Quantitative Finance

ISSN: 1469-7688 (Print) 1469-7696 (Online) Journal homepage: https://www.tandfonline.com/loi/rquf20

Optimizing a portfolio of mean-reverting assets with transaction costs via a feedforward neural network

John M. Mulvey, Yifan Sun, Mengdi Wang & Jing Ye

To cite this article: John M. Mulvey, Yifan Sun, Mengdi Wang & Jing Ye (2020): Optimizing a portfolio of mean-reverting assets with transaction costs via a feedforward neural network, Quantitative Finance, DOI: 10.1080/14697688.2020.1729994

To link to this article: https://doi.org/10.1080/14697688.2020.1729994

Published online: 07 Apr 2020.

Quantitative Finance, 2020
https://doi.org/10.1080/14697688.2020.1729994

Optimizing a portfolio of mean-reverting assets with transaction costs via a feedforward neural network

JOHN M. MULVEY*†, YIFAN SUN‡, MENGDI WANG† and JING YE†

†Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ, USA
‡Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA, USA

(*Received 1 August 2019; accepted 28 January 2020; published online 7 April 2020*)

Optimizing a portfolio of mean-reverting assets under transaction costs and a finite horizon is severely constrained by the curse of high dimensionality. To overcome the exponential barrier, we develop an efficient, scalable algorithm by employing a feedforward neural network. A novel concept is to apply HJB equations as an advanced start for the neural network. Empirical tests with several practical examples, including a portfolio of 48 correlated pair trades over 50 time steps, show the advantages of the approach in a high-dimensional setting. We conjecture that other financial optimization problems are amenable to similar approaches.

Keywords: Asset allocation; Portfolio allocation; Portfolio optimization; Statistical learning theory; Stochastic programming

JEL Classification: C61, G11, C14, C32

1. Introduction

One of the most difficult and significant problems in computational finance is the multi-stage, stochastic financial planning model. The multi-stage framework affords a variety of practical considerations that are difficult to address consistently in a single-period model: transaction costs, turnover, temporal risks including drawdown, goals at differing time periods, tradeoffs of short term savings versus long-term benefits, inflows and outflows of capital, and Markovian transitions. These issues have been extensively studied by researchers over the past 60+ years. References Bertocchi *et al.* (2011), Dempster *et al.* (2007), Hakansson (1969), Mulvey *et al.* (2003), Shreve and Soner (1994), Ziemba and Mulvey (1998) and Zenios and Ziemba (2007) present illustrative examples.

Despite their general applicability, multi-stage stochastic planning models are severely constrained by the curse of high dimensionality. Accordingly, the model's scope is limited to allow for a practical solution. There are several avenues to pursue: (1) *Stochastic Control*: we render a set of severe assumptions such as Samuelson (1969) and Merton (1969, 1971, 1973) in their early and innovative work to convert a stylized multi-stage model into a sequence of myopic models with a fixed asset mix. The general idea can be interpreted as solving the HJB equations directly. (2) *Dynamic Programming*: given a reasonable-size state space and Markovian assumptions (and additive utility), we can apply DP and approximate DP algorithms. This idea has been much expanded in the AI/machine learning domain under the term reinforcement learning; (3) *Stochastic Programs*: set up a generic model with a finite set of scenarios (realizations of uncertainties) and decision branches and solve the resulting model as a large-scale nonlinear program; (4) *Policy-Rule Simulation*: assume one or more non-anticipative policy rules, apply a Monte Carlo simulation/optimization to evaluate and compare policy rules.

Inevitably, each of these avenues reduces the usefulness of the methods for assisting investors and portfolio managers faced with realistic multi-stage financial planning problem. There can be advantages for combining approaches (Konicz and Mulvey 2013).†

A similar computational barrier arises when addressing high-dimensional systems of partial differential equations. Recently, Weinan *et al.* (2017) have proposed applying

*Corresponding author. Email: mulvey@princeton.edu

† The Google deep learning group has implemented dual methods in their world-class gaming systems for playing chess, shogi, and Go (Silver *et al.* 2016).

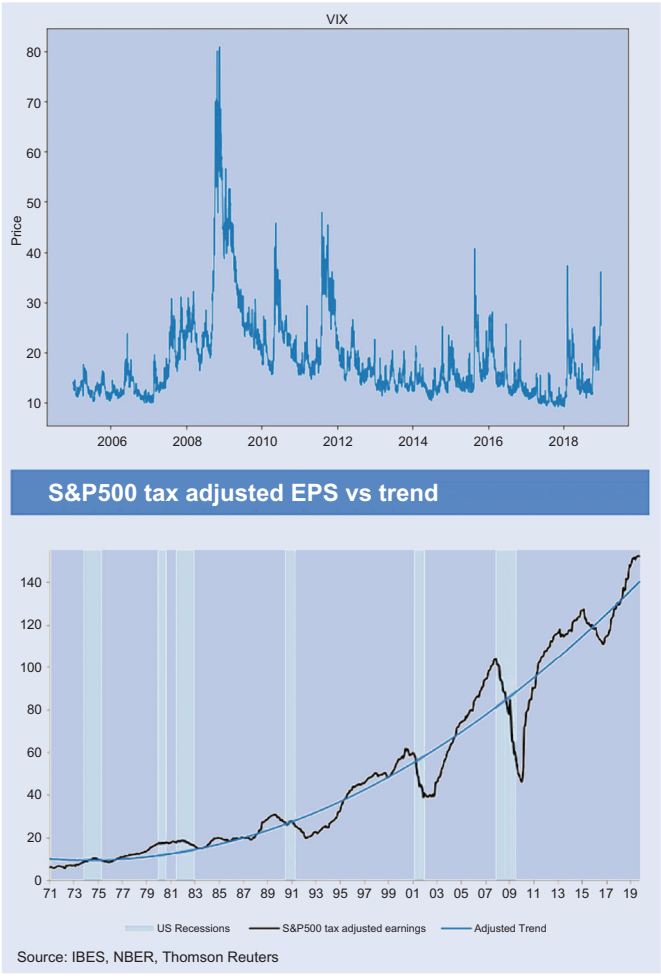

Figure 1. Assets with mean-reverting behavior.

feedforward neural networks to overcome the curse of dimensionality in the PDE domain (Han and Weinan 2016, Weinan *et al.* 2017, Han *et al.* 2018). In this paper, we take a similar approach.

To improve efficiency, we device an advanced starting algorithm based on solving a system of differential HJB equations. The advanced start improves the efficacy of the integrated solution process. In particular, we combine approaches by linking the HJB system for an approximate problem (without transaction costs) with the feedforward neural network to address the high-dimensional elements. The network approximates the stochastic derivative of the objective as a function of the transactional boundaries.

For this study, we focus on optimizing a portfolio of correlated assets with mean-reverting behavior, as characterized by Ornstein–Ulenbeck (OU) processes. Many investments display mean reversion, including volatility and earnings growth over a target rate (Figure 1). Our goal is to maximize the expected utility of wealth at the end of a finite horizon T, wherein the utility function assumes constant relative risk aversion (CRRA). Importantly, we add linear transaction costs for the purchase or sale of an asset, and assume a finite set of time steps and zero cashflows over the planning period.

The traditional approach for addressing transaction costs is to apply non-trade zones around a specified asset allocation. Thus given an allocation for any particular asset at time t, say x_t^*, we would designate boundaries x-lower and x-upper around x_t^*, and the investor would trade whenever the asset value falls outside the limits. References Davis and Norman (1990), Mulvey and Simsek (2002), Shreve and Soner (1994) and Taksar *et al.* (1988) provide further details. We build on this concept in this study, wherein the feedforward neural network sets the no-trade zone boundaries at each time point and state of the system.

The remainder of this paper is organized as follows. The next section surveys research on multi-period portfolio models, with and without transaction costs. In Section 3, we derive the theoretical results for the finite horizon optimal allocation problem under the no-transaction costs assumption. Section 4 describes our deep neural network (DNN) method to parametrize the no-trade zone and derive a trading strategy based on a no-trade zone. Section 5 provides a training procedure and numerical results from synthetic data. Results in this section show the no-trade zone characterized by the DNN meets our intuition. Section 6 shows backtest results of our hybrid algorithm applied to real data of stock-ETF pairs trading with transaction costs.

2. Literature review

The dynamic portfolio optimization problem is an essential topic in mathematical finance that has been studied since the 1970s. Merton (1969, 1971) established the framework for dynamic portfolio choice with stochastic variation in investment opportunities by explicitly solving a continuous-time portfolio problem where the investor can invest between stocks modeled as geometric Brownian motions and a money market account with a fixed risk-free rate to maximize the expected utility of consumption and terminal wealth. Herein the optimal strategy is a fix-mix strategy for the constant relative risk aversion (CRRA) utility when there are no transaction costs. Mossin (1968), Samuelson (1969) and Hakansson (1969) solve analytically a similar problem in discrete settings.

There has been a significant literature extending Merton's problem from differing perspectives. One main stream is to incorporate stock return predictability into portfolio optimization, for example, by modeling the stock returns as mean reverting (Ornstein–Uhlenbeck) processes. Campbell and Viceira (1999) assume an infinite investment horizon and derive an approximate analytical solution. Kim and Omberg (1996) derive an exact solution under hyperbolic absolute risk aversion (HARA) utility by limiting the investor to maximizes utility over terminal wealth only. Wachter (2002) solves, in closed form, the optimal portfolio choice problem for an investor with utility over both consumption and terminal wealth under a complete market assumption. Liu (2006) extends the work to multiple risky assets and explicitly solves up to the solution of an ODE system when the investor has a constant relative risk aversion (CRRA) utility and the asset returns follow a quadratic process which includes both the affine process and the Ornstein–Uhlenbeck process as special cases. Munk *et al.* (2004) consider not only stocks with mean-reverting

excess return, but also bond trading and inflation. Ma and Zhu (2019) assume a continuous-time cointergration model where the risky assets follow Ornstein–Uhlenbeck processes, and derive the optimal investment and consumption for investors with exponential utility.

In the above-mentioned studies, the authors either solve or approximate the optimal trading strategy by assuming no transaction costs. However, dynamic portfolio optimization often requires frequent rebalancing and hence transaction costs are usually not negligible. Such trading costs are caused by several factors such as the bid-offer spread, execution commissions, market impact or tax, etc. Magill and Constantinides (1976) initiated the research in transaction costs by proposing that the investors only trades in securities when the variation in the underlying security prices forces his portfolio proportions outside a no-trade zone. Davis and Norman (1990) were the first to provide a detailed formulation and analysis, along with an algorithm and numerical computation of the optimal policy for an infinite-horizon investment and consumption decision problem. Shreve and Soner (1994) relax some assumptions in Davis and Norman's work and conduct an analysis of the optimal trading strategies over an infinite horizon. They prove existence, uniqueness and regularity of the value function. Liu and Loewenstein (2002) focus on the finite-horizon optimal trading problem with a single risky asset. The multi-asset portfolio optimization problem is more difficult to solve. Liu (2004) obtains an almost closed-form solution for fixed and proportional costs in continuous time for infinite lived constant absolute risk aversion (CARA) investors when asset returns are uncorrelated. Muthuraman and Kumar (2006) develop numerical methods to solve the free boundary Hamilton–Jacobi–Bellman (HJB) equation for the case of two correlated risky assets. Muthuraman and Zha (2008) construct numerical methods and provide a computational scheme with runtime scaling polynomially in the number of assets. Lynch and Tan (2010) numerically solve over a finite discrete horizon a similar problem with two risky assets to Muthuraman and Kumar (2006) but incorporate return predictability for the first time. Their methods are based on a grid approximation of the state space for the associated dynamic program.

Yet, there is no guarantee that the aforementioned numerical methods are close to optimal or any indication of how much better one might do with an optimal strategy. Brown *et al.* (2010), Brown and Smith (2011) develop a dual bounding technique that helps evaluate the quality of the trading strategies. Broadie and Shen (2017) provide three lower bounds for the optimal solution of the transaction cost involved in a portfolio choice problem with 20 risky assets and 40 investment periods: the value function optimization (VF), the hypersphere and the hyper-cube policy parameterizations (HS and HC). They also achieve tighter upper bounds by improving the duality method in Brown *et al.* (2010), Brown and Smith (2011).

In addition to proportional transaction cost, we mention research on non-linear transaction cost. Grinold (2006) derives the optimal steady-state position with quadratic trading costs and a single predictor of returns per security. Gârleanu and Pedersen (2013) derive a closed-form solution for a model with linear dynamics for return predictors, quadratic functions for transaction costs, and quadratic penalty terms for risk. Chan and Sircar (2015) consider a class of dynamic portfolio optimization problems incorporating return predictability, and stochastic volatility with quadratic transaction cost.

Rebalancing a multi-asset portfolio back to a target no-trade zone is a practically important problem. Mulvey and Simsek (2002), Mulvey *et al.* (2003) set up a multi-stage optimization model for investing in assets over an extended time horizon. In particular, they show that the rebalancing problem can be posed as a generalized network with side constraints and propose a search algorithm for finding a feasible solution with the lowest transaction cost.

Despite prior work on transaction-cost models, there are few scalable solutions the optimal trading strategy with multiple risky assets due to the curse of dimensionality, i.e. the computational cost scales exponentially with the number of assets. Ritter (2017) propose a reinforcement learning framework for solving the trading strategy under the assumptions of mean-reverting assets and quadratic transaction costs. The author discretizes the state space as well as the action space and defines the reward function accordingly. The algorithm works well even when the agent does not, at least initially, have any information about the movement of the underlying risky asset price and about the form of transaction costs. Yet, this reinforcement learning method suffers from the curse of dimensionality and may not be suitable for multiple risky assets. On the other hand, during recent years, deep learning has made breakthroughs in many areas such as image recognition and game playing, as well as finance (LeCun *et al.* 2015). Culkin and Das (2017) survey how and why deep learning can influence the field of finance in a very general way with a specific application to reproducing the Black and Scholes option pricing formula to a high degree of accuracy by training a fully-connected feed-forward deep learning neural network. Weinan *et al.* (2017) propose a new method for solving high-dimensional fully nonlinear second-order partial differential equations (PDEs) herein. The PDEs are reformulated as a control theory problem with the gradient of the unknown solution approximated by neural networks, like deep reinforcement learning with the gradient acting as the policy function. Their technique has inspired us to use a feedforward neural network to solve the free boundary HJB equation.

Last, we briefly discuss applications in the financial markets, particularly in pairs trading, which was first initiated in Morgan Stanley in 1980s. The original idea is straightforward to find a pair of stocks whose spread exhibits potentially mean-reverting behavior. Research on pairs trading or statistical arbitrage has flourished over the next three decades, see Pole (2007) and Leung and Li (2016) for a comprehensive review on statistical arbitrage. Benth and Karlsen (2005) analyze the classical Merton's portfolio optimization problem when portfolio contains one risky asset following an exponential Ornstein–Uhlenbeck (XOU) process. Jurek and Yang (2007) consider a similar problem to the one we did in this paper, a finite portfolio optimization problem consisting of a single OU asset subject to CRRA utility and Epstein-Zin recursive utility, but they only provide explicit solutions to an optimal trading strategy with a single mean-reverting asset and uncorrelated multi-assets. Tourin and Yan (2013), Liu

and Timmermann (2013) solve the portfolio choice problem comprising of a risk-free asset and two co-integrated and correlated stocks. Yet none of the aforementioned researches assumed transaction cost in the model. For constructing the mean-reverting tradable, we slightly modify the stock-ETF pairs trading methods in Avellaneda and Lee (2010). They use principal component analysis (PCA) and regress stock returns on sector exchange traded funds (ETFs) to construct trading signal and model the idiosyncratic returns as mean-reverting processes. The exact construction process and related carry cost will be discussed in the corresponding section. In addition, there are other applications of the methodology to asset allocation as well as asset and liability management.

First, we model the asset prices, rather than asset returns or risk premiums as in much other literature, as Ornstein–Uhlenbeck process and extend the model to correlated multi-assets portfolio when there is no transaction cost. This is meaningful when portfolios contains mean-reverting assets such as CBOE volatility index (VIX) future or assets created by pairs trading. Second, we develop a novel numerical method using a feedforward neural network to parametrize the trading boundaries of the no-trade zone in a dynamic fashion. Our study shows when there are transaction costs, the integrated method significantly improves the policy we derived under the no-transaction-cost assumption. Moreover, our method is scalable to the high-dimensional case. To our best knowledge, previous work has not been done on studying the transaction boundary in the high-dimensional case with asset prices modeled as correlated Brownian motions or mean-reverting processes.

3. Optimal strategy under zero transaction costs

In this section, we analyze the optimal trading strategy for the basic setting. We start with the case where there are one risky asset and one risk-free asset with zero transaction costs and then extend to the case where there are multiple risky assets. The solution is characterized by an ODE system. It will provide a starting solution for training deep neural networks.

3.1. Optimal strategy with one risky asset

Following Jurek and Yang (2007), suppose there is one risky asset whose price X_t follows an Ornstein–Uhlenbeck (OU) process and a money market account value by Y_t:

$$\begin{aligned} \mathrm{d}X_t &= \lambda(\mu - X_t)\,\mathrm{d}t + \sigma\,\mathrm{d}Z_t \\ \frac{\mathrm{d}Y_t}{Y_t} &= r\,\mathrm{d}t, \end{aligned} \tag{1}$$

where $\lambda \in \mathbb{R}^+$ is the mean-reversion speed parameter, $\sigma \in \mathbb{R}^+$ is the volatility parameter and $\mu \in \mathbb{R}$ is the mean parameter of the OU process and Z_t denotes the standard Brownian motion, $r \in \mathbb{R}$ is the constant risk free rate. Let $\pi(t) \in \mathbb{R}$ be the current proportion of wealth invested in the risky asset at time t. The total wealth W_t follows:

$$\frac{\mathrm{d}W_t}{W_t} = \pi_t \frac{\mathrm{d}X_t}{X_t} + (1-\pi_t)\frac{\mathrm{d}Y_t}{Y_t}. \tag{2}$$

Our goal is to maximize the expected utility of the terminal wealth W_T at a given finite horizon T:

$$\max_{\pi} \mathbb{E}[U_\gamma(W_T)], \tag{3}$$

where we choose $U_\gamma(W)$ to be the CRRA utility:

$$U(W) = \begin{cases} \dfrac{W^{1-\gamma}-1}{1-\gamma} & \text{if } \gamma > 0 \text{ and } \gamma \neq 1, \\ \ln(W) & \text{if } \gamma = 1. \end{cases} \tag{4}$$

In this section, we focus on solving the above problem analytically with a fixed value $\gamma > 1$. We use the notation $U(W)$ instead of $U_\gamma(W)$ for simplicity.

