

Deep Learning for Portfolio Optimization

ZIHAO ZHANG, STEFAN ZOHREN, AND STEPHEN ROBERTS

ZIHAO ZHANG

is a D.Phil. student with the Oxford-Man Institute of Quantitative Finance and the Machine Learning Research Group at the University of Oxford in Oxford, UK.
zihao.zhang@worc.ox.ac.uk

STEFAN ZOHREN

is an associate professor (research) with the Oxford-Man Institute of Quantitative Finance and the Machine Learning Research Group at the University of Oxford in Oxford, UK.
zohren@robots.ox.ac.uk

STEPHEN ROBERTS

is the director of the Oxford-Man Institute of Quantitative Finance, the founding director of the Oxford Centre for Doctoral Training in Autonomous Intelligent Machines and Systems, and the Royal Academy of Engineering/Man Group Professor in the Machine Learning Research Group at the University of Oxford in Oxford, UK.
sjrob@robots.ox.ac.uk

*All articles are now categorized by topics and subtopics. **View at PM-Research.com.**

KEY FINDINGS

- In this article, the authors utilize deep learning models to directly optimize the portfolio Sharpe ratio. They present a framework that bypasses traditional forecasting steps and allows portfolio weights to be optimized by updating model parameters.
- The authors trade exchange-traded funds of market indexes to form a portfolio. Doing this substantially reduces the scope of possible assets to choose from, and these indexes have shown robust correlations.
- The authors backtest their methods from 2011 to the end of April 2020, including the financial instabilities due to COVID-19. Their model delivers good performance under transaction costs, and a detailed study shows the rationality of their approach during the crisis.

ABSTRACT: *In this article, the authors adopt deep learning models to directly optimize the portfolio Sharpe ratio. The framework they present circumvents the requirements for forecasting expected returns and allows them to directly optimize portfolio weights by updating model parameters. Instead of selecting individual assets, they trade exchange-traded funds of market indexes to form a portfolio. Indexes of different asset classes show robust correlations, and trading them substantially reduces the spectrum of available assets from which to choose. The authors compare their method with a wide range of algorithms, with results showing that the model obtains the best performance over the testing period of 2011 to the end of April 2020, including the financial instabilities of the first quarter of 2020. A sensitivity analysis is included to clarify the relevance of input features, and the authors further study the performance of their approach under different cost rates and different risk levels via volatility scaling.*

TOPICS: *Exchange-traded funds and applications, mutual fund performance, portfolio construction**

Portfolio optimization is an essential component of a trading system. The optimization aims to select the best asset distribution within a portfolio to maximize returns at a given risk level. This theory was pioneered by Markowitz (1952) and is widely known as modern portfolio theory (MPT). The main benefit of constructing such a portfolio comes from the promotion of diversification that smooths the equity curve, leading to a higher return per risk than trading an individual asset. This observation has been proven (see, e.g., Zivot 2017), showing that the risk (volatility) of a long-only portfolio is always lower than

that of an individual asset, for a given expected return, as long as assets are not perfectly correlated. We note that this is a natural consequence of Jensen's inequality (Jensen 1906).

Despite the undeniable power of such diversification, selection of the right asset allocations in a portfolio is not straightforward because the dynamics of financial markets change significantly over time. Assets that exhibit, for example, strong negative correlations in the past could be positively correlated in the future. This adds extra risk to the portfolio and degrades subsequent performance. Furthermore, the universe of available assets for constructing a portfolio is enormous. Taking the US stock markets as a single example, more than 5,000 stocks are available from which to choose (Wild 2008). Indeed, a well-rounded portfolio consists not only of stocks but also is typically supplemented with bonds and commodities, further expanding the spectrum of choices.

