sum_{i=1}^{N_a} (v_i)^+}. \tag{3.12}$$

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

$$\begin{cases} (\boldsymbol{y})^+ = (\boldsymbol{v})^+ \cdot \frac{p_{max}}{\sum_{i=1}^{N_a} (v_i)^+}, \\ (\boldsymbol{y})^- = (\boldsymbol{v})^- \cdot \frac{1-p_{max}}{1-\sum_{i=1}^{N_a} (v_i)^+}. \end{cases} \tag{3.13}$$

Then, we have

$$\begin{cases} \sum_{i=1}^{N_a} (y_i)^+ = \frac{p_{max}}{\sum_{i=1}^{N_a} (v_i)^+} \cdot \left(\sum_{i=1}^{N_a} (v_i)^+ \right) = p_{max} \leq p_{max}, \\ \sum_{i=1}^{N_a} (y_i)^- = \frac{1-p_{max}}{1-\sum_{i=1}^{N_a} (v_i)^+} \cdot \left(\sum_{i=1}^{N_a} (v_i)^- \right) = \frac{1-p_{max}}{\sum_{i=1}^{N_a} (v_i)^-} \cdot \left(\sum_{i=1}^{N_a} (v_i)^- \right) = 1 - p_{max}, \end{cases} \tag{3.14}$$

and

$$\sum_{i=1}^{N_a} y_i = \sum_{i=1}^{N_a} (y_i)^+ + \sum_{i=1}^{N_a} (y_i)^- = 1. \tag{3.15}$$

Therefore, both conditions in (3.11) are satisfied, thus concluding the proof. □

Remark 3.1. (Intuition behind the RCNN design). As shown in Definition 3.3, the proposed RCNN function is constructed by applying the relaxed-constraint activation function ϕ (Definition 3.2) on top of a FNN (Definition 3.1). The FNN provides the approximation power by connecting several hidden layers via the sigmoid activation functions. The relaxed-constraint activation function ϕ, on the other hand, guarantees that the RCNN function satisfies the relaxed constraints. Particularly, recall the three mappings ϕ_1, ϕ_2 and ϕ_3 in Definition 3.2. ϕ_1 maps the output region of the FNN (which can be any point in $\mathbb{R}^{N_a-1}$) into a bounded region of $(1-\alpha,\alpha)^{N_a-1}$, if $\alpha > \frac{1}{2}$ or $(\alpha, 1-\alpha)^{N_a-1}$, if $\alpha < \frac{1}{2}$. Intuitively, the output of ϕ_1 represents an initial estimate of the allocation fraction for the first $N_a - 1$ assets. Due to the maximum total long position of $p_{max} > 1$, any feasible allocation fraction for each asset falls into $[1-p_{max}, p_{max}]$. Therefore, we choose α to be slightly larger than p_{max} (in computational study, 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 that the RCNN function can approximate the optimal relaxed-constraint 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 relaxed-constraint strategy. However, our computational results suggest that applying ϕ_1 leads to a faster convergence in the training of the neural network.

Next we show that the mapping ϕ has a right inverse if $\alpha > p_{max}$, which is necessary to demonstrate that the proposed RCNN can generate any feasible strategy. This property is also needed for establishing convergence of RCNN as mentioned in Remark 3.1.

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

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

$$Im(\phi) \subseteq \mathcal{Z}. \tag{3.16}$$

Therefore, $y_i \in [1-p_{max}, p_{max}], \forall i \in \{1, \cdots, N_a\}$. Then,

$$\frac{y_i - 1 + \alpha}{2\alpha - 1} \in \Big[\frac{\alpha - p_{max}}{2\alpha - 1}, \frac{\alpha + p_{max} - 1}{2\alpha - 1}\Big] \subset \Big(\frac{0}{2\alpha-1}, \frac{2\alpha - 1}{2\alpha - 1}\Big) = (0,1). \tag{3.17}$$

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

$$\overrightarrow{\phi}(\boldsymbol{y}) = \Big(\sigma^{-1}\big(\frac{y_1 - 1 + \alpha}{2\alpha - 1}\big), \cdots, \sigma^{-1}\big(\frac{y_{N_a-1} - 1 + \alpha}{2\alpha - 1}\big)\Big)^\top, \tag{3.18}$$