Let $\tau = T - t$ be the horizon of the investment period. Assume $J(W, X, \tau)$ to be the indirect utility function which satisfies the boundary condition $J(W, X, 0) = U(W)$. We derive the Hamilton–Jacobi–Bellman equation:

$$\begin{aligned} \max_{\pi} \Bigg\{ & -J_\tau + \left[\frac{\pi\lambda}{X}(\mu - X) + (1-\pi)r\right] W J_W \\ & + \frac{1}{2}\left(\frac{\pi\sigma}{X}\right)^2 W^2 J_{WW} \\ & + \lambda(\mu - X)J_X + \frac{1}{2}\sigma^2 J_{XX} + \sigma\frac{\pi\sigma}{X} W J_{WX} \Bigg\} = 0, \end{aligned} \tag{5}$$

where J_τ,J_W, and J_X denote the partial derivatives of J with respect to t, W, and X, respectively. Similarly, J_{WW}, J_{XX} and J_{WX} denote the higher order partial derivatives. From equation (5), we can compute the optimal asset allocation:

$$\pi^*(\tau, X, W) = -\frac{J_W}{W J_{WW}}\left[\frac{\lambda(\mu - X)/X - r}{(\sigma/X)^2}\right] - \frac{X J_{WX}}{W J_{WW}}. \tag{6}$$

The first term in equation (6) is the scaled mean-variance efficient portfolio weight, also called the myopic demand because this is the vector of portfolio weights for an investor who only optimizes over one single period. The coefficient $-J_W/WJ_{WW}$ is the counterpart of inverse relative risk aversion $1/\gamma = -U'(W)/WU''(W)$ in the indirect utility. The higher the relative risk aversion is, the less risky asset one should hold as a proportion of the portfolio. The second term in equation (6) represents the inter-temporal hedging demand, which arises in a multi-period portfolio choice problem when an investor accounts for changes in the investment opportunity set and tries to hedge against adverse future shocks.

The HJB equation (5) is solved by first 'guessing' a general form for the solution which is then verified later. Following the past literature on portfolio optimization of assets with mean-reverting properties (see Kim and Omberg 1996, Wachter 2002, Benth and Karlsen 2005, Jurek and Yang 2007), we assume the indirect utility takes the form:

$$J(W, X, \tau) = \frac{(W\phi(\tau))^{1-\gamma} - 1}{1-\gamma} \tag{7}$$

$$\phi(\tau) = \exp(A(\tau) + B(\tau)X + C(\tau)X^2/2) \tag{8}$$

$$A(0) = B(0) = C(0) = 0 \tag{9}$$

The idea of the ansatz (7)–(9) comes from the bond pricing: bond prices under an affine term structure can be expressed

as exponentials of a polynomial of the underlying state variables. Substituting π^* given by (6) and the ansatz (7)–(9) into equation (5) generates a quadratic equation for X_t. Making all the coefficients zeros, we obtain the following ODE system of $A(\tau)$, $B(\tau)$ and $C(\tau)$:

$$\begin{aligned} C'(\tau) &= aC^2(\tau) + bC(\tau) + c \\ B'(\tau) &= aB(\tau)C(\tau) + \frac{b}{2}B(\tau) + dC(\tau) + g \\ A'(\tau) &= \frac{a}{2}B(\tau)^2 + dB(\tau) + \frac{\sigma^2}{2}C(\tau) + \frac{(\lambda\mu)^2}{2\gamma\sigma^2} + r, \end{aligned} \tag{10}$$

with boundary condition $A(0) = B(0) = C(0) = 0$ and parameters:

$$\begin{aligned} a &= \frac{1-\gamma}{\gamma}\sigma^2, \quad b = \frac{2(\gamma r - r - \lambda)}{\gamma} \\ c &= \frac{(\lambda + r)^2}{\gamma\sigma^2}, \quad d = \frac{\lambda\mu}{\gamma}, \quad g = -\frac{\lambda\mu(\lambda + r)}{\gamma\sigma^2}. \end{aligned} \tag{11}$$

Notice that the first equation in (10) only contains $C(\tau)$, the second contains $B(\tau)$ and $C(\tau)$ and the third contains all of the $A(\tau)$, $B(\tau)$ and $C(\tau)$. Hence we can solve them sequentially.

THEOREM 3.1 *Given risky asset price X follows* (1), *fixed risk free rate r, and investment horizon* τ, *the optimal asset allocation to risky asset that maximizes the expected CRRA utility of terminal wealth* $\mathbb{E}[U(W_T)]$ *is*:

$$\begin{aligned} \pi^*(\tau, X) = \frac{1}{\gamma}\left[\frac{\lambda(\mu - X)/X - r}{(\sigma/X)^2}\right] \\ + \frac{1-\gamma}{\gamma}[C(\tau)X + B(\tau)]X, \end{aligned} \tag{12}$$

where

$$\begin{aligned} C(\tau) &= \frac{2c(1 - e^{-\eta\tau})}{2\eta - (b+\eta)(1 - e^{-\eta\tau})}, \\ B(\tau) &= \frac{-4gr(1 - e^{-\eta\tau/2})^2 + 2g\eta(1 - e^{-\eta\tau})}{\eta[2\eta - (b+\eta)(1 - e^{-\eta\tau})]}, \\ A(\tau) &= \int_0^\tau \frac{a}{2}B(t)^2 + dB(t) + \frac{\sigma^2}{2}C(t) + \frac{(\lambda\mu)^2}{2\gamma\sigma^2} + r\,\mathrm{d}t \end{aligned} \tag{13}$$

with $\eta = \sqrt{b^2 - 4ac}$ *and parameters a, b, c, d, g described in* (11).

Notice, we leave $A(\tau)$ in the integral form since this term does not appear in our optimal investing proportion π^*, nevertheless, the integral can be computed by checking the integral table. We include the detailed proof and computation in Appendix 1. Here, again, the first term is equation (13) is the myopic demand. Since the expected return of the risky asset is $\lambda(\mu - X)/X$, this term is exactly the optimal solution for a single-period model. The larger γ is, the more risk averse the investor is and therefore the less exposure to the risky asset. The second term represents the inter-temporal hedging demand, which is related to the current price level, mean-reversion speed as well as risk aversion coefficient.

3.2. Optimal strategy with multiple risky assets

In this section, we further extend our analysis to multiple risky assets. Suppose there are n risky assets, which jointly follow the multi-dimensional OU process:

$$\mathrm{d}\mathbf{X} = \mu_X(\mathbf{X})\,\mathrm{d}t + \sigma_X\,\mathrm{d}\mathbf{Z}, \tag{14}$$

where:

$$\begin{aligned} &\mu_X(\mathbf{X}) = \Lambda(M - \mathbf{X}) \\ &\Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix}, \quad M = (\mu_1, \mu_2, \ldots, \mu_n)^T, \\ &\sigma_X = \begin{pmatrix} \sigma_1 & & 0 \\ & \sigma_2 & \\ & \ddots & \\ 0 & & \sigma_n \end{pmatrix}. \end{aligned} \tag{15}$$

$\mathbf{Z}$ is a vector of n correlated Brownian motions. The correlation matrix of the risk factors is characterized by:

$$\mathrm{d}\mathbf{Z}\,\mathrm{d}\mathbf{Z}^T = \sigma_\rho\,\mathrm{d}t = \begin{pmatrix} 1 & \rho_{12} & \cdots & \rho_{1n} \\ \rho_{21} & 1 & & \rho_{2n} \\ \cdots & & \ddots & \\ \rho_{n1} & \rho_{n2} & & 1 \end{pmatrix}\mathrm{d}t. \tag{16}$$

Y is the value of risk-free money market account with a deterministic growth rate r:

$$\mathrm{d}Y = rY\,\mathrm{d}t. \tag{17}$$

Let $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_n) \in \mathbb{R}^n$ be the portfolio weights invested in $\mathbf{X} = (X_1, X_2, \ldots, X_n)$. Then the total wealth satisfies:

$$\frac{\mathrm{d}W}{W} = [\boldsymbol{\pi}^T\mu_W + r(1 - \boldsymbol{\pi}^T\not\vdash)]\,\mathrm{d}t + \boldsymbol{\pi}^T\sigma_W\,\mathrm{d}\mathbf{Z}, \tag{18}$$

where

$$\begin{aligned} \mu_W &= \left[\frac{\lambda_1}{X_1}(\mu_1 - X_1), \frac{\lambda_2}{X_2}(\mu_2 - X_2), \ldots, \frac{\lambda_n}{X_n}(\mu_n - X_n)\right] \\ \sigma_W &= \begin{pmatrix} \sigma_1/X_1 & & 0 \\ & \sigma_2/X_2 & \\ & \ddots & \\ 0 & & \sigma_n/X_n \end{pmatrix}. \end{aligned} \tag{19}$$

Following the aforementioned method, we assume $J(W, \mathbf{X}, \tau)$ to be the indirect utility function, where τ is the horizon of the

investment period. We can derive the HJB equation:

$$\max_{\boldsymbol{\pi}} \left\{ -J_\tau + \frac{1}{2} W^2 \boldsymbol{\pi}^T \Sigma_W \boldsymbol{\pi} J_{WW} + W[\boldsymbol{\pi}^T(\mu_W - r\not{1}) + r] J_W + W\boldsymbol{\pi}^T \Sigma_{WX} J_{W\mathbf{X}} + {\mu_X}^T J_{\mathbf{X}} + \frac{1}{2}\mathrm{tr}(\Sigma_X J_{\mathbf{X}\mathbf{X}^T}) \right\} = 0, \tag{20}$$

where

$$\begin{aligned} \Sigma_W &= \sigma_W \sigma_\rho \sigma_W^T \\ \Sigma_{WX} &= \sigma_W \sigma_\rho \sigma_X^T \\ \Sigma_X &= \sigma_X \sigma_\rho \sigma_X^T. \end{aligned} \tag{21}$$

similar to single asset case, J_τ, J_W, and $J_{\mathbf{X}}$ denote the derivatives of J with respect to t, W, and $\mathbf{X}$, respectively; J_{WW}, $J_{W\mathbf{X}}$ and $J_{\mathbf{X}\mathbf{X}^T}$ denote the higher derivatives; $\mathrm{tr}(\cdot)$ denotes the trace function. The optimal trading strategy is:

$$\begin{aligned} \boldsymbol{\pi}^*(\tau, \mathbf{X}, W) = &-\frac{J_W}{W J_{WW}} {\Sigma_W}^{-1}(\mu_W - r\not{1}) \\ &- \frac{1}{W J_{WW}} {\Sigma_W}^{-1} \Sigma_{WX} J_{WX}. \end{aligned} \tag{22}$$

Now suppose the solution of $J(W, \mathbf{X}, \tau)$ takes the form:

$$\begin{aligned} J(W, \mathbf{X}, \tau) &= \frac{(W\phi(\tau, \mathbf{X}))^{1-\gamma} - 1}{1-\gamma} \\ \phi(\tau, \mathbf{X}) &= \exp\left\{ A(\tau) + B(\tau)^T \mathbf{X} + \frac{1}{2} \mathbf{X}^T C(\tau) \mathbf{X} \right\}. \end{aligned} \tag{23}$$

The format of $\phi(\tau, \mathbf{X})$ is a natural extension of equation (8) in the single asset case to higher dimensions. $A(\tau)$ is a scalar, $B(\tau)$ is a n-by-1 vector and $C(\tau)$ is a n-by-n matrix. substituting $J(W, \mathbf{X}, \tau)$ and matching the orders, we can obtain the ODE systems for $A(\tau)$, $B(\tau)$ and $C(\tau)$.

THEOREM 3.2 *Given risky asset prices* $\mathbf{X}$ *follow* (14)–(16), *fixed risk free rate* r, *and investment horizon* τ, *the optimal asset allocation to risky assets that maximizes the expected CRRA utility of terminal wealth* $\mathbb{E}[U(W_T)]$ *is*:

$$\begin{aligned} \boldsymbol{\pi}^*(\tau, \mathbf{X}) = &-\frac{1}{\gamma} {\Sigma_W}^{-1} \{ (r\not{1} - \mu_W) \\ &- (1-\gamma) \Sigma_{WX} [B(\tau) + C(\tau)\mathbf{X}] \}, \end{aligned} \tag{24}$$

where

$$\begin{aligned} C'(\tau) = &\frac{1-\gamma}{\gamma} C(\tau)^T \Sigma_X C(\tau) - \frac{1}{\gamma}[\Lambda C(\tau) + C(\tau)^T \Lambda] \\ &- \frac{2r(1-\gamma)}{\gamma} C(\tau) + \frac{1}{\gamma}(\Lambda + r\mathcal{I})^T \Sigma_X^{-1} (\Lambda + r\mathcal{I}) \end{aligned} \tag{25}$$

$$\begin{aligned} B'(\tau) = &\left[\frac{1-\gamma}{\gamma} \Sigma_X C(\tau) - \frac{1}{\gamma}\Lambda - \frac{1-\gamma}{\gamma} r\mathcal{I} \right]^T B(\tau) \\ &- \frac{1}{\gamma}(\Lambda + r\mathcal{I})^T \Sigma_X^{-1} \Lambda M + \frac{1}{\gamma} C(\tau)^T \Lambda M \end{aligned} \tag{26}$$

$$\begin{aligned} A'(\tau) = &\frac{1}{2\gamma}(\Lambda M)^T \Sigma_X^{-1} (\Lambda M) + \frac{1}{\gamma} M^T \Lambda B(\tau) \\ &+ \frac{1-\gamma}{2\gamma} B(\tau)^T \Sigma_X B(\tau) + r + \frac{1}{2}\mathrm{Tr}[\Sigma_X C(\tau)] \end{aligned} \tag{27}$$

with boundary conditions:

$$A(0) = 0, \quad B(0) = 0, \quad C(0) = 0 \tag{28}$$

where $\not{1}$ *is the n-by-1 all-ones vector and* $\mathcal{I}$ *is the n-by-n identity matrix.*

The detailed proof and computation is included in the Appendix 2. The ODE system (25)–(28) does not admit a closed-form solution, yet we can discretize the ODEs and simulate them using initial conditions. In particular, at t_{n+1} we approximate $C_{t_{n+1}}$ using (25), we approximate $B_{t_{n+1}}$ using B_{t_n} and C_{t_n} and (26), we approximate $A_{t_{n+1}}$ using B_{t_n} and C_{t_n} and (27).