In this article, we consider directly optimizing a portfolio, using deep learning models (LeCun, Bengio, and Hinton 2015; Goodfellow, Bengio, and Courville 2016). Unlike classical methods (Markowitz 1952), in which expected returns are first predicted (typically through econometric models), we bypass this forecasting step to directly obtain asset allocations. Several works (Moody et al. 1998; Moody and Saffell 2001; Zhang, Zohren, and Stephen 2020) have shown that the return forecasting approach is not guaranteed to maximize the performance of a portfolio because the prediction

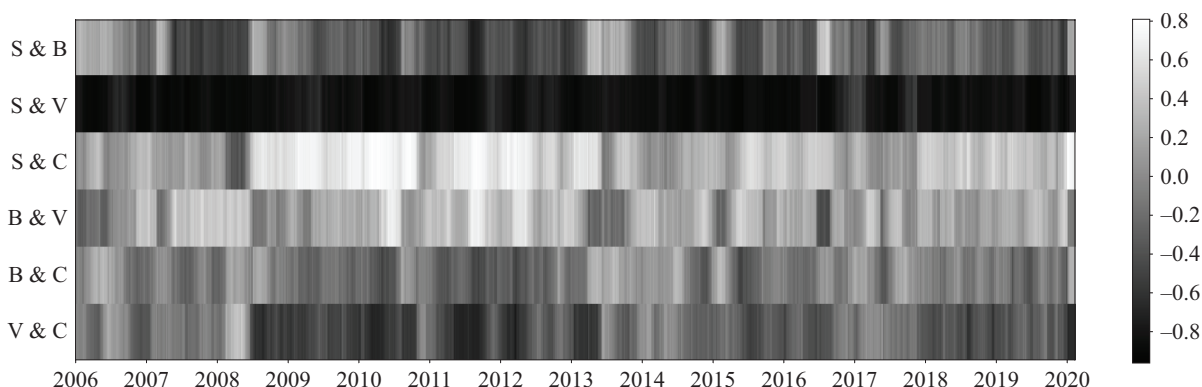
steps attempt to minimize a prediction loss, which is not the overall reward from the portfolio. In contrast, our approach is to directly optimize the Sharpe ratio (Sharpe 1994), thus maximizing return per unit of risk. Our framework starts by concatenating multiple features from different assets to form a single observation and then uses a neural network to extract salient information and output portfolio weights so as to maximize the Sharpe ratio.

Instead of choosing individual assets, exchange-traded funds (ETFs) (Gastineau 2008) of market indexes are selected to form a portfolio. We use four market indexes: US total stock index (VTI), US aggregate bond index (AGG), US commodity index (DBC), and the Volatility Index (VIX). All of these indexes are popularly traded ETFs that offer high liquidity and relatively small expense ratios. Trading indexes substantially reduces the possible universe of asset choices and gains exposure to most securities. Furthermore, these indexes are generally uncorrelated, or even negatively correlated, as shown in Exhibit 1. Individual instruments in the same asset class, however, often exhibit strong positive correlations. For example, more than 75% stocks are highly correlated with the market index (Wild 2008); thus, adding them to a portfolio helps less with diversification.

We are aware that subsector indexes, rather than the total market index, can be included in a portfolio; subindustries perform at different levels, and a weighting on good performance in a sector would therefore deliver extra returns. However, we see subsector indexes as

EXHIBIT 1

Heatmap for Rolling Correlations between Different Index Pair



B = bond index; C = commodity index; S = stock index; V = volatility index.

highly correlated; thus, adding them again provides minimal diversification for the portfolio and risks lowering returns per unit risk. If higher returns are desired, we can use (for example) volatility scaling to upweight our positions and amplify returns. We therefore do not believe there is a need to find the best-performing sector. Instead, we aim to provide a portfolio that delivers high return per unit risk and allows for volatility scaling (Moskowitz, Ooi, and Pedersen 2012; Harvey et al. 2018; Lim, Zohren, and Roberts 2019) to achieve desired return levels.

The remainder of the article is structured as follows. We first introduce the relevant literature and present our methodology. We then describe our experiments and detail the results of our method compared with a range of baseline algorithms. At the end, we summarize our findings and discuss possible future work.