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

$$\phi\big(\overrightarrow{\phi}(\boldsymbol{y})\big) = \boldsymbol{y}. \tag{3.19}$$

□

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

$$\text{(Unconstrained parameterized problem):} \quad \inf_{\boldsymbol{\theta}\in\mathbb{R}^{N_{\boldsymbol{\theta}}}} \mathbb{E}_{f_{\boldsymbol{\theta}}}^{(t_0,w_0)}\big[F(\mathcal{W}_{\boldsymbol{\theta}}, \hat{\mathcal{W}}_{\hat{p}})\big]. \tag{3.20}$$

An essential question remains unanswered: for the optimal relaxed-constraint strategy p^*, is it possible to find a 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 (3.20) can yield a sufficiently accurate approximation of the optimal relaxed-constraint strategy. To address this crucial question, we establish an approximation theorem in the following section, providing a formal proof of the existence of such approximations. This theorem theoretically justifies effectiveness of the neural network methodology for approximating the optimal relaxed-constraint strategy.

4 Universal approximation theorem for RCNN

Before we prove the approximation theorem for the RCNN, we first present some mild assumptions.

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 common and intuitive.

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 $\big(K, (d_1, \cdots, d_K)^\top\big)$ 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

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

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) whose output values form 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 of which the image almost covers $\mathcal{Y}$.

We then proceed to present the approximation theorem for the RCNN.

Theorem 4.1. *(Approximation of optimal relaxed-constraint strategy). Assume that the constant α in Definition 3.2 satisfies $\alpha > p_{max}$ and Assumption 4.1 holds. Given the optimal control p^*, $\forall \epsilon > 0$, there exists $\Big(K, (d_1, \cdots, d_K)^\top\Big)$, and $\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}$ such that the corresponding RCNN $f_{\boldsymbol{\theta}}$ defined 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, $\forall \boldsymbol{z} = (z_1, \cdots, z_{N_a})^\top \in \mathcal{Z}$, $z_i \in [1 - p_{max}, p_{max}], \forall i \in \{1, \cdots, N_a\}$. In addition, ϕ is continuous and has a right-inverse $\overrightarrow{\phi}$, following Lemma 3.2. Hence $\overrightarrow{\phi}(\boldsymbol{z})$ is well-defined and $\phi(\overrightarrow{\phi}(\boldsymbol{z})) = \boldsymbol{z}$ (see also (3.17)). Therefore,

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

Combine (4.4) with (4.3), $Im(\phi) = \mathcal{Z}$, and thus $Im(\phi)$ is dense in $\mathcal{Z}$.

Applying Lemma 4.1, 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 constraints (2.2), there exists hyperparameters and parameter values that enables the corresponding RCNN to approximate the control arbitrarily well. Consequently, when the RCNN is sufficiently large in terms of the number of hidden nodes, solving the unconstrained optimization problem (3.20) results in an approximate solution that closely approximates the optimal control p^*.

5 Performance assessment of optimal strategies

Historically, relaxed-constraint portfolios have achieved a meagre advantage, if any, over their long-only counterparts. For example, around 2007, the concept of the 130/30 portfolio gained much popularity (Johnson et al., 2007; Gastineau, 2008; Lo and Patel, 2008; Krusen et al., 2008)). However, reports indicate that even when compared to long-only portfolios, which the 130/30 portfolios were designed to replace, they did not demonstrate superior returns (Johnson, 2013). This is counter-intuitive from a mathematical standpoint, since relaxed-constraint portfolios theoretically offer a larger solution space than long-only portfolios due to the relaxed portfolio constraint.

In this section, we computationally compare performance of an optimal relaxed-constraint portfolio with that of an optimal long-only portfolio, when both portfolios are optimized under the same investment objective function, with the only difference being the portfolio constraints.

Particularly, we use the 130/30 portfolio as an example of the relaxed-constraint portfolio. However, the methodology can be readily applied to other relaxed-constraint portfolios.

We conduct computational assessment for the optimal RCNN strategy and long-only strategy based on historical market data. This computational investigation requires solving problem (3.20) and evaluating performance of the RCNN model associated with the computed optimal parameters $\boldsymbol{\theta}^*$.

Next we first motivate the objective function we choose for outperforming a benchmark. In addition we provide a brief overview on how to compute the optimal solution.