3.3. *Proportional transaction cost*

Now we assume a proportional transaction cost occurs at each transaction (buy or sell of the risky assets). Assuming an investor has a portfolio $\mathbf{X} = (X_1, X_2, \ldots, X_n)$, where X_i is the dollar values in ith risky asset and X_0 is the dollar values in the bank account. In the presence of transaction costs, the dynamics follow:

$$\begin{aligned} \mathrm{d}X_0 &= rX_0\,\mathrm{d}t - \sum_{i=1}^{n}(1+l_i)\,\mathrm{d}L_i + \sum_{i=1}^{n}(1-m_i)\,\mathrm{d}M_i \\ \mathrm{d}X_i &= \mu_i(X_i)\,\mathrm{d}t + \sigma_i(X_i)dB_i + \mathrm{d}L_i - \mathrm{d}M_i \end{aligned} \tag{29}$$

$L_i(t)$ and $M_i(t)$ are nondecreasing, right continuous adapted processes with $L_i(0) = M_i(0) = 0$, representing cumulative dollar values of buying and selling the ith risky asset. $l_i \in [0, \infty)$ and $m_i \in [0, 1)$, $i = 1, 2, \ldots, n$, accounts for the proportional transaction costs incurred in buying and selling the ith risky asset. The solvency region $\mathcal{S}$ is defined as the set of dollar amounts in each assets such that the net wealth is always positive:

$$\begin{aligned} \mathcal{S} = \Bigg\{ &x = (x_0, x_1, \ldots, x_n) \in \mathbb{R}^{n+1} : x_0 \\ &+ \sum_{i=1}^{n} \left[(1-m_i)x_i^+ - (1+l_i)x_i^- \right] > 0 \Bigg\} \end{aligned} \tag{30}$$

Given initial positions $x^0 = (x_0, x_1, \ldots, x_n) \in \mathcal{S}$ in the risk-free and risky assets, respectively, the trading strategy $(\{L_i\}, \{M_i\})$ is admissible for a position x from time $s \in [0, T)$, if X_i, $i = 0, \ldots, n$ follows (29) with $X_s = x$ is in $\mathcal{S}$. We denote by $A_s(x)$ the set of all admissible investment strategies for x from time s. Finally, the investor's goal is to maximize the utility of terminal wealth among all admissible strategies:

$$\sup_{(L_i, M_i) \in \mathcal{A}_0(x^0)} \mathbb{E}[U(W_T)] \tag{31}$$

Following Davis *et al.* (1993), Shreve and Soner (1994), Muthuraman and Zha (2008) or Dai and Zhong (2010), the solvency region can be divided into three type of regions: In the no-trade-zone (NT zone, defined as none of the assets

trade), the indirect utility function $J(W, \mathbf{X}, \tau)$ satisfies the following HJB equation:

$$-\frac{\partial J}{\partial \tau} + \mathscr{L} J = 0. \tag{32}$$

In the ith asset's buy-zone (BZ_i), the marginal cost of decreasing the amount in the bank account must be equal to the marginal benefit of buying the risky asset:

$$\mathscr{B}_i J = 0, \quad \text{for } i = 1, 2, \ldots, n \tag{33}$$

In the ith asset's sell-zone (SZ_i), the marginal benefit of increasing the amount in the bank account must be equal to the marginal cost of selling the risky asset:

$$\mathscr{S}_i J = 0, \quad \text{for } i = 1, 2, \ldots, n \tag{34}$$

with terminal condition:

$$J(x, T) = U\left\{x_0 + \sum_{i=1}^{n}\left[(1 - m_i)x_i^+ - (1 + l_i)x_i^-\right]\right\} \tag{35}$$

where

$$\begin{aligned}
\mathscr{L} &= \frac{1}{2}\sum_{i,j=1}^{n} \rho_{ij}\sigma_i(x_i)\sigma_j(x_j)\frac{\partial^2 J}{\partial x_i \partial x_j} + \sum_{i=1}^{n}\mu_i(x_i)\frac{\partial J}{\partial X_i} + r x_0 \frac{\partial J}{\partial X_0} \\
\mathscr{B}_i &= -(1 + l_i)\frac{\partial J}{\partial x_0} + \frac{\partial J}{\partial x_i} \\
\mathscr{S}_i &= (1 - m_i)\frac{\partial J}{\partial x_0} - \frac{\partial J}{\partial x_i}
\end{aligned} \tag{36}$$

The problem becomes a free-boundary problem, all that needs to be found are the boundaries of the regions such that the respective equations hold within the regions and the following equation holds in the entire state space:

$$\max\left\{-\frac{\partial J}{\partial \tau} + \mathscr{L} J, \mathscr{B}_i J, \mathscr{S}_i J\right\} = 0, \quad x \in \mathscr{S},\ t \in [0, T) \tag{37}$$

Previous research focuses on numerical implementation of PDE (33)–(37) as analytical solutions are difficult to obtain for assets with mean-reverting characteristics. A similar but easier problem of proportional transaction costs with assets following geometric Brownian motion is well-studied in early papers such as Taksar *et al.* (1988). They are able to use the PDE method described in this section to find the optimal no-trade zone for the case where the underlying risky assets follow geometric Brownian motion. The no-trade zone is shown to have a wedge shape around the optimal solution under zero transaction costs. This strategy enables an investor to stay close enough to the optimal solution without suffering from too much transaction costs. Given the difficulties of solving this PDE directly, in this paper, we propose a different approach of learning the two boundaries r_u^i and r_l^i for each asset i by neural network. The parameterization and training details are elaborated in Section 3.

4. Learning the no-trade zone using deep neural networks

Taking advantage of mean-reverting dynamics, we aim to simultaneously long and short several correlated assets that are tradable on the market. However, one has to pay transaction costs for all the underlying tradables. When transaction costs are significant, the analytic strategies we derived in the previous section are no longer appropriate. However, they may be used as an initial point for us to search for the optimal strategy under transaction costs. On one hand, one should keep her position close to the analytic solution in order to maximize the terminal utility; on the other hand, consistently rebalancing to the analytic solution makes one worse off with the expressly high cumulative transaction costs.

Most existing studies on optimal asset allocation with transaction costs (see Bichuch and Sircar 2014, 2018, Liu and Loewenstein 2002) assume that asset prices follow the geometric Brownian motion (GBM), for the one-asset case the problem comes down to solving a HJB equation with two unknown boundaries (free boundaries). The region between the two boundaries is called the *no-trade (NT) zone*. If current holding position is in NT zone then no action is required, otherwise one should rebalance positions to the nearest point on the NT zone.

In order to compute the NT zone when there is transaction cost, we will use deep neural network (DNN) to approximate the upper and lower boundaries of the NT zone. A deep neural network is an artificial neural network with many layers between the input layer and the output layer. It typically has a feedforward structure, i.e. without looping back to previously visited layers. Each neuron gets inputs from previous layer, takes the weighted sum of the inputs, and transforms it through some function. By optimizing the weights over the inputs, deep neural networks are able to manipulate the desired outputs.

Our goal is to solve a sequential decision problem in the context of optimal asset allocation. We employ the deep neural network to learn the optimal NT zone under the assumption of mean-reversion and linear transaction cost. We pick the loss function to be the difference between expected terminal utility under the policy we derived analytically in previous section and policy output by the DNN. By minimizing loss we are able to train the neural network to better approximate boundaries of the NT zone. When there is transaction cost, the policy learned by DNN can yield significantly higher terminal utility and wealth than the analytical policy we derived in Section 2.

4.1. Single asset no-trade zone parameterization

When the asset price follows a geometric Brownian motion and the utility function is CRRA, Liu and Loewenstein (2002) showed that the lower boundary r_l and upper boundary r_u of the NT zone are functions of time, they take the form $r_l(t)$ and $r_u(t)$. The values of r_l and r_u denote the proportion of total wealth to invest in the risky asset. However, when the asset price is mean reverting, these two boundaries should also be functions of asset price and take the form $r_l(t, X_t)$, $r_u(t, X_t)$.

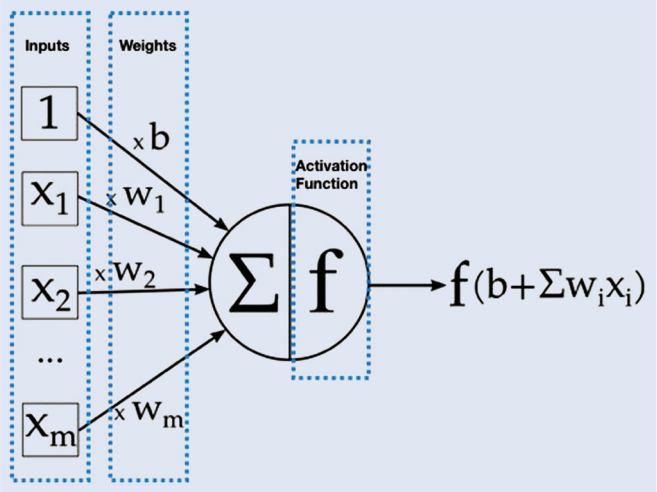

Figure 2. A single layer feedforward neural network $f_\theta(x)$. The input is a m-dimensional vector $(x_1, \ldots x_m)$. The parameters θ to be estimated are $w_1 \cdots w_m$ and b. The activation function is f which is usually taken as a Sigmoid function or Rectified Linear Unit(ReLU). The output of the network is $f(b + \sum w_i x_i)$.

The exact forms of $r_l(t, X_t)$ and $r_u(t, X_t)$ are unknown. We use a neural network to parametrize these two functions. A neural network can be regarded as a non-linear parametrization of a function. It has been shown that a feedforward neural network with a single hidden layer could approximate arbitrary functions to high accuracy as long as the number of neurons is large enough (see Hornik *et al.* 1990). Such a network is used as a non-parametric method in statistics and time series analysis (see Tsay 2010). More recently, deep (i.e. many layered) neural networks have been used to approximate derivatives of unknown functions and for solving high dimensional non-linear partial differential equations (see Beck *et al.* 2019, Weinan *et al.* 2017, Han *et al.* 2018, Sirignano and Spiliopoulos 2017). For more details of feedforward neural networks, please refer to Chapter 6 of Goodfellow *et al.* (2016). When used for function approximation, we usually feed the neural network with training data $\{x_i, y_i\}$ where y_i is the target function value corresponding to independent variable x_i. Iterative methods such as stochastic gradient descent are employed to minimize the error (for example, least squares error) between the target function value y_i and the output of the neural network $f_\theta(x_i)$. The goal is to find the best set of parameters θ that minimize the loss associated with the training data. An example of a single layer neural network is given in Figure 2.

Suppose the investment horizon is $[0, T]$. We discretize the time horizon into N sub-intervals of equal length $\Delta t = t_k - t_{k-1} = T/N$. At time t_k, the lower and upper boundaries of the NT zone are only a function of X_{t_k} because t is held fixed at $t = t_k$. Denoting the analytic policy we derived for the no transaction cost case by $\pi^*(t, x)$, we can parametrize the lower and upper boundaries at t_k by neural networks:

$$r_u^{t_k}(x) \approx \pi^*(t_k, x) + f_{\theta_{t_k}^u}(x) \tag{38}$$

$$r_d^{t_k}(x) \approx \pi^*(t_k, x) - f_{\theta_{t_k}^d}(x) \tag{39}$$

Importantly, the parametrization of $r_u^{t_k}(x)$ and $r_d^{t_k}(x)$ makes use of prior knowledge from the analytic policy $\pi^*(t_k, x)$ derived under the zero transaction cost assumption. Intuitively, when there is no transaction cost, we should always rebalance our position to the curve defined by $\pi^*(t_k, x)$ and this means $r_u^{t_k}(x) = r_d^{t_k}(x) = \pi^*(t_k, x)$. With increasing transaction costs, the NT zone should widen and $\pi^*(t_k, x)$ should fall into the NT zone, this amounts to saying that $f_{\theta_{t_k}^u}(x)$ and $f_{\theta_{t_k}^d}(x)$ should take small positive values. When implementing the algorithm, we heuristically initialize the output of $f_{\theta_{t_k}^u}(x)$ and $f_{\theta_{t_k}^d}(x)$ to be positive, but we also allow them to take negative values. This is achieved by using a leaky ReLU function (Xu *et al.* 2015) on the output layer. Such a parametrization guides the neural network to search for solutions within a relatively small region where the output takes small positive numbers. By contrast, if one does not incorporate the prior knowledge $\pi^*(t_k, x)$ and directly parametrizes the lower and upper boundary by

$$r_u^{t_k}(x) \approx f_{\theta_{t_k}^u}(x)$$
$$r_d^{t_k}(x) \approx f_{\theta_{t_k}^d}(x),$$

we have observed in our experiments that it takes much more time to train the neural network. With the parametrization (38)–(39) using a warm start and prior knowledge, we narrow down the search region and reduce the likelihood of getting trapped in a local optimal solution.

4.2. *One-asset trading strategy parameterization*

In the remainder of the paper, we constrain the range of π hence $r_l^{t_k}$ and $r_u^{t_k}$ to $[-1, 1]$ because risky asset rebalancing happens at discrete time points. This constraint could be relaxed as rebalancing becomes more frequent, in particular, there is no constraint on π in Section 2 where rebalancing is continuous.

When we arrive at time t_k, the proportion we invested in the risky asset is π_{t_k-}, we rebalance this proportion to π_{t_k+} using the policy:

$$\pi_{t_k+} = \begin{cases} r_l^{t_k}(X_{t_k}) & \pi_{t_k-} < r_l^{t_k}(X_{t_k}) \\ \pi_{t_k-} & r_l^{t_k}(X_{t_k}) \le \pi_{t_k-} \le r_u^{t_k}(X_{t_k}) \\ r_u^{t_k}(X_{t_k}) & \pi_{t_k-} > r_u^{t_k}(X_{t_k}) \end{cases} \tag{40}$$

We require $\pi_{T+} = 0$, i.e. we need to clear all the risky asset position at the end of investment horizon. However, it is easy to change the end-horizon position to any target asset mix. See Grinold (1983) for more details.

We explain the policy (40) involving three different cases:

- When the proportion π_{t_k-} invested in risky asset is below the lower boundary $r_l^{t_k}(X_{t_k})$ of the NT zone, we buy more of the risky asset to rebalance our position π_{t_k+} to the lower boundary of the no trade zone.
- When $r_l^{t_k}(X_{t_k}) \le \pi_{t_k-} \le r_u^{t_k}(X_{t_k})$, our position π_{t_k-} is above the lower boundary $r_l^{t_k}(X_{t_k})$ and below the upper boundary $r_u^{t_k}(X_{t_k})$. We are in the NT zone, so no action is needed and $\pi_{t_k+} = \pi_{t_k-}$.

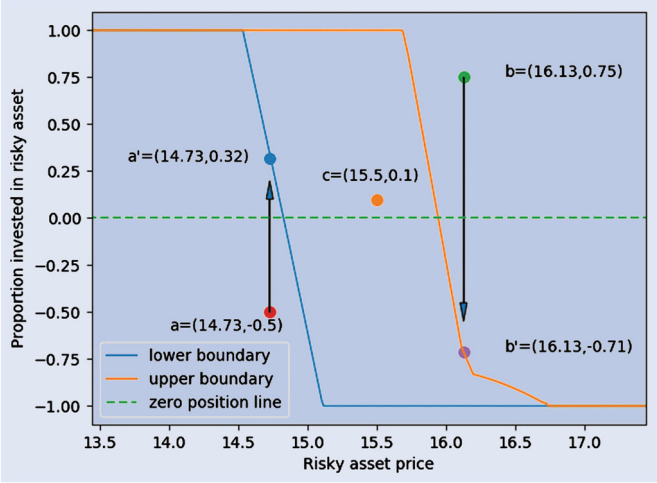

Figure 3. A typical no trade zone and corresponding rebalance operations. The mean-reverting level is 15.4463 and transaction cost is 2.0%. (i) From point a to a': point a is outside the NT zone, we rebalance it to point a'. We convert the position from investing 50% of wealth in shorting asset to investing 32% of wealth in longing the asset because the current observed asset price is 14.73 which is way below the mean-reverting level 15.4463. (ii) From b to b': b is outside the NT zone, we rebalance it to b'. We convert the position from investing 75% of wealth in longing asset to investing 71% of wealth in shorting the asset because the current observed asset price is 16.13 which is way above the mean-reverting level 15.4463. (iii) At c: because point c is within NT zone, no rebalance operation is needed.

- When $\pi_{t_k-} > r_u^{t_k}(X_{t_k})$, our position in the risky asset is above the upper boundary so we need to sell some risky asset to rebalance our position π_{t_k+} to the upper boundary of the NT zone.

At a fixed time t_k, a typical no trade zone and its induced policy is given in Figure 3:

The NT zone in Figure 3 is simulated with parameters estimated from VIX futures in year 2013–2014 and the assumption that there is a 2% transaction cost. The NT zone is for $t_k = 0.4$ when investment horizon is $[0, T] = [0, 1]$. The blue curve denotes the lower boundary of NT zone and orange curve is the upper boundary of the NT zone. The region between two curves is the NT zone. The mean-reverting level is 15.4463, and the three scattered points a, b and c correspond to the three cases we discussed following: (40)

- Red point $a = (14.73, -0.5)$. This point means $X_{t_k} = 14.73$ and the proportion π_{t_k-} we invested in risky asset is -0.5, i.e. we are using 50% of our wealth to short the asset. However, 14.73 is below the mean-reverting level 15.4463 so we should switch our position to hold the asset, and we rebalance our position to the blue point $a' = (14.73, 0.32)$ on the lower boundary of the NT zone, this means we clear our short position and put $\pi_{t_k+} = 32\%$ of our wealth to hold the asset.
- Orange point $c = (15.5, 0.1)$. This point means $X_{t_k} = 15.5$ and $\pi_{t_k-} = 0.1$. This point lies in the NT zone so no action is needed. Because 15.5 is close to the mean-reverting level, the price change in the near future will be mainly driven by noise, there is no deterministic trend, hence to avoid transaction cost we should not rebalance.
- Green point $b = (16.13, 0.75)$. This point means $X_{t_k} = 16.73$ and $\pi_{t_k-} = 0.75$, we are using 75% of our wealth to hold the asset. Because 16.73 is well above the mean-reverting level, we should switch our long position to a short position $\pi_{t_k+} = -0.71$, i.e. putting 71% of our wealth to shorting the asset. We move from the green point b to the purple point b' on the upper boundary of the NT zone.

Note that if we start from a point out of the NT zone, we only rebalance to the boundaries of the NT zone. To prove its optimality, let us suppose there is another point $a'' = (14.73, \pi'')$ right above point a' and inside the NT zone. Since the transaction cost is linear, it is equivalent to (i) rebalance from point a to point a'' and to (ii) rebalance from point a to point a' and then immediately rebalance from point a' to point a''. By definition of NT zone, there is no benefit rebalancing from point a' to point a'', and therefore rebalancing to the boundary of NT zone is optimal.

4.3. Training DNN for one asset

Assuming that in the deep neural network there is a linear transaction cost rate α, then the transaction cost from each rebalancing operation is given by:

$$c_{t_k} = \alpha W_{t_k} |\pi_{t_k+} - \pi_{t_k-}|. \tag{41}$$

According to continuous time dynamics formula (1) and (2) for X, Y and W together with the transaction cost formula (41), the system dynamics from t_{k-1} to t_k are given by:

$$\Delta_{t_{k-1}} X = X_{t_k} - X_{t_{k-1}} = (e^{-\lambda h}) X_{t_{k-1}} + \mu(1 - e^{-\lambda h}) + N(t, h) \tag{42}$$

$$\Delta_{t_{k-1}} Y = Y_{t_k} - Y_{t_{k-1}} = e^{rt}(e^{rh} - 1) \tag{43}$$

$$\Delta_{t_{k-1}} W = W_{t_k} - W_{t_{k-1}} = \frac{\pi_{t_{k-1}^+} W_{t_{k-1}}}{X_{t_{k-1}}} \Delta_{t_{k-1}} X + \frac{(1 - \pi_{t_{k-1}}^+) W_{t_{k-1}}}{Y_{t_{k-1}}} \Delta_{t_{k-1}} Y - c_{t_{k-1}}, \tag{44}$$

where

$$N(t, h) := \sigma e^{-\lambda(t+h)} \int_t^{t+h} e^{\lambda u} \, \mathrm{d}Z_u \sim N\left(0, \frac{\sigma^2(1 - e^{-2\lambda h})}{2\lambda}\right)$$

$$h := t_k - t_{k-1} = \frac{T}{N}.$$

At each training step, we start at $t_0 = 0$ with n sample paths initialized as:

$$X_0 = [x_0, \ldots, x_0]_{1\times n}^T$$

$$W_0 = [w_0, \ldots, w_0]_{1\times n}^T$$

$$\pi_{0-} = [0, \ldots, 0]_{1\times n}^T.$$

At each time step t_k our position rebalances from the output of the neural network $r_u^{t_k}(x)$ and $r_d^{t_k}(x)$ together with the

rebalancing rule (40). We then move to the next time step t_{k+1} using equations (42)–(44).