LITERATURE REVIEW

In this section, we review popular portfolio optimization methods and discuss how deep learning models have been applied to this field. A vast literature is available on this topic, so we aim merely to highlight key concepts, popular in the industry or in academic study. One of the popular practical approaches is the reallocation strategy (Wild 2008) adopted by many pension funds (e.g., LifeStrategy Equity Fund, Vanguard). This approach constructs a portfolio by investing only in stocks and bonds. A typical risk-moderate portfolio would, for example, comprise 60% equities and 40% bonds, and the portfolio needs to be rebalanced only semi-annually or annually to maintain this allocation ratio. The method delivers good performance over the long term; however, the fixed allocation ratio means that investors who prefer to place more weight on stocks need to tolerate potentially large drawdowns during dull markets.

Mean-variance analysis or MPT (Markowitz 1952) is used for many institutional portfolios that solve a constraint optimization problem to derive portfolio weights. Despite its popularity, the assumptions of the theory face criticism because they are often not obeyed in real financial markets. In particular, returns are assumed to follow a Gaussian distribution in MPT; therefore, investors only consider expected return and variance of the portfolio returns to make decisions. However, it is widely accepted (see, e.g., Cont and Nitiens 1999; Zhang, Zohren, and Roberts 2019b) that returns tend

to have fat tails and extreme losses are more likely to occur in practice, leading to severe drawdowns that are not bearable. The maximum diversification (MD) portfolio is another promising method, introduced by Choueifaty and Coignard (2008), that aims to maximize the diversification of a portfolio, thereby aiming to have minimally correlated assets so the portfolio can achieve higher returns (and lower risk) than other classical methods. We compare our model with both these strategies, and results suggest that our methods deliver better performance and tolerate larger transaction costs than either of these benchmarks.

Stochastic portfolio theory (SPT) was recently proposed by Fernholz (2002) and Fernholz and Karatzas (2009). Unlike other methods, SPT aims to achieve relative arbitrages, meaning to select portfolios that can outperform a market index with probability of one. Such investment strategies have been studied by Fernholz and Karatzas (2010, 2011), Ruf (2013), and Wong (2015). However, the number of relative arbitrage strategies remains small because theory does not suggest how to construct such strategies. We can check whether a given strategy is a relative arbitrage, but it is nontrivial to develop one *ex ante*. In this article, we include a particular class of SPT called the functionally generated portfolio (Fernholz 1999) in our experiment, but the result suggests this method delivers inferior performance compared with other algorithms and generates large turnovers, making it unprofitable under heavy transaction costs.

The idea of our end-to-end training framework was first initiated by Moody et al. (1998) and Moody and Saffell (2001). However, they mainly focused on optimizing the performance for a single asset, so there is little discussion of how portfolios should be maximized. Furthermore, their testing period is from 1970 to 1994, whereas our dataset is up to date and we study the behavior of our strategy under the current crisis due to COVID-19. We can also link our approach to reinforcement learning (RL) (Williams 1992; Mnih et al. 2013; Sutton and Barto 2018), in which an agent interacts with an environment to maximize cumulative rewards. Bertoluzzo and Corazza (2012), Huang (2018), and Zhang, Zohren, and Stephen (2020) have studied this stream and adopted RL to design trading strategies. However, the goal of RL is to maximize expected cumulative rewards such as profits, whereas the Sharpe ratio cannot be directly optimized.

METHODOLOGY

In this section, we introduce our framework and discuss how the Sharpe ratio can be optimized through gradient ascent. We discuss the types of neural networks used and detail the functionality of each component in our method.