5.1 Investment objective

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

$$IR_p^{(t_0,w_0)} = \frac{\mathbb{E}_p^{(t_0,w_0)}\left[W(T) - \hat{W}(T)\right]}{Stdev_p^{(t_0,w_0)}\left[W(T) - \hat{W}(T)\right]}, \tag{5.1}$$

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 difference 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:

$$(CD): \quad F(\mathcal{W}_p, \hat{\mathcal{W}}_{\hat{p}}) = \sum_{t \in \mathcal{T} \cup \{T\}} \Big(W(t) - e^{\beta t}\hat{W}(t)\Big)^2 \Delta t. \tag{5.2}$$

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 rate* $e^{\beta t}\hat{W}(t)$, a concept proposed in Ni et al. (2022). Unlike (5.1), the CD objective captures the intermediate tracking performance of the portfolio, and the parameter β provides a clear performance goal.

However, CD metric (5.2) minimizes the relative performance between the active portfolio and the elevated target quadratically. In other words, both underperformance and outperformance are penalized. In practice, 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 computational investigation.

$$(CS): \quad F(\mathcal{W}_p, \hat{\mathcal{W}}_{\hat{p}}) = \sum_{t \in \mathcal{T} \cup \{T\}} \Big(\min\big(W(t) - e^{\beta t}\hat{W}(t), 0 \big) \Big)^2 \Delta t. \tag{5.3}$$

Consequently, we investigate the following optimization problem in computational assessment:

$$\inf_{\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}} \left\{ \frac{1}{N_d} \sum_{j=1}^{N_d} \sum_{t \in \mathcal{T} \cup \{T\}} \Big(\min\big(W_{\boldsymbol{\theta}}^{(j)}(t) - e^{\beta t}\hat{W}^{(j)}(t), 0 \big) \Big)^2 \Delta t \right\}. \tag{5.4}$$

It is worth noting that in Equation (5.3), the β parameter also reflects the risk appetite of the investor. In order to achieve a higher β value, the investor needs to take more risk to achieve a higher expected return, thus investing in riskier assets. In §5, we will explore varying value of β computationally and examine how this affects the optimal strategy.

Fundamentally, the goal of the CS objective is to find a balance between portfolio return and risk. It is clear that investors aim to exceed the benchmark portfolio return, as demonstrated by the outperformance parameter β, which indicates an expected annualized premium over the benchmark return. Simultaneously, the CS objective also aims to bound the portfolio's tail risks by penalizing underperformance relative to the elevated target quadratically. For further discussion of the CS objective function, we refer the reader to van Staden et al. (2024); Ni et al. (2024).

For the computational investigation, we approximate the expectation in (3.20) by utilizing samples from a finite training sample set $\boldsymbol{Y} = Y^{(j)} : j = 1, \cdots, N_d$, where N_d denotes the total number of samples. Here, $Y^{(j)}$ represents the j^{th} sample return path comprising joint observations of asset returns $\big\{R_i(t), i \in \{1, \cdots, N_a\}\big\}$, observed at $t \in \mathcal{T}$.[2] Mathematically, the approximation of problem (3.20) can be formulated as follows:

$$\inf_{\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}} \left\{ \frac{1}{N_d} \sum_{j=1}^{N_d} F\left(\mathcal{W}_{\boldsymbol{\theta}}^{(j)}, \hat{\mathcal{W}}_{\hat{p}}^{(j)} \right) \right\}. \tag{5.5}$$

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 control model parameterized by $\boldsymbol{\theta}$. Similarly, $\hat{\mathcal{W}}_{\hat{p}}^{(j)}$ denotes

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

the wealth trajectory of the benchmark portfolio, following the benchmark strategy $\hat{p}$, i.e., $\hat{\mathcal{W}}_{\hat{p}}^{(j)} = \big(\hat{W}^{(j)}(t_0), \cdots, \hat{W}^{(j)}(t_N)\big)$. Both portfolios are evaluated based on the j-th sample return path, $Y^{(j)}$.