Algorithm 1: resulting trading algorithm

initialize $X_{t_0}, Y_{t_0}, W_{t_0}, \pi_{t_0-}$;
$k = 0$;
while $k < T$ **do**
  $\pi_{t_k+} = min\{r_u^{t_k}(X_{t_k}), max\{r_l^{t_k}(X_{t_k}), \pi_{t_k-}\}\}$;
  $k = k + 1$;
  Simulate X_{t_k} based on Eq.(42);
  Simulate Y_{t_k} based on Eq.(43);
  Simulate W_{t_k} based on Eq.(44)
end

At terminal time T, the $n \times 1$ vector W_T represents the terminal wealth over n sample paths, and the empirical loss over n sample paths is defined as:

$$\text{loss} \triangleq \frac{1}{n}\sum_{i=1}^{n}(J(w_0, x_0, T) - U(W_T^{(i)})), \tag{45}$$

where $J(W_0, X_0, T)$ is the value function given by equation (7)–(8) and equation (13).

The computational graph defines the data-flow of the deep neural network (Figure 4). In the computational graph, the 'Rebalance Rule' follows equation (40), and the 'System Dynamics' follows equations (42)–(44).

Similar to the network structure as described in Han and Weinan (2016), there are three types of connections in our computational graph:

- At each time step, there are two parallel feedforward neural networks that approximate the lower bound $r_l^t(x)$ and upper bound $r_u^t(x)$ of the NT zone, respectively. They are captured in the computational graph as $(W_{t-}, \pi_{t-}, X_t) \to h_{t,l}^1 \to h_{t,l}^2 \to \cdots \to h_{t,l}^K \to r_l^t$, and $(W_{t-}, \pi_{t-}, X_t) \to h_{t,u}^1 \to h_{t,u}^2 \to \cdots \to h_{t,u}^K \to r_u^t$, where the outputs r_l^t and r_l^t are the calibrated rebalancing rules. The weights in these neural networks are the objects that we aim to optimize over so as to minimize the loss function.
- Using the rebalance rules and the position and wealth before rebalancing, we calculate the position and wealth after rebalancing at time t. In the computational graph, this type of connection is shown as $((W_{t-}, \pi_{t-}, X_t), r_l^t, r_u^t) \to (W_{t+}, \pi_{t+})$. Given the position and wealth before rebalancing as well as the rebalancing rules, there is neither randomization nor optimization involved in this step.
- The last type of connection relates to the system dynamics, i.e. the simulation of the price of the risky asset. In particular, $X_t \to X_{t+1}$ takes use of equation (42) to simulate the price path of the risky asset. Then, we fully characterize the position, wealth, and price of the risky asset at time $t + 1$ following the system dynamics described above, as captured by $((W_{t+}, \pi_{t+}), X_t, X_{t+1}) \to (W_{(t+1)-}, \pi_{(t+1)-})$ in the computational graph.

We follow this procedure until the investment horizon T. At horizon T, the neural network outputs the loss function which we aim to minimize. Suppose that at every time step, there are K hidden layers in each feedforward subnetwork, then there are in total $(K + 2)N$ layers in the whole network structure.

4.4. *Parameterizing multi-assets NT zone and trading strategy*

If there are d risky assets, and the asset price at time t_k is $\mathbf{X}_{t_k} \in \mathbb{R}^d$, the upper and lower boundaries of the NT zone

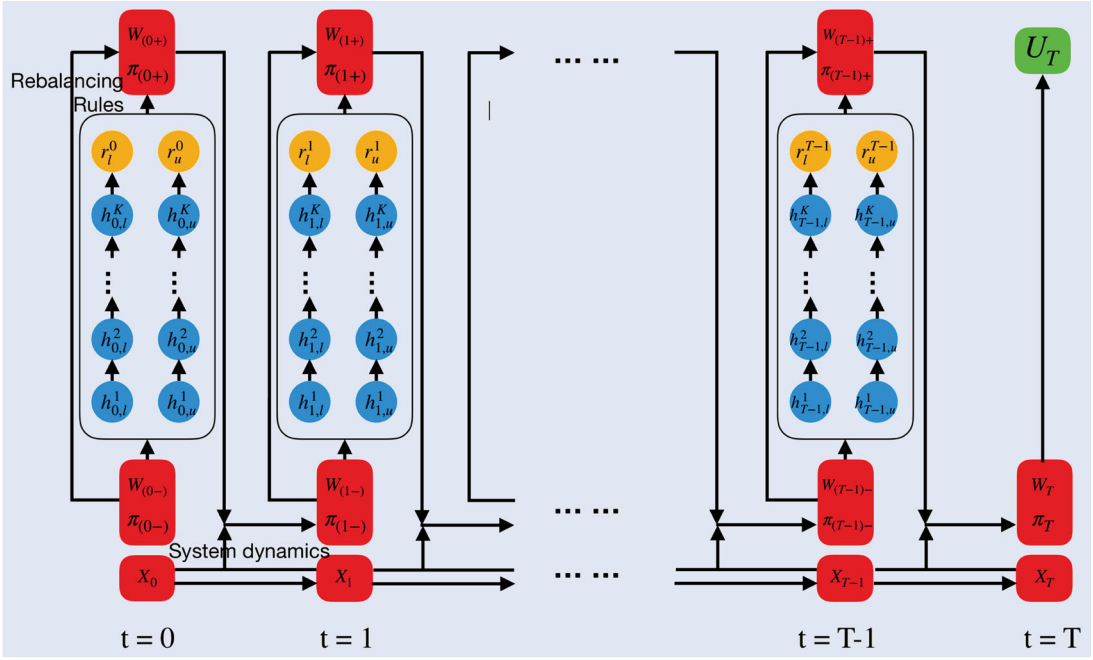

Figure 4. Computational graph of the deep neural network.

is characterized by two d-dimensional vectors $r_u^{t_k}(\mathbf{X}_{t_k})$ and $r_d^{t_k}(\mathbf{X}_{t_k}) \in \mathbb{R}^d$. We parametrize the two boundaries by neural networks like we did for the one-asset case, except that the output of the neural network is d-dimensional:

$$r_u^{t_k}(\mathbf{X}) \approx \pi^*(t_k, \mathbf{X}_{t_k}) + f_{\theta_{t_k}^u}(\mathbf{X}) \in \mathbb{R}^d \tag{46}$$

$$r_d^{t_k}(\mathbf{X}) \approx \pi^*(t_k, \mathbf{X}_{t_k}) - f_{\theta_{t_k}^d}(\mathbf{X}) \in \mathbb{R}^d. \tag{47}$$

The rebalance rule is similar to the rule (40) for one-asset case, where here the subscript (i) denotes the ith component:

$$\pi_{t_k+}^{(i)} = \begin{cases} r_l^{t_k}(\mathbf{X}_{t_k})^{(i)} & \pi_{t_k-}^{(i)} < r_d^{t_k}(X_{t_k})^{(i)} \\ \pi_{t_k-}^{(i)} & r_l^{t_k}(\mathbf{X}_{t_k})^{(i)} \le \pi_{t_k-}^{(i)} \le r_u^{t_k}(\mathbf{X}_{t_k})^{(i)} \\ r_u^{t_k}(\mathbf{X}_{t_k})^{(i)} & \pi_{t_k-}^{(i)} > r_u^{t_k}(X_{t_k})^{(i)}. \end{cases} \tag{48}$$

Similarly to the one-asset case, we add a leverage ratio constraint: $\|\pi_{t+}\|_\infty \le 1$, i.e. the leverage ratio used on each individual asset should not exceed a $\pm 100\%$ limit.

4.5. Training the DNN for multiple assets

The computational graph for the multi-asset case is exactly the same as for the one-asset case in Figure 4. We give next the environment dynamics in this computational graph. We follow Wan (2010) to calibrate the parameters of the mean-reverting process. The form of the OU process given by equations (14) and (16) simplifies the procedure to derive (25)–(27), but it is hard to calibrate σ_X and σ_ρ separately. In practice, we calibrate another matrix Σ_{OU} which contains the information from σ_X and σ_ρ. We rewrite the dynamics of the asset price $\mathbf{X}$ as:

$$\mathrm{d}\mathbf{X} = \mu_X(\mathbf{X})\,\mathrm{d}t + \Sigma_{\text{OU}}\,\mathrm{d}W \tag{49}$$

where $\mathrm{d}W$ is the differential of a d-dimensional Brownian motion $\mathbf{W}$ starting at 0 and satisfying the uncorrelated-increment condition:

$$\mathrm{d}\mathbf{W}\,\mathrm{d}\mathbf{W} = \mathbf{I}_{d\times d}$$

The following lemma proves the equivalence between (49) and (14) and shows how to rewrite the optimal policy in terms of Σ_{OU}. Then it is sufficient to calibrate Σ_{OU} rather than calibrating σ_X and σ_ρ. The introduction of σ_X and σ_ρ simplifies the derivation of the ODEs (25)–(27). Similarly, the environment dynamics could be written using Σ_{OU} via equation (49) rather than (18).

LEMMA 4.1 *Let A be the Cholesky decomposition of σ_ρ, if $\Sigma_{\text{OU}} = \sigma_X A$, then SDE of* (49) *and* (14) *has same distribution. The optimal policy still follows* (24) *provided we can rewrite the Σ_W, Σ_{WX} and Σ_X in terms of Σ_{OU} as*:

$$\Sigma_W = \begin{pmatrix} \frac{1}{X_1} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{X_d} \end{pmatrix} \Sigma_{\text{OU}}\Sigma_{\text{OU}}^T \begin{pmatrix} \frac{1}{X_1} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{X_d} \end{pmatrix} \tag{50}$$

$$\Sigma_{WX} = \begin{pmatrix} \frac{1}{X_1} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{X_d} \end{pmatrix} \Sigma_{\text{OU}}\Sigma_{\text{OU}}^T \tag{51}$$

$$\Sigma_X = \sigma_X A A^T \sigma_X = \Sigma_{\text{OU}}\Sigma_{\text{OU}}^T. \tag{52}$$

Proof By equation (16) we have:

$$\mathrm{d}\mathbf{Z}\,\mathrm{d}\mathbf{Z} = \sigma_\rho\, r dt.$$

We also have:

$$\mathrm{d}\mathbf{W}\,\mathrm{d}\mathbf{W} = \mathbf{I}_{d\times d}\,\mathrm{d}t.$$

Hence given A as the Cholesky decomposition of σ_ρ, $AA^T = \sigma_\rho$, we have:

$$\mathrm{d}\mathbf{Z} \overset{d}{=} A\,\mathrm{d}\mathbf{W}.$$

where $\overset{d}{=}$ means equal in distribution. Matching the term $\Sigma_{\text{OU}}, \mathrm{d}\mathbf{W}$ term in (49) and $\sigma_X\,\mathrm{d}\mathbf{Z}$ in (14) becomes:

$$\Sigma_{\text{OU}}\,\mathrm{d}\mathbf{W} = \sigma_X\,\mathrm{d}\mathbf{Z} = \sigma_X A\,\mathrm{d}\mathbf{W}$$

which is equivalent to:

$$\Sigma_{\text{OU}} = \sigma_X A. \tag{53}$$

To prove (50)–(52), we combine (21) and (53):

$$\begin{aligned}
\Sigma_W &= \sigma_W \sigma_\rho \sigma_W^T \\
&= \sigma_W A A^T \sigma_W^T \\
&= \begin{pmatrix} \frac{1}{X_1} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{X_d} \end{pmatrix} \sigma_X A A^T \sigma_X^T \begin{pmatrix} \frac{1}{X_1} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{X_d} \end{pmatrix} \\
&= \begin{pmatrix} \frac{1}{X_1} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{X_d} \end{pmatrix} \Sigma_{\text{OU}}\Sigma_{\text{OU}}^T \begin{pmatrix} \frac{1}{X_1} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{X_d} \end{pmatrix}
\end{aligned}$$

$$\begin{aligned}
\Sigma_{WX} &= \sigma_W \sigma_\rho \sigma_X^T \\
&= \begin{pmatrix} \frac{1}{X_1} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{X_d} \end{pmatrix} \sigma_X A A^T \sigma_X \\
&= \begin{pmatrix} \frac{1}{X_1} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{X_d} \end{pmatrix} \Sigma_{\text{OU}}\Sigma_{\text{OU}}^T
\end{aligned}$$

$$\begin{aligned}
\Sigma_X &= \sigma_X \sigma_\rho \sigma_X^T \\
&= \sigma_X A A^T \sigma_X^T \\
&= \Sigma_{\text{OU}}\Sigma_{\text{OU}}^T \blacksquare
\end{aligned}$$

Next, letting $\{N_i\}_{i=1}^d$ be d independent standard normal variables and $\{\sigma_{ij}\}$ be the entries of Σ_{OU}, by solving the SDE (49) together with (17) and (18) the dynamics of the system are:

$$\Delta_{t_{k-1}}\mathbf{X}^{(i)} = \mathbf{X}_{t_k}^{(i)} - \mathbf{X}_{t_{k-1}}^{(i)} = (\mu_i - \mathbf{X}_{t_{k-1}})(1 - e^{-\lambda_i h}) + \sum_{j=1}^{d} \sigma_{ij}\sqrt{\frac{1 - e^{-2\lambda_i h}}{2\lambda_i}} N_j \tag{54}$$

$$\Delta_{t_{k-1}} Y = Y_{t_k} - Y_{t_{k-1}} = e^{rt_{k-1}}(e^{rh} - 1) \tag{55}$$

$$\Delta_{t_{k-1}} W = W_{t_k} - W_{t_{k-1}} = W_{t_{k-1}}\pi_{t_{k-1}+}^T \frac{\Delta_{t_{k-1}}\mathbf{X}}{\mathbf{X}_{t_{k-1}}} + \frac{(1 - \pi_{t_{k-1}+}^T \mathbf{1})W_{t_{k-1}}}{Y_{t_{k-1}}}\Delta_{t_{k-1}} Y - c_{t_{k-1}}, \tag{56}$$

where the transaction cost is

$$c_{t_k} := \sum_{i=1}^{d} \alpha W_{t_k} |\pi_{t_k+}^{(i)} - \pi_{t_k-}^{(i)}|.$$

We employ the same method as in Section 3.3 to initialize and generate n sample paths for training the deep neural network, the only differences are that the sizes of X_{t_k}, W_{t_k} and $\pi_{t_k\pm}$ should be $n \times d$ rather than $n \times 1$, and the environment dynamics are given by (54)–(56) rather than (42)–(44). The empirical loss is the same as that defined by (45):

$$\text{loss} \triangleq \frac{1}{n}\sum_{i=1}^{n} \left(J(w_0, x_0, T) - U(W_T^{(i)})\right). \tag{57}$$

5. Numerical experiments based on model simulation

In this section, we describe the feedforward neural network architecture and training details for both single and multiple asset portfolio optimization. Using parameters estimated from the VIX front month future, the simulation results illustrate how transaction costs affect the NT zone.

5.1. Neural network structure and training procedure

For each time step t_k, we build two neural networks for representing functions $f_{\theta_{t_k}^u}(x), f_{\theta_{t_k}^d}(x)$ and the lower/upper boundaries as given by (38)–(39) or (46)–(47). Each neural network is a feedforward neural network with three hidden layers, where the number of neurons used in the hidden layers are [20,40,80]. Batch Normalization speeds up the training (Ioffe and Szegedy 2015). We use the leaky ReLU (Xu *et al.* 2015) activation function on the layers. A typical ReLU is a piecewise linear activation function which takes zero on the negative part, and retains the positive part. To allow a small positive gradient even when the unit is not activated, we take the leaky ReLU where

$$f(x) = \begin{cases} x, & \text{for } x \geq 0 \\ \alpha x & \text{for } x < 0. \end{cases}$$

with $\alpha \in (0, 1)$ being the leaky coefficient. In our experiment, we take the leaky coefficient to be $\alpha = 0.2$ in each hidden layer, and on the output layer we take $\alpha = 0.05$. The reason we are not using identity function in the last layer is because intuitively lower boundary should be below π^* and upper boundary should be above π^*, since the NT zone is an area around and including the optimal position π^* derived under the no-transaction-costs assumption. In other words, the output $f_{\theta_{t_k}^u}(x)$ and $f_{\theta_{t_k}^d}(x)$ must be non-negative. One way to ensure positive output is to employ ReLU on the last layer. However, doing so will cause the gradient to be zero for all negative outputs and therefore prevent gradient backpropagation. Hence, we choose leaky ReLU with a small coefficient $\alpha = 0.05$ in the last layer to allow effective gradient backpropagation. The Adam optimizer (Kingma and Ba 2014) is implemented with a decaying learning rate that starts at 1 and decays by $\frac{1}{2}$ every 500 steps. If the learning rate is too high, the training can end up unstable, while learning rate being too low may cause the training to fail. Thus we carefully choose the decaying learning rate to avoid such undesirable situations. The total number of training steps is 3000, and for each training step, we generate 100,000 sample paths. The neural network is implemented and trained using Tensorflow (Abadi *et al.* 2015), a widely available deep learning tool developed by Google.