Objective Function

The Sharpe ratio is used to gauge the return per risk of a portfolio and is defined as expected return over volatility (excluding the risk-free rate for simplicity):

$$L = \frac{E(R_p)}{\text{Std}(R_p)} \quad (1)$$

where $E(R_p)$ and $\text{Std}(R_p)$ are the estimates of the mean and standard deviation of portfolio returns. Specifically, for a trading period of $t = \{1, \dots, T\}$, we can maximize the following objective function:

$$L_T = \frac{E(R_{p,t})}{\sqrt{E(R_{p,t}^2) - (E(R_{p,t}))^2}} \quad (2)$$

$$E(R_{p,t}) = \frac{1}{T} \sum_{t=1}^T R_{p,t}$$

where $R_{p,t}$ is realized portfolio return over n assets at time t denoted as

$$R_{p,t} = \sum_{i=1}^n w_{i,t-1} \cdot r_{i,t} \quad (3)$$

where $r_{i,t}$ is the return of asset i with $r_{i,t} = (p_{i,t}/p_{i,t-1} - 1)$. We represent the allocation ratio (position) of asset i as $w_{i,t} \in [0, 1]$ and $\sum_i w_{i,t} = 1$. In our approach, a neural network f with parameters θ is adopted to model $w_{i,t}$ for a long-only portfolio:

$$w_{i,t} = f(\theta|x_t) \quad (4)$$

where x_t represents the current market information and we bypass the classical forecasting step by linking the inputs with positions to maximize the Sharpe over trading period T , namely L_T . However, a long-only portfolio imposes constraints that require weights to be positive and summed to one; we use softmax outputs to fulfill these requirements:

$$w_{i,t} = \frac{\exp(\tilde{w}_{i,t})}{\sum_j^n \exp(\tilde{w}_{j,t})}, \quad \text{in which } \tilde{w}_{i,t} \text{ are the raw weights.} \quad (5)$$

Such a framework can be optimized using unconstrained optimization methods. In particular, we use gradient ascent to maximize the Sharpe ratio. The gradient of L_T with respect to parameters θ is readily calculable, with an excellent derivation presented by Moody et al. (1998) and Molina (2016). Once we obtain $\partial L_T / \partial \theta$, we can repeatedly compute this value from training data and update the parameters by using gradient ascent:

$$\theta_{\text{new}} := \theta_{\text{old}} + \alpha \frac{\partial L_T}{\partial \theta} \quad (6)$$

where α is the learning rate, and the process can be repeated for many epochs until the convergence of Sharpe ratio or the optimization of validation performance is achieved.

Model Architecture

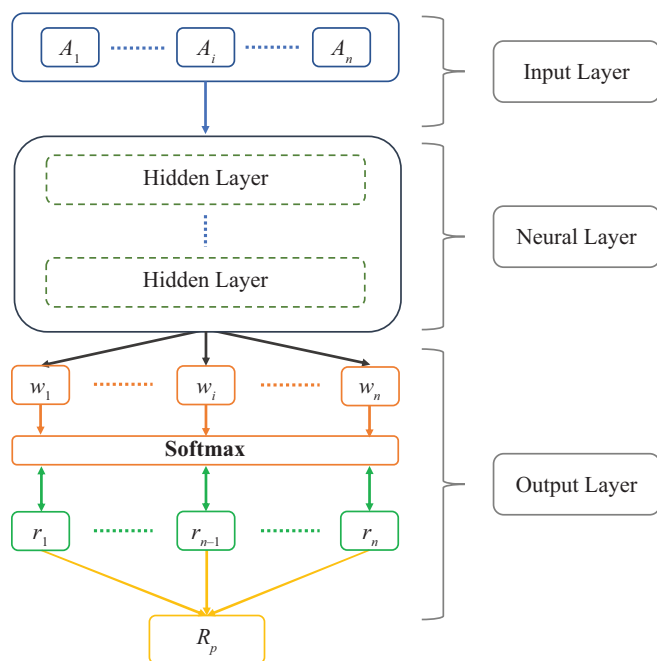
We depict our network architecture in Exhibit 2. Our model consists of three main building blocks: input layer, neural layer, and output layer. The idea of this design is to use neural networks to extract cross-sectional features from input assets. Features extracted from deep learning models have been suggested to perform better than traditional hand-crafted features (Zhang, Zohren, and Stephen 2020). Once features have been extracted, the model outputs portfolio weights, and we obtain realized returns to maximize the Sharpe ratio. We detail each component of our method.