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 consists of a 3-tuple vector $(t, W_{\boldsymbol{\theta}}(t), \hat{W}(t))^\top$. Here, at any time $t \in [t_0, T]$, $W_{\boldsymbol{\theta}}(t)$ represents the wealth of the active portfolio determined 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 that the control model parameters can be computed directly using gradient descent-based methods. Essentially, the control model function is a recurrent neural network (RNN), and the procedure for calculating the gradient of the objective function along the j^{th} path is outlined as follows.

$$\nabla_{\boldsymbol{\theta}} F\left(\mathcal{W}_{\boldsymbol{\theta}}^{(j)}, \hat{\mathcal{W}}_{\hat{p}}^{(j)}\right) = \sum_{i=1}^{N} \frac{\partial F}{\partial W_{\boldsymbol{\theta}}^{(j)}(t_i)} \nabla_{\boldsymbol{\theta}} W_{\boldsymbol{\theta}}^{(j)}(t_i). \tag{5.6}$$

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

$$W_{\boldsymbol{\theta}}^{(j)}(t_i) = f_{\boldsymbol{\theta}}\big(W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}), \hat{W}^{(j)}(t_{i-1}), t_{i-1}\big)^\top \big(1 + \boldsymbol{R}(t_i)\big) W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}) \mathbf{1}_{W_{\boldsymbol{\theta}}^{(j)}(t_{i-1})>0}, \tag{5.7}$$

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 (5.6) can be obtained recursively using the chain rule, i.e.,

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} W_{\boldsymbol{\theta}}^{(j)}(t_i) &= \nabla_{\boldsymbol{\theta}} \Big(f_{\boldsymbol{\theta}}\big(W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}), \hat{W}^{(j)}(t_{i-1}), t_{i-1}\big)^\top \big(1 + \boldsymbol{R}(t_i)\big) W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}) \mathbf{1}_{W_{\boldsymbol{\theta}}^{(j)}(t_{i-1})>0} \Big) && (5.8) \\
&= \nabla_{\boldsymbol{\theta}} \Big(f_{\boldsymbol{\theta}}\big(W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}), \hat{W}^{(j)}(t_{i-1}), t_{i-1}\big)^\top \big(1 + \boldsymbol{R}(t_i)\big) \Big) W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}) \mathbf{1}_{W_{\boldsymbol{\theta}}^{(j)}(t_{i-1})>0} \\
&\quad + \Big(f_{\boldsymbol{\theta}}\big(W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}), \hat{W}^{(j)}(t_{i-1}), t_{i-1}\big)^\top \big(1 + \boldsymbol{R}(t_i)\big) \mathbf{1}_{W_{\boldsymbol{\theta}}^{(j)}(t_{i-1})>0} \Big) \nabla_{\boldsymbol{\theta}} W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}) && (5.9) \\
&= \Big(\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}\big(W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}), \hat{W}^{(j)}(t_{i-1}), t_{i-1}\big) \Big) \big(1 + \boldsymbol{R}(t_i)\big) W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}) \mathbf{1}_{W_{\boldsymbol{\theta}}^{(j)}(t_{i-1})>0} \\
&\quad + \Big(\frac{\partial f_{\boldsymbol{\theta}}\big(W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}), \hat{W}^{(j)}(t_{i-1}), t_{i-1}}{\partial W_{\boldsymbol{\theta}}^{(j)}(t_{i-1})} \Big)^\top \big(1 + \boldsymbol{R}(t_i)\big) \mathbf{1}_{W_{\boldsymbol{\theta}}^{(j)}(t_{i-1})>0} \nabla_{\boldsymbol{\theta}} W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}) \\
&\quad + \Big(f_{\boldsymbol{\theta}}\big(W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}), \hat{W}^{(j)}(t_{i-1}), t_{i-1}\big)^\top \big(1 + \boldsymbol{R}(t_i)\big) \mathbf{1}_{W_{\boldsymbol{\theta}}^{(j)}(t_{i-1})>0} \Big) \nabla_{\boldsymbol{\theta}} W_{\boldsymbol{\theta}}^{(j)}(t_{i-1}) && (5.10)
\end{aligned}$$

Subsequently, the optimal parameter $\boldsymbol{\theta}^*$ can be determined numerically by solving problem (5.5) 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 test dataset $\boldsymbol{Y}_{test}$, which consists of samples unseen in training dataset $\boldsymbol{Y}$.

5.2 Bootstrap resampled data

To evaluate performance of the optimal leveraged strategy RCNN over the long-only strategy in outperforming a benchmark, we use the U.S. monthly data from the Center for Research in Security Prices (CRSP)[3] from January 1926 to January 2023. Particularly, we obtain the real historical returns of the equal-weighted/cap-weighted U.S. stock indexes and 10-year/30-day treasury indexes by adjusting for the CPI index.