At each training step, we generate a batch of new training data using (42)–(44) or (54)–(56). More specifically, we generate a batch of independent normal random variables and feed them into equation (42) or (54). In this experiment, we observe that after 3000 training steps, the outputs already converge well, and therefore choose to use 3000 training steps. One could also employ an 'early stopping' rule and terminate the algorithm within 3000 training steps, by using a validation data set. The validation set could be a data set generated before the training starts, or it could be generated online independent to the training set. If the loss on validation set is stable for several steps or begins to increase, then the training terminates. After training, one could generate a new batch of sample paths as the test set. In this paper, the test data equals 10 000 sample paths.

5.2. Relations between the no-trade zone and transaction cost

For these empirical tests, the parameters are estimated from the data of the VIX front month future in the year 2013 following maximum likelihood estimation (Iacus 2009):

$$\begin{array}{ccc} \mu = 15.4463 & \lambda = 0.113 \times 252 & \sigma = 0.606 \times \sqrt{252} \\ T = 1 & N = 50 & r = 0.05 \\ x_0 = 13.950 & w_0 = 1000 & \gamma = 2. \end{array}$$

As shown in Figure 5, the NT zone at time $t_k = 0.4$ illustrates how transaction cost affects the no trade zone.

In Figure 5, 'lb' denotes the lower boundary and 'ub' denotes the upper boundary of the NT zone, and the percentage number denotes the transaction cost rate α in (41) and varies from 0.7% to 2.8%. As expected, the NT zone gets wider as the transaction cost increases.

Another way to characterize the NT zone is to consider the 4 points where the lower/upper boundary hits the $+1/-1$ limit. These are the four vertices of NT zone. By plotting the price level at which lower/upper boundary hits the limit,

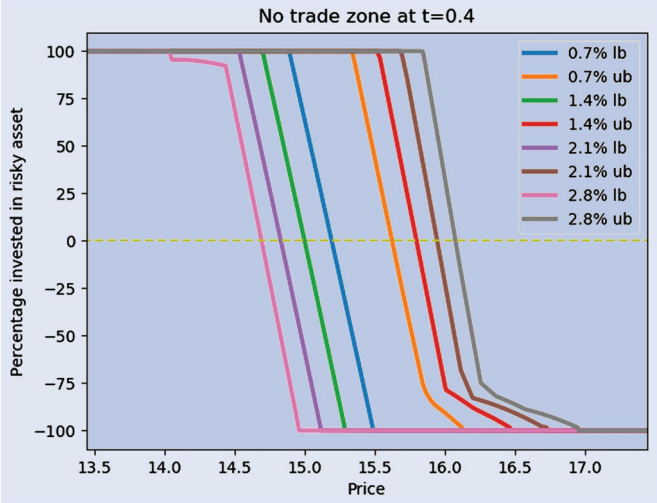

Figure 5. No trade zone at $t = 0.4$ under different transaction cost. Larger transaction costs gives wider NT zone.

we gain insight into the NT zone evolution over time with different transaction costs (Figure 6).

We make several observations:

- In each subplot, as transaction cost increases, the divergence between the curve and the 0-transaction-cost curve becomes larger. This means that the NT zone gets wider at every time step as the transaction cost increases.
- The price at which the lower boundary reaches $+1$ decreases as time approaches the end of the horizon, while the price at which upper boundary hits -1 increases. This phenomenon gets more obvious as transaction cost becomes higher. This suggests that the investor should be more cautious and less aggressive as the horizon approaches.
- When the transaction cost increases, the computed curves become more jagged. This phenomenon could be overcome by expanding the batch size or increase the training steps. These small spikes happen at prices that deviate more from the mean-reverting level. Because these prices are rarely seen in training hence the values for the NT zone boundaries at these prices are less accurate than the values for the NT zone boundaries around the mean-reverting level.

5.3. *Performance based on simulation*

We assume there is 2.0% transaction cost. The utility is scaled by subtracting 0.999 and multiplying by 10^6 to speed up training. The performance of the policy given by the neural network and the optimal policy we derived under zero transaction cost is shown in Figure 7. The result is averaged over 10 000 randomly generated sample paths. The policy learned by neural network not surprisingly significantly outperforms the policy derived under zero transaction cost.

6. Numerical experiments with market data

In this section, we provide numerical experiments with real data. We first describe the method of constructing

Figure 6. Price level at which boundary hits the leverage ratio limit. (a) Upper left: Price level at which lower boundary hits $+1$ limit. (b) Upper right: Price level at which upper boundary hits -1. (c) Lower left: Price level at which lower boundary hits -1. (d) Lower right: Price level at which upper boundary hits $+1$.

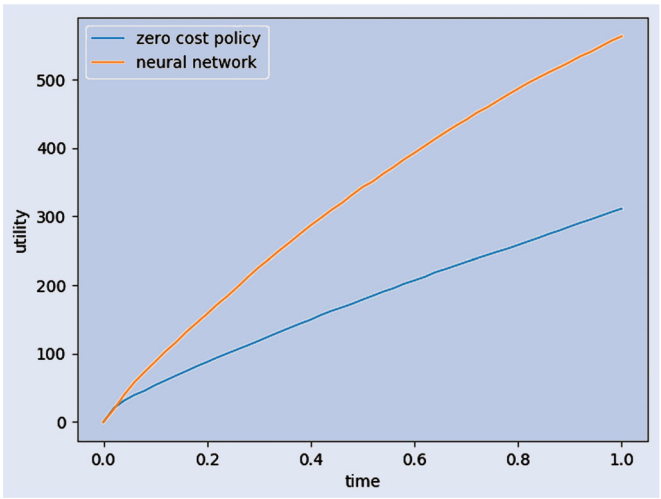

Figure 7. Mean utility over 10,000 sample paths. The blue curve is mean utility over time following the optimal strategy we derived under 0 transaction cost. The orange curve is the mean utility overtime following the NT zone policy.

mean-reverting tradables. Although the discovery of pairs is not the focus of this paper, the construction process is important for us to estimate the transaction cost. Followed by that, we present the backtest results for a single and multiple risky asset portfolio for year 2014–2017. In particular, we are able to solve the multi-asset portfolio choice problem with 48 OU assets, 1 risk-free asset and 50 time steps in 3 h computer time, which is considerable faster than traditional numerical methods. Finally, we summarize the computational cost of portfolios containing different numbers of assets and conjecture the approximate complexity of the DNN method.

6.1. *Constructing mean-reverting assets*

In practice, we slightly modified the method employed by Avellaneda and Lee (2010) to create mean-reverting assets by longing and shorting correlated tradable assets on the market. We regress the stock price on the ETF price, and then long and short according to the beta so that the resulting assets are market-neutral. Avellaneda and Lee (2010) suggest that in many cases, the drift term in the regression is negligible compared to the fluctuation of the residual, and that the residual would be mean-reverting if it is a stationary process. By carefully choosing the assets we use in this experiment, we model the residual as a mean-reverting process, and estimate their parameters using maximum likelihood estimators. The stock and ETF daily prices are obtained from Thomson Reuters Datastream. Denoting the stock price by S_t, the relationship between stock price and ETF follows:

$$S_t = \beta \times ETF_t + \epsilon_t$$

Calculating the estimate $\hat{\beta}$ of β using 60 days of data on S_t and ETF_t, we found that the residual ϵ_t is mean reverting around 0 for a variety of stocks. Then one share of mean-reverting asset X is created by longing one share of stock and short $\hat{\beta}$ share of ETF. To avoid negative values of X, we pretend its value is actually $S_t - \hat{\beta} \times ETF_t + C$ where C is the constant amount of extra cash we have to put into one share of X, i.e. $X_t = S_t - \hat{\beta} \times ETF_t + C$. As long as C is large enough, X_t will never go below 0. We set $C = 20$ in our experiment. Assume the transaction cost rate for 1 share of both ETF and the stock is 40 bps.

In Section 5.2, we use stock JPM and XLF ETF to construct the mean-reverting asset. There are two sources of transaction cost:

- The carry cost of risky asset X. This is the transaction cost associated with changing β. Each day we recalibrate $\hat{\beta}$ using the past 60 days' data, and the recalibrated $\hat{\beta}$ changes from $\hat{\beta_{t-}}$ to $\hat{\beta_{t+}}$ hence for each share of X that we are holding, we need to pay $|\hat{\beta_{t-}} - \hat{\beta_{t+}}| \times ETF_t \times 0.004$ amount of transaction fee. Usually the daily change $|\hat{\beta_{t-}} - \hat{\beta_{t+}}|$ is around 0.02, and the ETF price is around \$20. So the daily transaction cost for maintaining 1 share of X is around $20 \times 0.02 \times 0.0004 = 0.0016$. Most of the time our wealth takes values in $[1000, 2000]$, so the number of shares of X we are longing or shorting is between $[50, 75]$, the total carry cost of X should not exceed $\$0.0016 \times 75 = \$0.12$ per day.
- The transaction cost to long or short one share of X. To long or short one share of X, we need to long (respectively, short) 1 share of stock and short (respectively, long) $\hat{\beta}$ share of ETF, the total transaction cost of this operation will be $S_t \times 0.004 + |\hat{\beta}| \times ETF_t \times 0.004$. Recently $S_t \approx 100$ and $ETF_t \approx 20$, hence the transaction cost of creating 1 share of X is around $(100 + 20 * 3) \times 0.004 \approx 0.064$, given the price $X_t \approx 20$, this amounts to $\frac{0.064}{20} \approx 3.2\%$ to 1 share of X.

In our experiments, we take into account the two kinds of transaction cost discussed above. In particular, we assume the daily carry cost of each share of X is \$0.00016 and the transaction cost for each share of X is 3.2% for all mean-reverting assets.

6.2. *Experiment with one asset*

As an illustrative example, we combine the JPM stock and the XLF ETF to create the mean-reverting asset, and backtest the approach with data for the year 2014–2017. At the start of each year, the parameters are calibrated using historical data from 2009 to the previous year. The investment horizon is one year, using the daily closing price and the time step to be one trading day. The starting wealth of each year is 1000. The risk-free rate is $r = 0.05$ and the risk aversion coefficient is $\gamma = 2$. Parameters μ, λ and σ for the OU process are estimated using maximum likelihood estimation (MLE) (Iacus 2009). For each year, the parameters are estimated using all data points prior to the year. For the year 2014, the maximum likelihood estimations are $\hat{\mu} = 20.0151$, $\hat{\lambda} = 9.2031$ and $\hat{\sigma} = 6.3969$. For 2015, $\hat{\mu} = 19.9367$, $\hat{\lambda} = 9.2995$ and $\hat{\sigma} = 6.2873$. For 2016, $\hat{\mu} = 20.0574$, $\hat{\lambda} = 8.5021$ and $\hat{\sigma} = 6.1741$. For 2017, $\hat{\mu} = 20.1488$, $\hat{\lambda} = 8.4734$ and $\hat{\sigma} = 6.1729$.

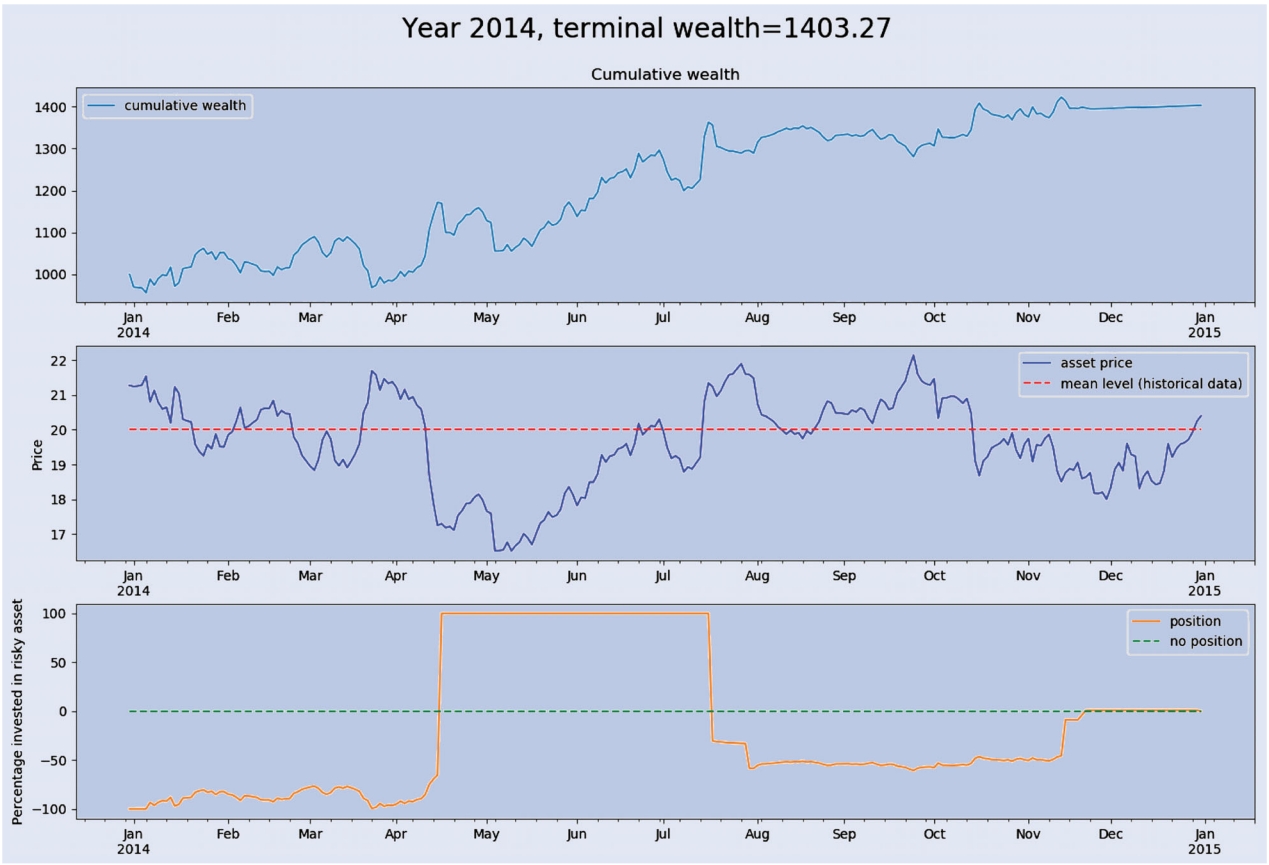

Figure 8. Backtest result for year 2014. The mean-reverting pair is created using listed stock JP Morgan Chase(JPM) and XLF ETF. (a) Cumulative wealth. (b) Mean-reverting level estimated using historical data and asset price. (c) Percentage invested in risky asset.

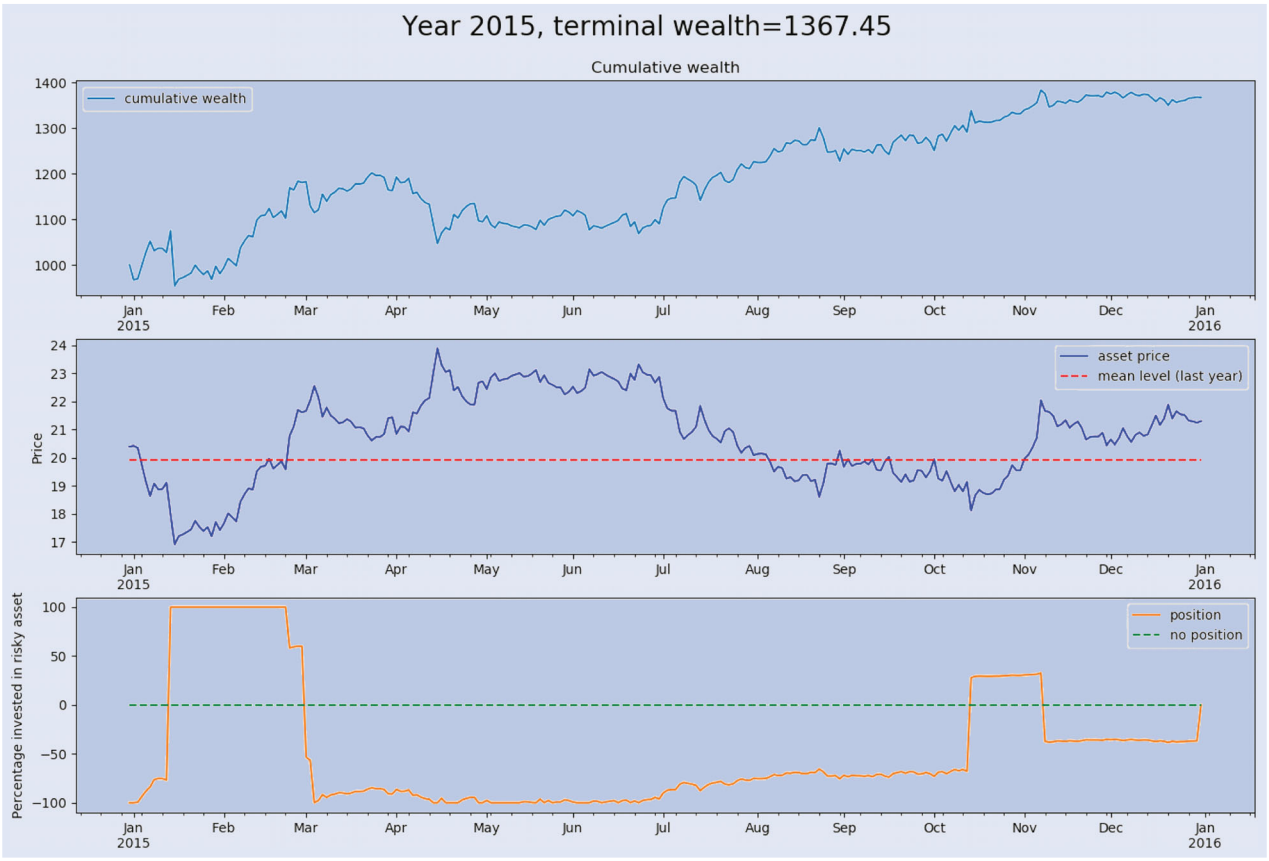

Figure 9. Backtest result for year 2015. The mean-reverting pair is created using listed stock JP Morgan Chase(JPM) and XLF ETF. (a) Cumulative wealth, (b) mean-reverting level estimated using historical data and asset price, (c) percentage invested in risky asset.