Conventional approaches in mathematical finance often involve fitting a parametric synthetic model, e.g., a stochastic process model, to the original historical asset price data and subsequently resampling from the fitted model (Merton, 1976; Kou, 2002). While such a synthetic model may offer the advantage of often providing a closed-form solution, it also presents 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 a chosen parametric synthetic model is likely to be inconsistent with the characteristics of the real-world financial markets; as such, the validity of synthetic models is often up to debate.

Understanding these limitations, alternative to parametric models, we employ a 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 is considered a relatively unbiased approach.

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 A. 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, 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).

[3] ©2023 Center for Research in Security Prices (CRSP), The University of Chicago Booth School of Business. Wharton Research Data Services (WRDS) was used in preparing this article. This service and the data available thereon constitute valuable intellectual property and trade secrets of WRDS and/or its third-party suppliers.

5.3 Investment specifications

We consider four assets for the experiment: 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 mentioned in §5.2, we use monthly CRSP data from January 1926 to January 2023. Since active portfolios are often evaluated by their relative performance against a passive benchmark, we choose a simple 70/30 portfolio as the benchmark, which always maintains 70% wealth in the equal-weighted stock index,[4] and 30% in the 30-day T-bill index.

Table 5.1 outlines the investment scenario is outlined. In summary, both the active portfolio and the benchmark portfolio commence with an initial wealth of 100 at time $t_0 = 0$. Monthly rebalancing is implemented for both portfolios over a 10-year investment horizon.

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
Investment objective	Cumulative quadratic shortfall (CS)
Outperformance target rate β	0.5% - 5%, incremental by 0.5%

Table 5.1: Investment scenario.

The optimal long-only portfolio under the cumulative quadratic shortfall objective is computed using the neural network model for long-only constraints, as proposed in (Li and Forsyth, 2019; Ni et al., 2022). Briefly, a two-layer feed-forward neural network, with a softmax activation function at the output layer, is used to approximate the optimal control function. This neural network uses the same state vector in this article as input (i.e. wealth of portfolios and time), and outputs an allocation vector which satisfies the long-only constraint, which is a consequence of using the softmax activation function.

The optimal relaxed-constraint portfolio is computed using the RCNN as described in §2. We note, however, that the proposed neural network methodology is agnostic to the choice of the objective function and can be applied to a broad range of performance metrics.

5.4 Enhanced performance of RCNN over long-only

By varying the outperformance target parameter β across the range of 0.5% to 5% (incremental with a 0.5% step size), we obtain the corresponding optimal portfolios through the

[4] In Ni et al. (2024), bootstrap simulations based on long term historical data show that equal weight indexes partially stochastically dominate capitalization weighted indexes.

cumulative quadratic shortfall (CS) objective.

In Table 5.2, we present the performance of the computed optimal 130/30 portfolio and optimal long-only portfolio for tracking elevated targets. Particularly, the outperforming performance is reflected in the value of the CS objective, which measures the cumulative quadratic shortfall with respect to elevated targets defined by the target rate β.

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

Table 5.2: CS objective function values for the optimally trained control models on the test data for various β (lower is better). The results are based on the performance of trained models evaluated on the test dataset $\boldsymbol{Y}_{test}$.

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

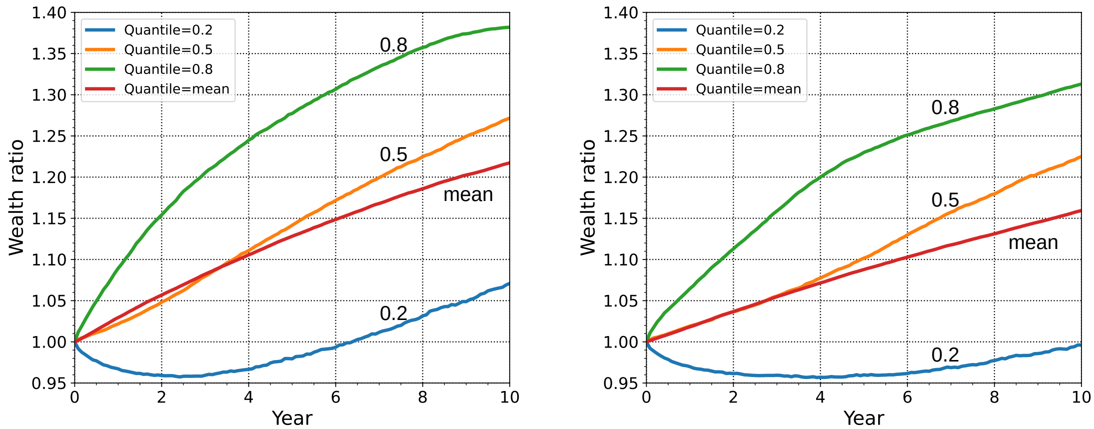
Particularly, the gap between the CS objective function values widens as the target outperformance rate β increases, indicating that the long-only portfolio is further restricted by the long-only constraints as the outperformance target becomes more ambitious. This phenomenon is further demonstrated in Table 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 of the optimal trained control models on test data. The benchmark portfolio has a median annualized return of 6.7%. The results are based on the performance of trained models evaluated on the test dataset $\boldsymbol{Y}_{test}$.