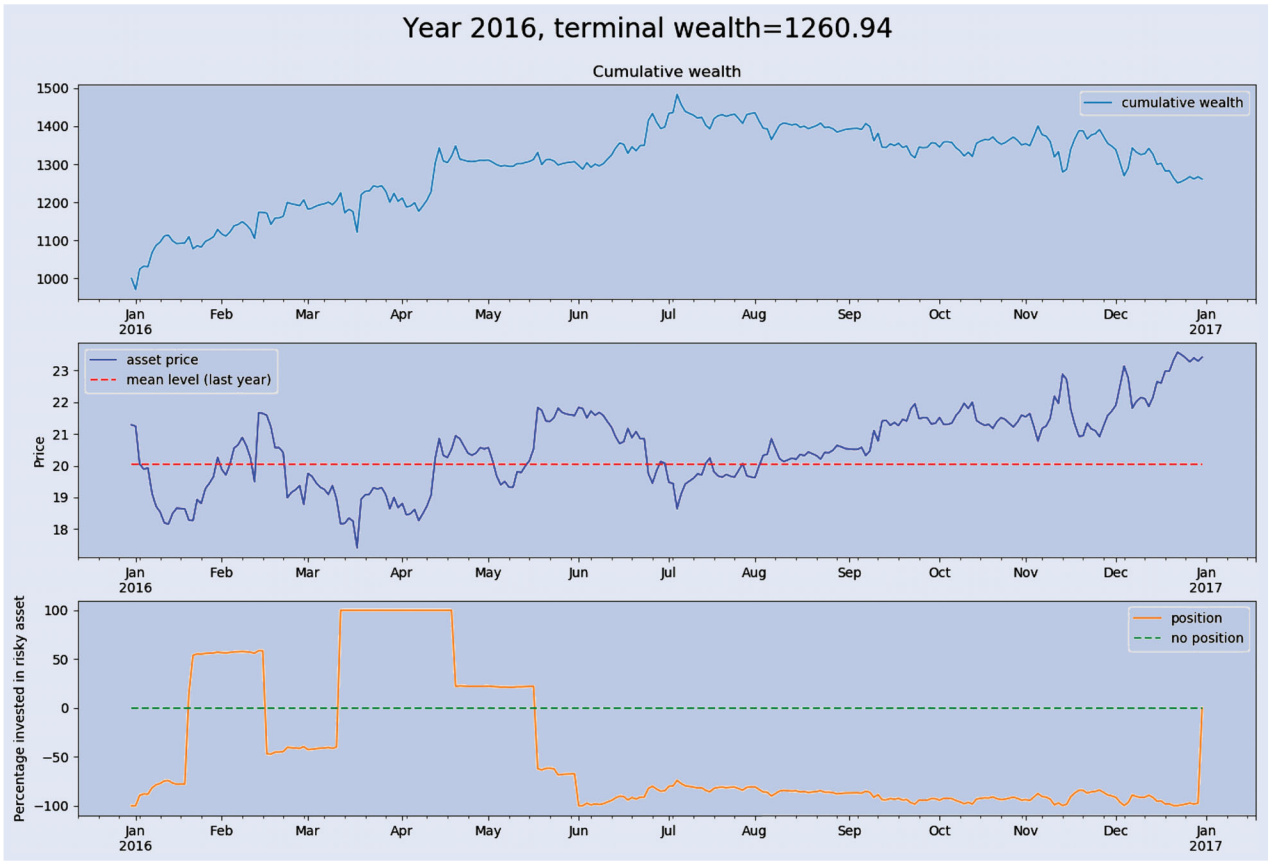

Figure 10. Backtest result for year 2016. The mean-reverting pair is created using listed stock JP Morgan Chase (JPM) and XLF ETF. (a) Cumulative wealth, (b) mean-reverting level estimated using historical data and asset price, (c) percentage invested in risky asset.

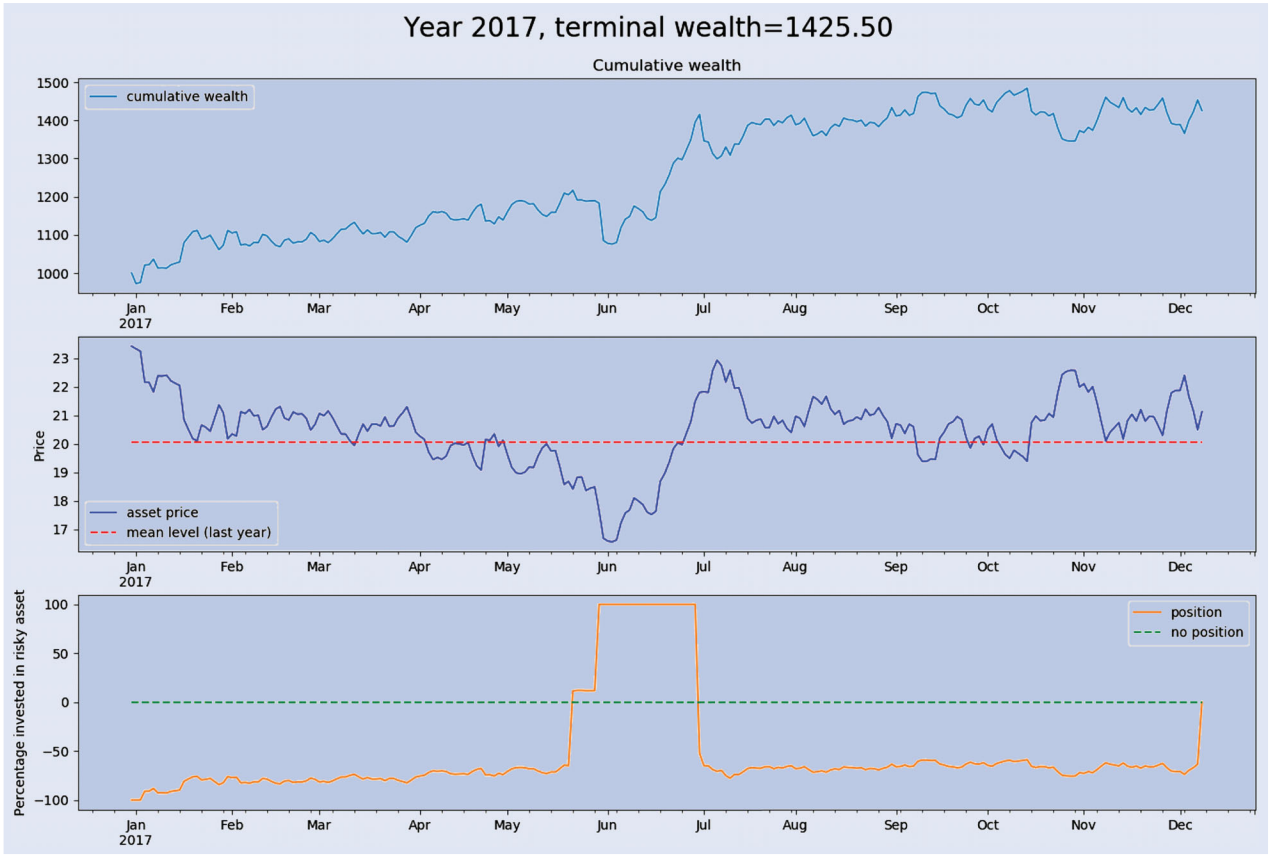

Figure 11. Backtest result for year 2017. The mean-reverting pair is created using listed stock JP Morgan Chase (JPM) and XLF ETF. (a) Cumulative wealth, (b) mean-reverting level estimated using historical data and asset price, (c) percentage invested in risky asset.

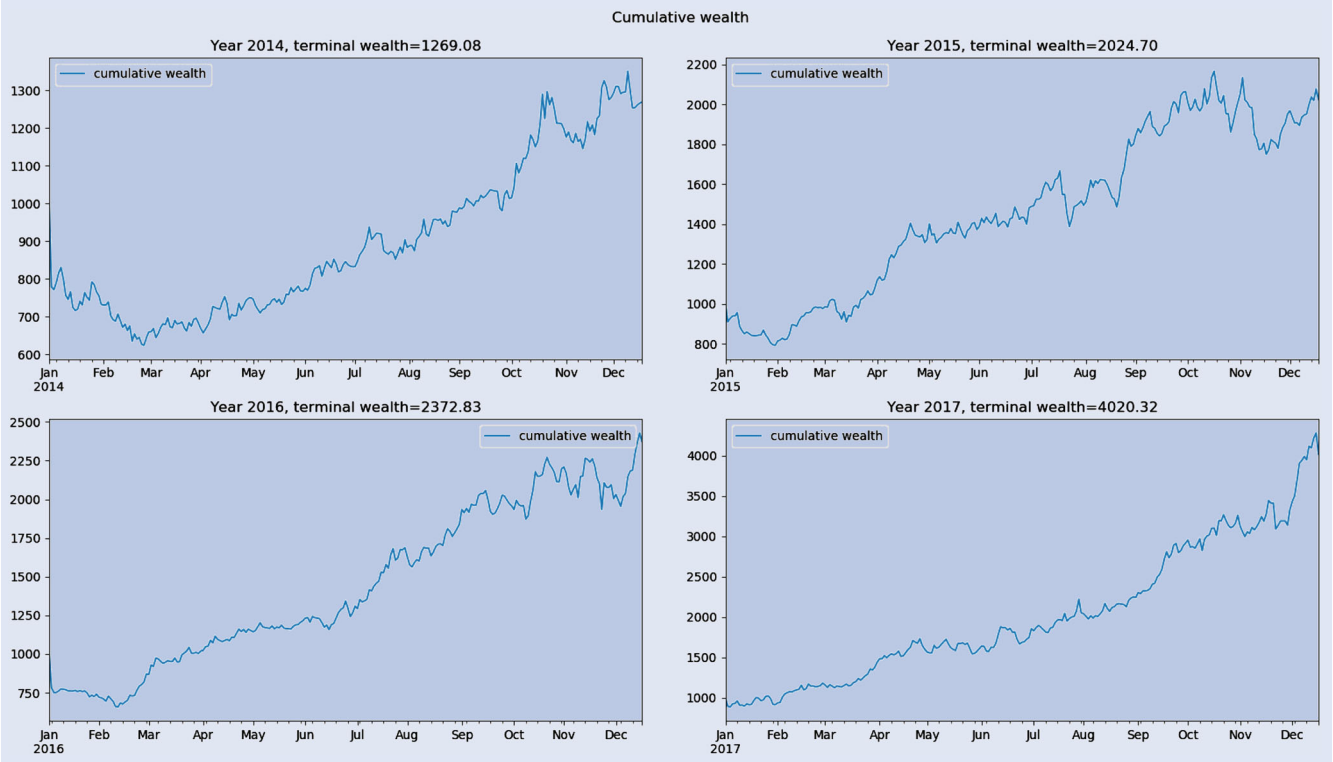

Figure 12. Cumulative wealth when backtest using 48 mean-reverting assets from 2014 to 2017.

The out of sample results for the year of 2014 are plotted in Figure 8. By comparing (b) and (c) in Figure 8, we have following observations:

- At the beginning we short the asset because its price is more than \$1 above its mean level.
- Starting in April we shift our position from fully shorting the asset to fully holding the asset. This change is due to asset price dropping from about \$1 above mean level to \$3 below mean level.
- In July, we change our position from holding the asset to shorting roughly about 50% of our wealth in the asset. This is because asset price goes above the mean-reverting level in July, however the price is not high enough for us to fully short the asset. From July onwards, the asset price oscillates around the mean-reverting level hence we only rebalance the position slightly.

The backtest results for 2015–2017 have similar behaviors (Figures 9–11):

In the backtests, despite the carry cost and high transaction cost, the strategy steadily generates positive returns over the last 4 years. Rebalancings are mostly taken at a 'reasonable time', i.e. fully short around the local high price and fully long at the local low price. Rebalancings are sparse due to the NT zone, which is derived from the deep neural network using the ODE system without transaction costs as a starting point.

6.3. Experiment with multi-asset portfolios

We form a portfolio of 48 stocks together with their corresponding ETFs (16 in total) to construct the mean-reverting assets as in the previous section. The information for stock-ETF pairs is summarized in Appendix 3. Assuming a 3.2% transaction cost for each asset, and the same parameters of $r = 0.05$ and $\gamma = 2$ as we did in one-asset case, we employ a fully connected feed-forward neural network with three hidden layers at each time step for both the lower and the upper bound, with the number of neurons $\{64, 128, 256\}$, respectively. Note that the number of parameters for simulating the computations in each neural network is about eight times more than for single asset case. Instead of building one neural network for each trading day, we build one neural network for every five trading days, with the same NT zone values for each day between weekly rebalancing dates. The initial wealth is $W_0 = 1000$. The output of the neural network is a 16-dimensional vector. Moreover, leverage constraints are added so that $\|\pi\|_\infty \leq 0.5$, i.e. the long or short position in each asset should not exceed 50% of the total wealth, so the maximum leverage should not exceed eight times of the current wealth. In practice, the total leverage is much less than 800% because the short and long positions in different assets offset each other. We estimate Λ and Σ_{OU} in the OU process (49) according to the method in Wan (2010). Similar to the single asset case, parameters for each year are estimated from the historical data of the previous year except that the parameters used for year 2014 backtest are estimated from historical data of 2010–2013. The estimated parameters are summarized in the online Appendix. From Figure 12, we see that the strategy learned by deep neural network steadily generates profits. In the years 2014 and 2016, the initial position of holding no risky assets might be far from the NT zone given the asset prices at that time. Since that the total allowed leverage, 800%, is high, and that the transaction cost is 3.2%, the investor may have an immediate loss due to the first

Table 1. Number of parameters and computation time used for different numbers of assets.

# of Assets	Layer1	Layer 2	Layer 3	Time Used (HH:MM:SS)
16	64	128	256	3:24:35
32	128	256	512	5:37:05
48	128	256	512	6:34:44

rebalancing. Yet the loss is quickly recovered and terminal wealth shows that the algorithm is promising.

6.4. *Computation time*

We carry out backtest on different number of assets and record the total computation time needed. The experiment is implemented with TensorFlow-GPU and run on the Princeton University's GPU Server with one NVDIA P100 GPU node.

The dimension of the input and output layers of each Neural Network is the same as number of assets. In the above table, *Layer i* means the *i*th hidden layer in each subnetwork. Table 1 shows that the computation time scales polynomially (if not linearly) with the dimension of the problem and our approach can be applied to high-dimensional problems.

7. Conclusion

In this paper, we present a combined method for solving the multi-period portfolio optimization problem where the underlying assets follow Ornstein–Uhlenbeck processes and transaction costs are not negligible. We first solve the ODE solutions for the no transaction cost case, and then approximate the boundaries of the no-trade zones by via feedforward neural network.

Backtesting with real data for years 2014–2017 shows that the strategy performs well in both the one-asset and multi-asset cases, yielding excellent annual return in the single-asset case and doubling the initial wealth each year in the multi-asset case. Importantly, our method enjoys superb run-time efficiency; it does not suffer from the curse of dimensionality affecting other conventional numerical methods. Hence the trading strategy can be extended to portfolios with a large number of assets. This opens a door for future research opportunities as one can further test out other promising neural network architectures or combine them with reinforcement learning and other concepts to explore better dynamic trading strategies.

To this end, there are significant applications that could be amenable to a similar strategy: combining a fast start algorithm with a deep neural network. An example is a multi-period Markov switching model with transaction costs. Here, the form of the optimal policy is complicated by the curse of dimensionality, the Markov transition structure, and transaction costs. Chauvet and Potter (2013) compares nine popular models and find that it is particularly difficult to forecast output growth during recessions, and they show that the Markov models are the only quantitative method to forecast either the 2001/02 or the 2008/09 crashes. Nystrup *et al.* (2018) tackles a portfolio optimization problem with hidden Markov regime switching, and presents a strategy for the case of one risky asset with solving a truncated problem at each time step. Online versions of the combined strategy have direct usage by high frequency traders, wherein the large amount of data allows for effective and semi-real-time calibration of the underlying stochastic processes. These multi-period financial models are able to include realistic considerations and have widespread applications, but suffer from severe computational limits. Overcoming these barriers will be a central challenge for data scientists going forward.

Acknowledgements

The paper has benefited by the referees' comments and the work of the Princeton doctoral student Xiaoyue Li.

Disclosure statement

No potential conflict of interest was reported by the authors.

Supplemental data

Supplemental data for this article can be accessed at http://dx.doi.org/10.1080/14697688.2020.1729994.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M. and Ghemawat, S., TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Available online at: www.tensorflow.org/about/bib (accessed 29 July 2019).

Avellaneda, M. and Lee, J.H., Statistical arbitrage in the US equities market. *Quant. Finance*, 2010, **10**(7), 761–782.

Beck, C., Weinan, W. and Jentzen, A., Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations. *J. Nonlinear Sci.*, 2019, **29**(4), 1563–1619.

Benth, F.E. and Karlsen, K.H., A note on Merton's portfolio selection problem for the Schwartz mean-reversion model. *Stoch. Anal. Appl.*, 2005, **23**(4), 687–704.

Bertocchi, M., Consigli, G. and Dempster, M.A. (Eds.), *Stochastic Optimization Methods in Finance and Energy: New Financial Products and Energy Market Strategies*, Vol. 2011 (Springer: New York).

Bichuch, M. and Sircar, R., Optimal investment with transaction costs and stochastic volatility, 2014. Available online at: arxiv.org/abs/1401.0562 (accessed 29 July 2019).

Bichuch, M. and Sircar, R., Optimal investment with transaction costs and stochastic volatility Part II: Finite Horizon, 2018. Available online at: https://ssrn.com/abstract = 2659918 (accessed 29 July 2019).

Broadie, M. and Shen, W., Numerical solutions to dynamic portfolio problems with upper bounds. *Comput. Manag. Sci.*, 2017, **14**(2), 215–227.

Brown, D.B. and Smith, J.E., Dynamic portfolio optimization with transaction costs: Heuristics and dual bounds. *Manag. Sci.*, 2011, **57**(10), 1752–1770.

Brown, D.B., Smith, J.E. and Sun, P., Information relaxations and duality in stochastic dynamic programs. *Oper. Res.*, 2010, **58**(4-part-1), 785–801.

Campbell, J.Y. and Viceira, L.M., Consumption and portfolio decisions when expected returns are time varying. *Quart. J. Econ.*, 1999, **114**(2), 433–495.

Chan, P. and Sircar, R., Optimal trading with predictable return and stochastic volatility, 2015. Available online at: https://ssrn.com/abstract = 2623747 (accessed 29 July 2019).

Chauvet, M. and Potter, S., Forecasting output. *Handbook of Economic Forecasting*, 2013, **2**, 141–194. (Elsevier: North Holland).

Culkin, R. and Das, S.R., Machine learning in finance: The case of deep learning for option pricing. *J. Invest. Manag.*, 2017, **15**(4), 92–100.

Dai, M. and Zhong, Y., Penalty methods for continuous-time portfolio selection with proportional transaction costs. *J. Comput. Finance*, 2010, **13**(3), 1.

Davis, M.H. and Norman, A.R., Portfolio selection with transaction costs. *Math. Oper. Res.*, 1990, **15**(4), 676–713.

Davis, M.H.A., Panas, V.G. and Zariphopoulou, T., European option pricing with transaction costs. *SIAM J. Control Opt.*, 1993, **31**(2), 470–493.

Dempster, M.A., Mitra, G. and Pflug, G. (Eds.), Financial planning in a dynamic setting. *Quant. Finance*, 2007, **7**(2), 111–112.

Gârleanu, N. and Pedersen, L.H., Dynamic trading with predictable returns and transaction costs. *J. Finance*, 2013, **68**(6), 2309–2340.

Goodfellow, I., Bengio, Y. and Courville, A., *Deep Learning*, 2016 (MIT Press: Cambridge, MA).

Grinold, R.C., Model building techniques for the correction of end effects in multistage convex programs. *Oper. Res.*, 1983, **31**(3), 407–431.

Grinold, R., A dynamic model of portfolio management. *J. Invest. Manag.*, 2006, **4**(2), 5–22.

Hakansson, N.H., Optimal investment and consumption strategies under risk, an uncertain lifetime, and insurance. *Int. Econ. Rev.*, 1969, **10**(3), 443–466.

Han, J. and Weinan, E., Deep learning approximation for stochastic control problems, 2016. Available online at: https://arxiv.org/abs/1611.07422 (accessed 29 July 2019).

Han, J., Jentzen, A. and Weinan, E., Solving high-dimensional partial differential equations using deep learning. *Proc. Natl. Acad. Sci.*, 2018, **115**(34), 8505–8510.

Hornik, K., Stinchcombe, M. and White, H., Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Net.*, 1990, **3**(5), 551–560.

Iacus, S.M., *Simulation and Inference for Stochastic Differential Equations: With R Examples*, 2009 (Springer: New York).

Ioffe, S. and Szegedy, C., Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. Available online at: https://arxiv.org/abs/1502.03167 (accessed 29 July 2019).

Jurek, J.W. and Yang, H., Dynamic portfolio selection in arbitrage, 2007. Available online at: https://ssrn.com/abstract = 882536 (accessed 29 July 2019).

Kim, T.S. and Omberg, E., Dynamic nonmyopic portfolio behavior. *Rev. Financ. Stud.*, 1996, **9**(1), 141–161.

Kingma, D.P. and Ba, J.L., Adam: A method for stochastic optimization, 2014. Available online at: https://arxiv.org/abs/1412.6980 (accessed 29 July 2019).

Konicz, A.K. and Mulvey, J.M., Applying a stochastic financial planning system to an individual: Immediate or deferred life annuities? *J. Retire.*, 2013, **1**(2), 46–60.

LeCun, Y., Bengio, Y. and Hinton, G., Deep learning. *Nature*, 2015, **521**(7553), 436–444.

Leung, T. and Li, X., *Optimal Mean Reversion Trading: Mathematical Analysis and Practical Applications*, 2016 (World Scientific: Singapore).

Liu, H., Optimal consumption and investment with transaction costs and multiple risky assets. *J. Finance*, 2004, **59**(1), 289–338.

Liu, J., Portfolio selection in stochastic environments. *Rev. Financ. Stud.*, 2006, **20**(1), 1–39.

Liu, H. and Loewenstein, M., Optimal portfolio selection with transaction costs and finite horizons. *Rev. Financ. Studies*, 2002, **15**(3), 805–835.

Liu, J. and Timmermann, A., Optimal convergence trade strategies. *Rev. Financ. Studies*, 2013, **26**(4), 1048–1086.

Lynch, A.W. and Tan, S., Multiple risky assets, transaction costs, and return predictability: Allocation rules and implications for us investors. *J. Financ. Quant. Anal.*, 2010, **45**(4), 1015–1053.

Ma, G. and Zhu, S.P., Optimal investment and consumption under a continuous-time cointegration model with exponential utility. *Quant. Finance*, 2019, **19**(7), 1135–1149.

Magill, M.J.P and Constantinides, G.M., Portfolio selection with transactions costs. *J. Econ. Theory*, 1976, **13**(2), 245–263.

Merton, R.C., Lifetime portfolio selection under uncertainty: The continuous-time case. *Rev. Econ. Stat.*, 1969, **51**, 247–257.

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

Merton, R.C., An intertemporal capital asset pricing model. *Econometrica*, 1973, **41**(5), 867–887.

Mossin, J., Optimal multiperiod portfolio policies. *J. Bus.*, 1968, **41**(2), 215.

Mulvey, J.M. and Simsek, K.D., Rebalancing Strategies for long-term investors. *Computational Methods in Decision-Making, Economics and Finance*, pp. 13–32, 2002 (Springer: Boston, MA).

Mulvey, J.M., Pauling, W.R. and Madey, R.E., Advantages of multiperiod portfolio models. *J. Portf. Manage.*, 2003, **29**(2), 35–45.

Munk, C., Sørensen, C. and Vinther, T.N., Dynamic asset allocation under mean-reverting returns, stochastic interest rates, and inflation uncertainty. *Int. Rev. Econ. Finance*, 2004, **13**(2), 141–166.

Muthuraman, K. and Kumar, S., Multidimensional portfolio optimization with proportional transaction costs. *Math. Fin.: An Int. J. Math., Stats. Fin. Econ.*, 2006, **16**(2), 301–335.

Muthuraman, K. and Zha, H., Simulation-based portfolio optimization for large portfolios with transaction costs. *Math. Fin.: An Int. J. Math., Stats. Fin. Econ.*, 2008, **18**(1), 115–134.

Nystrup, P., Madsen, H. and Lindström, E., Dynamic portfolio optimization across hidden market regimes. *Quant. Finance*, 2018, **18**(1), 83–95.

Pole, A., *Statistical Arbitrage: Algorithmic Trading Insights and Techniques*, 2007 (Wiley: Chichester).

Ritter, G., Machine learning for trading, 2017. Available online at: https://ssrn.com/abstract = 3015609 (accessed 29 July 2019).

Samuelson, P.A., Lifetime portfolio selection by dynamic stochastic programming. *Rev. Econ. Stat.*, 1969, **51**(3), 239–246.

Shreve, S.E. and Soner, H.M., Optimal investment and consumption with transaction costs. *Ann. Appl. Probab.*, 1994, **4**(3), 609–692.

Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M. and Dieleman, S., Mastering the game of Go with deep neural networks and tree search. *Nature*, 2016, **529**(7587), 484–489.

Sirignano, J. and Spiliopoulos, K., DGM: A deep learning algorithm for solving partial differential equations, 2017. Available online at: arXiv:1708.07469.

Taksar, M., Klass, M.J. and Assaf, D., A diffusion model for optimal portfolio selection in the presence of brokerage fees. *Math. Oper. Res.*, 1988, **13**(2), 277–294.

Tsay, R.S., *Analysis of Financial Time Series*. Wiley Series in Probability and Statistics, 2010 (John Wiley & Sons: Hoboken, NJ).

Tourin, A. and Yan, R., Dynamic pairs trading using the stochastic control approach. *J. Econ. Dyn. Control*, 2013, **37**(10), 1972–1981.

Wachter, J.A., Portfolio and consumption decisions under mean-reverting returns: An exact solution for complete markets. *J. Financ. Quant. Anal.*, 2002, **37**(1), 63–91.

Wan, Z., Modeling investment returns with a multivariate Ornstein-Uhlenbeck process. Doctoral Dissertation, Department of Statistics and Actuarial Science-Simon Fraser University, 2010.

Weinan, E., Han, J. and Jentzen, A., Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Commun. Math. Stat.*, 2017, **5**(4), 349–380.

Xu, B., Wang, N., Chen, T. and Li, M., Empirical evaluation of rectified activations in convolutional network, 2015. Available online at: https://arxiv.org/abs/1505.00853 (accessed 29 July 2019).

Zenios, S.A. and Ziemba, W.T. (Eds.), *Handbook of Asset and Liability Management: Applications andCaseStudies*, 2007 (Elsevier: North Holland).

Ziemba, W.T. and Mulvey, J.M. (Eds.), *Worldwide Asset and Liability Modeling*, 1998 (Cambridge University Press: Cambridge, UK).

Appendix 1. Solution to single-asset HJB equation

As mentioned in Section 2, we assume indirect utility J takes the following ansatz:

$$J = \frac{(W\phi(\tau,X))^{1-\gamma}-1}{1-\gamma}$$
$$\phi(\tau,\mathbf{X}) = \exp\{A(\tau)+B(\tau)^T X+\tfrac{1}{2}C(\tau)X^2\}. \tag{A1}$$

We first derive the partial derivatives:

$$\begin{aligned}
J_W &= \phi(\tau,X)(W\phi(\tau,X))^{-\gamma}\\
J_{WW} &= -\gamma\phi(\tau,X)^2(W\phi(\tau,X))^{-\gamma-1}\\
J_X &= (W\phi(\tau,X))^{1-\gamma}[B(\tau)+C(\tau)X]\\
J_{XX} &= (W\phi(\tau,X))^{1-\gamma}\{C(\tau)+(1-\gamma)[B(\tau)+C(\tau)X]^2\}\\
J_{WX} &= (1-\gamma)\phi(\tau,X)(W\phi(\tau,\mathbf{X}))^{-\gamma}[B(\tau)+C(\tau)X]\\
J_\tau &= (W\phi(\tau,X))^{1-\gamma}[A'(\tau)+B'(\tau)X+\tfrac{1}{2}C'(\tau)X^2].
\end{aligned} \tag{A2}$$

The optimal allocation π^* becomes:

$$.\pi^* = \left(-\frac{\lambda+r}{\gamma\sigma^2}X^2+\frac{\lambda\mu}{\gamma\sigma^2}X\right)+\frac{1-\gamma}{\gamma}[B(\tau)+C(\tau)X]X. \tag{A3}$$

Substituting (A2) and (A3) into (5), after simplification, we can obtain the following partial differential equation:

$$\begin{aligned}
&-\left[A'(\tau)+B'(\tau)X+\frac{1}{2}C'(\tau)X^2\right]\\
&+\left\{[\lambda(\mu-X)-rX]\left[\left(-\frac{\lambda+r}{\gamma\sigma^2}X+\frac{\lambda\mu}{\gamma\sigma^2}\right)\right.\right.\\
&\left.\left.+\frac{1-\gamma}{\gamma}(B(\tau)+C(\tau)X)\right]+r\right\}\\
&-\frac{\sigma^2\gamma}{2}\left[\left(-\frac{\lambda+r}{\gamma\sigma^2}X+\frac{\lambda\mu}{\gamma\sigma^2}\right)+\frac{1-\gamma}{\gamma}(B(\tau)+C(\tau)X)\right]^2\\
&+\lambda(\mu-X)[B(\tau)+C(\tau)X]\\
&+\frac{\sigma^2}{2}\{C(\tau)+(1-\gamma)[B(\tau)+C(\tau)X]^2\}+\sigma^2(1-\gamma)\\
&\times\left[\left(-\frac{\lambda+r}{\gamma\sigma^2}X+\frac{\lambda\mu}{\gamma\sigma^2}\right)+\frac{1-\gamma}{\gamma}(B(\tau)+C(\tau)X)\right]\\
&\times[B(\tau)+C(\tau)X]=0.
\end{aligned} \tag{A4}$$

The left-hand-side of equation (A4) is a quadratic equation of X. The right-hand-side being 0 forces the coefficients of quadratic term, first term and the constant term to be 0. To be specific, the quadratic term coefficient is:

$$\begin{aligned}
&-\frac{1}{2}C'(\tau)-(\lambda+r)\left[-\frac{\lambda+r}{\gamma\sigma^2}+\frac{1-\gamma}{\gamma}C(\tau)\right]\\
&-\frac{\sigma^2\gamma}{2}\left(-\frac{\lambda+r}{\gamma\sigma^2}+\frac{1-\gamma}{\gamma}C(\tau)\right)^2\\
&-\lambda C(\tau)+\frac{(1-\gamma)\sigma^2}{2}C(\tau)\\
&+\sigma^2(1-\gamma)\left[-\frac{\lambda+r}{\gamma\sigma^2}+\frac{1-\gamma}{\gamma}C(\tau)\right]C(\tau)=0.
\end{aligned} \tag{A5}$$

The first term coefficient is:

$$\begin{aligned}
&-B'(\tau)+\left[-(\lambda+r)\left(\frac{\lambda\mu}{\gamma\sigma^2}+\frac{1-\gamma}{\gamma}B(\tau)\right)\right.\\
&\left.+\lambda\mu\left(\frac{\lambda+r}{\gamma\sigma^2}+\frac{1-\gamma}{\gamma}C(\tau)\right)\right]\\
&-\sigma^2\gamma\left[\left(\frac{\lambda\mu}{\gamma\sigma^2}+\frac{1-\gamma}{\gamma}B(\tau)\right)\left(\frac{\lambda+r}{\gamma\sigma^2}+\frac{1-\gamma}{\gamma}C(\tau)\right)\right]\\
&+\lambda\big(\mu C(\tau)-B(\tau)\big)+\sigma^2(1-\gamma)B(\tau)C(\tau)\\
&+\sigma^2(1-\gamma)\left[\left(\frac{\lambda+r}{\gamma\sigma^2}+\frac{1-\gamma}{\gamma}C(\tau)\right)B(\tau)\right.\\
&\left.+\left(\frac{\lambda\mu}{\gamma\sigma^2}+\frac{1-\gamma}{\gamma}B(\tau)\right)C(\tau)\right]=0.
\end{aligned} \tag{A6}$$

The constant term is:

$$\begin{aligned}
&-A'(\tau)+\left[\lambda\mu\left(\frac{\lambda\mu}{\gamma\sigma^2}+\frac{1-\gamma}{\gamma}B(\tau)\right)+r\right]\\
&-\frac{\sigma^2\gamma}{2}\left(\frac{\lambda\mu}{\gamma\sigma^2}+\frac{1-\gamma}{\gamma}B(\tau)\right)^2+\lambda\mu B(\tau)\\
&+\frac{\sigma^2}{2}[C(\tau)+(1-\gamma)B(\tau)^2]\\
&+\sigma^2(1-\gamma)\left[B(\tau)\left(\frac{\lambda\mu}{\gamma\sigma^2}+\frac{1-\gamma}{\gamma}B(\tau)\right)\right]=0.
\end{aligned} \tag{A7}$$

By rearranging Equations (A5), (A6) and (A7), we can obtain the following ODE system:

$$\begin{aligned}
C'(\tau) &= aC^2(\tau)+bC(\tau)+c\\
B'(\tau) &= aB(\tau)C(\tau)+\frac{b}{2}B(\tau)+dC(\tau)+g\\
A'(\tau) &= \frac{a}{2}B(\tau)^2+dB(\tau)+\frac{\sigma^2}{2}C(\tau)+\frac{(\lambda\mu)^2}{2\gamma\sigma^2}+r
\end{aligned} \tag{A8}$$

with boundary condition $A(0)=B(0)=C(0)=0$ and parameters:

$$\begin{aligned}
a &= \frac{1-\gamma}{\gamma}\sigma^2, \quad b=\frac{2(\gamma r-r-\lambda)}{\gamma}\\
c &= \frac{(\lambda+r)^2}{\gamma\sigma^2}, \quad d=\frac{\lambda\mu}{\gamma}, \quad g=-\frac{\lambda\mu(\lambda+r)}{\gamma\sigma^2}.
\end{aligned} \tag{A9}$$

We start solving the ODE system (A8) from the first equation which only contains $C(\tau)$. Notice it can be solved by evaluating the following integral:

$$\int_0^\tau \frac{\mathrm{d}C}{aC^2(\tau)+bC(\tau)+c}=\tau. \tag{A10}$$

Define:

$$\eta=\sqrt{b^2-4ac}. \tag{A11}$$

Since we only consider $\gamma > 1$, it is easy to check $\eta \in \mathbb{R}$. Then rewrite Equation (A10) as:

$$\frac{1}{\eta}\int_0^\tau \frac{1}{C - \frac{-b-\eta}{2a}} - \frac{1}{C - \frac{-b+\eta}{2a}}\,\mathrm{d}C = \tau. \tag{A12}$$