As we can see from Table 5.3, when β is modest ($< 3\%$), the long-only portfolio shows similar median returns as the 130/30 portfolio (despite that the objective function value is slightly worse). However, as β becomes more ambitious ($\geq 3\%$), the long-only portfolio has a harder time keeping up with the 130/30 portfolio. Specifically, we can see that the median return of the long-only portfolio stagnates for $\beta \geq 3\%$. As we will discuss shortly, at $\beta = 3\%$, the optimal long-only portfolio is already allocating almost 100% allocation to the equal-weighted stock index, the riskiest asset with the highest expected return. Due to long-only constraints, there is less room for the long-only portfolio to take more risks for the more aggressive β targets. On the other hand, we can see that the median return of the optimal 130/30 portfolio continues to increase with β.

Next we present more detailed comparison of the optimal RCNN and long-only strategies on additional performance characteristics for $\beta = 3\%$. We plot the quantiles of the wealth ratio $\frac{W(t)}{\hat{W}(t)}$, which measures the relative pathwise performance of the active portfolio with respect to the benchmark portfolio throughout the investment horizon.



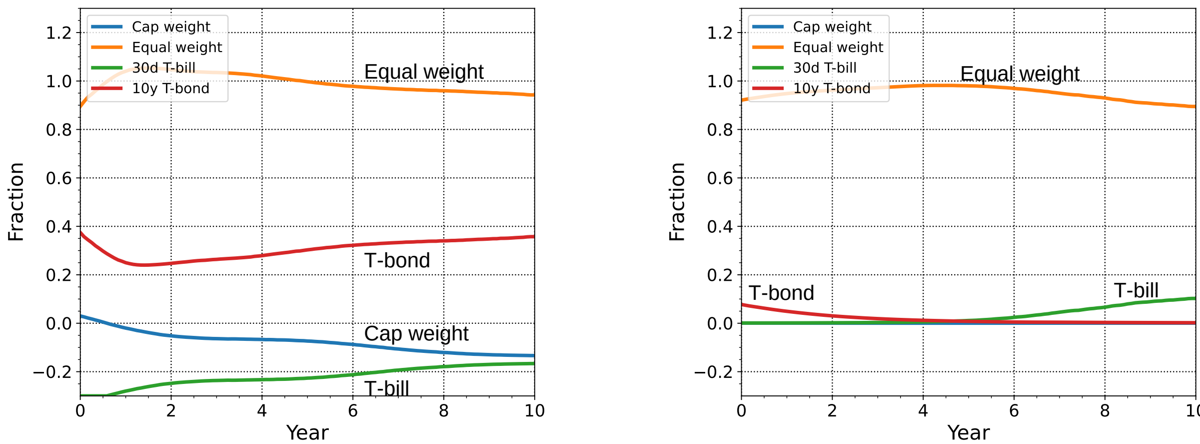
(a) 130/30 portfolio (b) Long-only portfolio

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

Based on the results and analysis presented in Figure 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. We emphasize that Figure 5.1 compares pathwise performance of the active (dynamic) portfolio compared to the benchmark. If wealth ratio is viewed as a risk measure, the leveraged portfolio is actually less risky than the unleveraged portfolio.