Together with the boundary condition $C(0) = 0$, we can solve for $C(\tau)$ to obtain:

$$C(\tau) = \frac{2c(1 - e^{-\eta\tau})}{2\eta - (b+\eta)(1 - e^{-\eta\tau})}. \tag{A13}$$

Then we continue by solving for $B(\tau)$ in the second equation in the ODE system (A8), which can be written as:

$$B'(\tau) - \left[aC(\tau) + \frac{b}{2}\right]B(\tau) = dC(\tau) + g. \tag{A14}$$

We denote:

$$\begin{aligned} p(\tau) &= aC(\tau) + \frac{b}{2} \\ f(\tau) &= dC(\tau). \end{aligned} \tag{A15}$$

Then we compute:

$$\begin{aligned} \mu(\tau) &= e^{\int p(\tau)\,\mathrm{d}\tau} = e^{\eta\tau/2}[2\eta - (b+\eta)(1 - e^{-\eta\tau})] \\ \int \mu(\tau)f(\tau)\,\mathrm{d}\tau &= \left[\frac{-4gr}{\eta} + 2g\right]e^{\eta\tau/2} + \left[\frac{-4gr}{\eta} - 2g\right]e^{-\eta\tau/2}. \end{aligned} \tag{A16}$$

Hence, $B(\tau)$ can be derived as:

$$B(\tau) = \frac{1}{\mu(\tau)}\left(\int \mu(\tau)f(\tau)\,\mathrm{d}\tau + \text{constant}\right). \tag{A17}$$

Together with boundary condition $B(0) = 0$, we can solve for $B(\tau)$ to obtain:

$$B(\tau) = \frac{-4gr(1 - e^{-\eta\tau/2})^2 + 2g\eta(1 - e^{\eta\tau})}{\eta[2\eta - (b+\eta)(1 - e^{-\eta\tau})]}. \tag{A18}$$

Finally, $A(\tau)$ can be easily solved by integrating the last equation in (A8) to yield:

$$A(\tau) = \int \frac{a}{2}B(\tau)^2 + dB(\tau) + \frac{\sigma^2}{2}C(\tau) + \frac{(\lambda\mu)^2}{2\gamma\sigma^2} + r\,\mathrm{d}\tau. \tag{A19}$$

Appendix 2. Solution to multi-asset HJB equation

Similar to single asset case, we assume indirect utility J has the following format:

$$\begin{aligned} J &= \frac{(W\phi(\tau,\mathbf{X}))^{1-\gamma} - 1}{1-\gamma} \\ \phi(\tau,\mathbf{X}) &= \exp\left\{A(\tau) + B(\tau)^T\mathbf{X} + \frac{1}{2}\mathbf{X}^T C(\tau)\mathbf{X}\right\}. \end{aligned} \tag{A20}$$

The partial derivatives of J of are:

$$\begin{aligned} J_W &= \phi(\tau,\mathbf{X})(W\phi(\tau,\mathbf{X}))^{-\gamma} \\ J_{WW} &= -\gamma\phi(\tau,\mathbf{X})^2(W\phi(\tau,\mathbf{X}))^{-\gamma-1} \\ J_{\mathbf{X}} &= (W\phi(\tau,\mathbf{X}))^{1-\gamma}[B(\tau) + C(\tau)\mathbf{X}] \\ J_{\mathbf{X}\mathbf{X}^T} &= (W\phi(\tau,\mathbf{X}))^{1-\gamma} \\ &\quad \times \{C(\tau) + (1-\gamma)[B(\tau) + C(\tau)\mathbf{X}][B(\tau) + C(\tau)\mathbf{X}]^T\} \\ J_{W\mathbf{X}} &= (1-\gamma)\phi(\tau,\mathbf{X})(W\phi(\tau,\mathbf{X}))^{-\gamma}[B(\tau) + C(\tau)\mathbf{X}] \\ J_\tau &= (W\phi(\tau,\mathbf{X}))^{1-\gamma}[A'(\tau) + B'(\tau)^T\mathbf{X} + \tfrac{1}{2}\mathbf{X}^T C'(\tau)\mathbf{X}]. \end{aligned} \tag{A21}$$

Hence, the optimal allocation $\boldsymbol{\pi}^*$ becomes:

$$\boldsymbol{\pi}^* = \frac{1}{\gamma}\Sigma_W^{-1}(\mu_W - r\not{1}) + \frac{1-\gamma}{\gamma}\Sigma_W^{-1}\Sigma_{WX}[B(\tau) + C(\tau)\mathbf{X}]. \tag{A22}$$

Substituting (A21) and (A22) into (20), we can obtain the following partial differential equation after simplification:

$$\begin{aligned} &-\left[A'(\tau) + B'(\tau)^T\mathbf{X} + \frac{1}{2}\mathbf{X}^T C'(\tau)\mathbf{X}\right] \\ &+ \frac{1}{2\gamma}(\mu_W - r\not{1})^T\Sigma_W^{-1}(\mu_W - r\not{1}) \\ &+ \frac{1-\gamma}{\gamma}(\mu_W - r\not{1})^T\Sigma_W^{-1}\Sigma_{WX}[B(\tau) + C(\tau)\mathbf{X}] + r \\ &+ \frac{(1-\gamma)^2}{2\gamma}[B(\tau) + C(\tau)\mathbf{X}]^T\Sigma_X[B(\tau) + C(\tau)\mathbf{X}] \\ &+ \mu_X[B(\tau) + C(\tau)X] \\ &+ \frac{1}{2}\,\mathrm{Tr}\{\Sigma_X[C(\tau) + (1-\gamma)(B(\tau) + C(\tau)\mathbf{X})(B(\tau) \\ &+ C(\tau)\mathbf{X})^T]\} = 0. \end{aligned} \tag{A23}$$

Again, the left-hand-side of the equation is a quadratic equation of $\mathbf{X}$, right-hand-side is 0, which means the coefficients of quadratic term, first term and the constant term have to vanish. The quadratic term is:

$$\begin{aligned} &-\frac{1}{2}\mathbf{X}^T C'(\tau)\mathbf{X} + \frac{1}{2\gamma}\{\mathbf{X}^T(\Lambda + r\not{1})^T\Sigma_X^{-1}(\Lambda + r\not{1})\mathbf{X}\} \\ &- \frac{1-\gamma}{\gamma}\mathbf{X}^T(\Lambda + r\not{1})^T C(\tau)\mathbf{X} \\ &+ \frac{(1-\gamma)^2}{2\gamma}\mathbf{X}C(\tau)^T\Sigma_X C(\tau)\mathbf{X} - \mathbf{X}\Lambda C\mathbf{X} \\ &+ \frac{1-\gamma}{2}\mathbf{X}C(\tau)^T\Sigma^X C(\tau)\mathbf{X} = 0. \end{aligned} \tag{A24}$$

The first term is:

$$\begin{aligned} &-B'(\tau)\mathbf{X} - \frac{1}{\gamma}(\Lambda M)^T\Sigma_X^{-1}(\Lambda + r\not{1})\mathbf{X} \\ &+ \frac{1-\gamma}{\gamma}\{M^T\Lambda C(\tau)\mathbf{X} - B(\tau)^T(\Lambda + r\not{1})^T\mathbf{X}\} \\ &+ \frac{(1-\gamma)^2}{\gamma}B(\tau)^T\Sigma_X C(\tau) + M^T\Lambda^T C(\tau)\mathbf{X} - B(\tau)^T\Lambda\mathbf{X} \\ &+ (1-\gamma)\mathbf{X}C(\tau)^T\Sigma_X B(\tau) = 0. \end{aligned} \tag{A25}$$

The constant term is:

$$\begin{aligned} &-A'(\tau) + \frac{1}{2\gamma}(\Lambda M)^T\Sigma_X^{-1}(\Lambda M) \\ &+ \frac{1-\gamma}{\gamma}M^T\Lambda B(\tau) + \frac{(1-\gamma)^2}{2\gamma}B(\tau)^T\Sigma_X B(\tau) \\ &+ r + M^T\Lambda B(\tau) + \frac{1-\gamma}{2}B(\tau)^T\Sigma_X B(\tau) + \frac{1}{2}\,\mathrm{Tr}(\Sigma_X C(\tau)) = 0. \end{aligned} \tag{A26}$$

After further simplification, we can obtain the ODE system:

$$\begin{aligned} C'(\tau) &= \frac{1-\gamma}{\gamma}C(\tau)^T\Sigma_X C(\tau) - \frac{1}{\gamma}[\Lambda C(\tau) + C(\tau)^T\Lambda] \\ &\quad - \frac{2r(1-\gamma)}{\gamma}C(\tau) + \frac{1}{\gamma}(\Lambda + r\not{1})^T\Sigma_X^{-1}(\Lambda + r\not{1}). \end{aligned} \tag{A27}$$

$$\begin{aligned} B'(\tau) &= \left[\frac{1-\gamma}{\gamma}\Sigma_X C(\tau) - \frac{1}{\gamma}\Lambda - \frac{1-\gamma}{\gamma}r\not{1}\right]^T B(\tau) \\ &\quad - \frac{1}{\gamma}(\Lambda + r\not{1})^T\Sigma_X^{-1}\Lambda M + \frac{1}{\gamma}C(\tau)^T\Lambda M \end{aligned} \tag{A28}$$

$$A'(\tau) = \frac{1}{2\gamma}(\Lambda M)^T \Sigma_X^{-1}(\Lambda M) + \frac{1}{\gamma} M^T \Lambda B(\tau) + \frac{1-\gamma}{2\gamma} B(\tau)^T \Sigma_X B(\tau) + r + \frac{1}{2}\,\mathrm{Tr}[\Sigma_X C(\tau)] \quad \text{(A29)}$$

with boundary conditions:

$$A(0) = 0, \quad B(0) = 0, \quad C(0) = 0. \quad \text{(A30)}$$

Appendix 3. Pairs trading data and parameter estimations

Table A1. Stock and ETF pairs Information.

Stock Ticker	Company Name	ETF Ticker	Industry/Sector
CMCSA	COMCAST CORP	XLY	Consumer Discretionary Select Sector SPDR Fund
SBUX	STARBUCKS CORP	XLY	Consumer Discretionary Select Sector SPDR Fund
F	FORD MOTOR CO	XLY	Consumer Discretionary Select Sector SPDR Fund
MDT	METTLER TOLEDO INTERNATIONAL	XLV	Health Care Select Sector SPDR Fund
DUK	DUKE ENERGY CORP	XLU	Utilities Select Sector SPDR Fund
D	DOMINION ENERGY INC	XLU	Utilities Select Sector SPDR Fund
AEP	AMERICAN ELECTRIC POWER	XLU	Utilities Select Sector SPDR Fund
PG	PROCTER & GAMBLE CO	XLP	Consumer Staples Select Sector SPDR Fund
KO	COCA COLA CO	XLP	Consumer Staples Select Sector SPDR Fund
PEP	PEPSICO INC	XLP	Consumer Staples Select Sector SPDR Fund
CL	COLGATE PALMOLIVE CO	XLP	Consumer Staples Select Sector SPDR Fund
MSFT	MICROSOFT CORP	XLK	Technology Select Sector SPDR Fund
INTC	INTEL CORP	XLK	Technology Select Sector SPDR Fund
CSCO	CISCO SYSTEMS INC	XLK	Technology Select Sector SPDR Fund
ORCL	ORACLE CORP	XLK	Technology Select Sector SPDR Fund
T	AT&T INC	XLK	Technology Select Sector SPDR Fund
VZ	VERIZON COMMUNICATIONS INC	XLK	Technology Select Sector SPDR Fund
HON	HONEYWELL INTERNATIONAL INC	XLI	Industrial Select Sector SPDR Fund
JPM	JPMORGAN CHASE & CO	XLF	Financial Select Sector SPDR Fund
WFC	WELLS FARGO & CO	XLF	Financial Select Sector SPDR Fund
IP	INTERNATIONAL PAPER CO	XLB	Materials Select Sector SPDR Fund
WY	WEYERHAEUSER CO	IYR	US Real Estate ETF
PLD	PROLOGIS INC	IYR	US Real Estate ETF
VTR	VENTAS INC	IYR	US Real Estate ETF
BK	BANK OF NEW YORK MELLON CORP	KBE	SPDR S&P Bank ETF
NTRS	NORTHERN TRUST CORP	KBE	SPDR S&P Bank ETF
FITB	FIFTH THIRD BANCORP	KBE	SPDR S&P Bank ETF
C	CITIGROUP INC	KBE	SPDR S&P Bank ETF
PNC	PNC FINANCIAL SERVICES GROUP	KBE	SPDR S&P Bank ETF
RF	REGIONS FINANCIAL CORP	KBE	SPDR S&P Bank ETF
FII	FEDERATED INVESTORS INC	KCE	SPDR S&P Capital Markets ETF
ETFC	E TRADE FINANCIAL CORP	KCE	SPDR S&P Capital Markets ETF
STT	STATE STREET CORP	KCE	SPDR S&P Capital Markets ETF
IVZ	INVESCO LTD	KCE	SPDR S&P Capital Markets ETF
LAZ	LAZARD LTD	KCE	SPDR S&P Capital Markets ETF
TOL	TOLL BROTHERS INC	XHB	SPDR S&P Homebuilders ETF
JCI	JOHNSON CONTROLS INTERNATION	XHB	SPDR S&P Homebuilders ETF
MAS	MASCO CORP	XHB	SPDR S&P Homebuilders ETF
PHM	PULTEGROUP INC	XHB	SPDR S&P Homebuilders ETF
LEN	LENNAR CORP	XHB	SPDR S&P Homebuilders ETF
ODP	OFFICE DEPOT INC	XRT	SPDR S&P Retail ETF
URBN	URBAN OUTFITTERS INC	XRT	SPDR S&P Retail ETF
CY	CYPRESS SEMICONDUCTOR CORP	XSD	SPDR S&P Semiconductor ETF
ADI	ANALOG DEVICES INC	XSD	SPDR S&P Semiconductor ETF
ATI	ALLEGHENY TECHNOLOGIES INC	XME	SPDR S&P Metals & Mining ETF
HAL	HALLIBURTON CO	XES	SPDR S&P Oil & Gas Equipment & Services ETF
PTEN	PATTERSON UTI ENERGY INC	XES	SPDR S&P Oil & Gas Equipment & Services ETF
SPN	SUPERIOR ENERGY SERVICES INC	XES	SPDR S&P Oil & Gas Equipment & Services ETF

Table A2. Λ for Years 2014–2017.

Ticker	2014	2015	2016	2017
CMCSA	0.1512044	0.09903023	0.088121957	0.079624755
SBUX	0.119861566	0.106692767	0.098077297	0.089953586
F	0.066364963	0.050982492	0.04489592	0.043963961
MDT	0.107307167	0.102353912	0.086329823	0.08685961
DUK	0.121337593	0.08755698	0.058748914	0.061531854
D	0.089492081	0.098878838	0.067219952	0.066434508
AEP	0.113546658	0.09884012	0.078860184	0.072107818
PG	0.126228638	0.096886326	0.069082293	0.05817882
KO	0.115512178	0.096943248	0.081430882	0.069477828
PEP	0.149839963	0.12689585	0.111955847	0.087106367
CL	0.174192155	0.135077399	0.105763485	0.082967234
MSFT	0.14835767	0.13526843	0.085187717	0.080934768
INTC	0.085382411	0.076182585	0.0530736	0.045079178
CSCO	0.10030578	0.083682861	0.081974905	0.07533517
ORCL	0.078216727	0.072394562	0.062725916	0.058971221
T	0.082766055	0.045339141	0.05609628	0.022236044
VZ	0.096536174	0.090379316	0.078847628	0.073606135
HON	0.194247732	0.128088189	0.12284851	0.08929841
JPM	0.077781933	0.082246427	0.062812423	0.058392144
WFC	0.140411733	0.101826513	0.08560014	0.069309928
IP	0.110442126	0.094170262	0.077487284	0.064448131
WY	0.148304805	0.111528619	0.084138711	0.059737781
PLD	0.127881484	0.124128489	0.10390303	0.083838078
VTR	0.123566047	0.103444615	0.094639029	0.092862104
BK	0.152665573	0.086988846	0.066685143	0.06256798
NTRS	0.085238352	0.09623448	0.080502361	0.078105738
FITB	0.185831425	0.146666291	0.119258936	0.087743548
C	0.121257087	0.097341215	0.085707487	0.080808048
PNC	0.111797979	0.076926986	0.07132905	0.068030345
RF	0.13340155	0.113899191	0.066644951	0.065864507
FII	0.070298788	0.0692512	0.063582922	0.048911198
ETFC	0.106017484	0.076158733	0.08130912	0.07331601
STT	0.069168964	0.067691624	0.076441998	0.068969932
IVZ	0.084301846	0.079766503	0.067302767	0.060173509
LAZ	0.165662314	0.162293273	0.148215258	0.122143918
TOL	0.120344576	0.101937009	0.101144192	0.092829882
JCI	0.134428205	0.12877133	0.121046752	0.077376392
MAS	0.124948355	0.101403999	0.085898027	0.068374814
PHM	0.105976914	0.093875333	0.069488354	0.058985726
LEN	0.102457074	0.06532253	0.05861495	0.054436747
ODP	0.130774618	0.069522574	0.069515129	0.05851729
URBN	0.099900322	0.07510007	0.054682685	0.044866865
CY	0.066689969	0.043098203	0.03907061	0.043387069
ADI	0.085303172	0.08190453	0.089768265	0.077714042
ATI	0.127979297	0.089665411	0.071413592	0.068926531
HAL	0.111331436	0.106928803	0.087892309	0.092583281
PTEN	0.087740809	0.08052275	0.072652146	0.069694441
SPN	0.120296202	0.102141582	0.091339108	0.069543253