Unsurprisingly, the superior performance of the 130/30 portfolio can be attributed to its relaxed portfolio constraints. We plot the median allocation fractions of the 130/30 portfolio and long-only portfolio in Figure 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 optimal long-only portfolio is obviously restricted by the long-only constraint. It yields an almost trivial strategy that has a close to 100% allocation to the equal-weighted stock index throughout the investment horizon.

We remark however that there is no free lunch, and the optimal 130/30 strategy achieves superior results with some compromises. Particularly, if we examine the extreme tail statistics such as the 1% CVaR of the terminal wealth (i.e. the average of the lowest 1% of the



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

Figure 5.2: Median allocation fractions over the investment horizon $[0, T]$ when $\beta = 3\%$. 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 $\boldsymbol{Y}_{test}$.

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 under rare and persistent bear market scenarios.

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 (mean of the worst one percent of the outcomes, higher is better). The results are based on the performance of trained models evaluated on $\boldsymbol{Y}_{test}$.

However, one cannot simply conclude that (optimal) 130/30 portfolios are riskier than (optimal) long-only portfolios. If we look at 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, i.e., better pathwise outperformance compared to the benchmark. This suggests that the 130/30 portfolio is capable of mitigating downside risks well in majority of scenarios.

We remind the reader that we cannot obtain a strategy which is guaranteed to outperform a benchmark along every path, since this would imply the existence of an arbitrage opportunity.

Overall, our computational investigation demonstrates the superiority of the relaxed-constraint portfolio under the tracking performance-based investment objective. As shown in the results, the 130/30 portfolio not only achieves more ambitious returns but also demonstrates good risk management. This can be attributed to the broader range of portfolio strategies available within the 130/30 structure, which allows for more flexibility and potential for generating excess return.

In addition, we present computational evidence that illustrates the effectiveness of the proposed RCNN approach, for which it is not necessary to determine *a priori* which assets need to be shorted. The optimal control solution will find the most effective strategy.

6 Conclusion

In this article, we introduced a neural network-based solution for the multi-period optimization problem under relaxed-constraint, which permits bounded leverage. By formulating the problem as a multi-period stochastic optimal control problem, we proposed a novel relaxed-constraint neural network (RCNN) model to approximate the optimal control.

The RCNN addresses the complexity of the original leverage constrained optimization by proposing a novel activation function and converting the leverage constraint formulation into an unconstrained optimization problem, which can be computationally solved efficiently. In addition, we provided mathematical proof demonstrating that the RCNN can accurately approximate any relaxed-constraint strategy.

Based on monthly U.S. market return data from Jan 1926 to Jan 2023, we computationally assess performance of the optimal relaxed-constraint strategy with long-only strategy. As an illustration, we consider the 130/30 portfolio. We compared the performance of the optimal relaxed-constraint portfolio with the optimal long-only portfolio under the same investment specifications. The optimal portfolios are computed and evaluated under the cumulative quadratic shortfall (CS) objective, which measures the relative performance of the active portfolio against a benchmark portfolio throughout the investment horizon. The computational assessment consistently demonstrates that the optimal relaxed-constraint portfolio outperforms the optimal long-only portfolio under the CS 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 is worth considering other types of securities such as options in the portfolio (Andersson and Oosterlee, 2023), which may yield even better results in practice.

7 Acknowledgement

Forsyth's work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) grant RGPIN-2017-03760. Li's work was supported by he Natural Sciences and Engineering Research Council of Canada (NSERC) grant RGPIN-2020-04331.

A Stationary block bootstrap algorithm

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

Algorithm A.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*.

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. *U. Chi. L. Rev. 52*, 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.

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.

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? Technical report, Lund University.

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. and J. M. Mulvey (2021). Portfolio optimization under regime switching and transaction costs: Combining neural networks and dynamic programs. *INFORMS Journal on Optimization 3*(4), 398–417.

Li, X., A. S. Uysal, and J. M. Mulvey (2022). Multi-period portfolio optimization using model predictive control with mean-variance and risk parity frameworks. *European Journal of Operational Research 299*(3), 1158–1176.

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, and P. Forsyth (2024). Neural network approach to portfolio optimization with leverage constraints: a case study on high inflation investment. *Quantitative Finance*, 1–25.

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.

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

Pun, C. S. and H. Y. Wong (2019). A linear programming model for selection of sparse high-dimensional multiperiod portfolios. *European Journal of Operational Research 273*(2), 754–771.

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.

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.

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