Input layer. We denote each asset as A_i , and we have n assets to form a portfolio. A single input is prepared by concatenating information from all assets. For example, the input features of one asset can be its past prices and returns, with a dimension of $(k, 2)$, in which k represents the lookback window. By stacking features across all assets, the dimension of the resulting input would be $(k, 2 \times n)$. We can then feed this input to the network and expect nonlinear features to be extracted.

Neural layer. A series of hidden layers can be stacked to form a network; however, in practice, this part requires many experiments because there are plentiful ways of combining hidden layers and performance

EXHIBIT 2

Model Architecture Schematic



Note: Overall, our model contains three main building blocks: input layer, neural layer, and output layer.

often depends on the architecture design. We have tested deep learning models including fully connected neural network (FCN) (Goodfellow, Bengio, and Courville 2016), convolutional neural network (CNN) (Krizhevsky, Sutskever, and Hinton 2012), and long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997). Overall, LSTMs deliver the best performance for modeling daily financial data, and a number of works (Tsantekidis et al. 2017; Lim, Zohren, and Roberts 2019; Zhang, Zohren, and Stephen 2020) support this observation.

We note a problem of FCN: severe overfitting. Because it assigns parameters to each input feature, this results in an excess number of parameters. The LSTM operates with a cell structure that has gate mechanisms to summarize and filter information from its long history, so the model ends up with fewer trainable parameters and achieves better generalization results. In contrast, CNNs with strong smoothing (typical of large convolutional filters) tend to have underfitting problems, such that overly smooth solutions are obtained. Because of the design of parameter sharing and the convolution

operations, CNNs overfilter the inputs, in our experience. However, we note that CNNs appear to be excellent candidates for modeling high-frequency financial data such as limit order books (Zhang, Zohren, and Roberts 2019a).

Output layer. To construct a long-only portfolio, we use the *softmax* activation function for the output layer, which naturally imposes constraints to keep portfolio weights positive and summing to one. The number of output nodes (w_1, \dots, w_n) is equal to the number of assets in our portfolio, and we can multiply these portfolio weights with associated assets' returns (r_1, \dots, r_n) to calculate realized portfolio returns (R_p). Once realized returns are obtained, we can derive the Sharpe ratio and calculate the gradients of the Sharpe ratio with respect to the model parameters and use gradient ascent to update the parameters.

EXPERIMENTS

Description of Dataset

We use four market indexes: US VTI, US AGG, US DBC, and VIX. These are popular ETFs (Gastineau 2008) that have existed for more than 15 years. As discussed before, trading indexes offers advantages over trading individual assets because these indexes are generally uncorrelated, resulting in diversification. A diversified portfolio delivers a higher return per risk, and the idea of our strategy is to have a system that delivers a good reward-to-risk ratio.

Our dataset ranges from 2006 to 2020 and contains daily observations. We retrain our model at every two years and use all data available up to that point to update parameters. Overall, our testing period is from 2011 to the end of April 2020, including the most recent crisis due to COVID-19.

Baseline Algorithms

We compare our method with a group of baseline algorithms. The first set of baseline models are reallocation strategies adopted by many pension funds. These strategies assign a fixed allocation ratio to relevant assets and rebalance portfolios annually to maintain these ratios. Investors can select a portfolio based on their risk preferences. In general, portfolios weighted more on

equities would deliver better performance at the expense of greater volatility. In this article, we consider four such strategies: Allocation 1 (25% shares, 25% bonds, 25% commodities, and 25% volatility index), Allocation 2 (50% shares, 10% bonds, 20% commodities, and 20% volatility index), Allocation 3 (10% shares, 50% bonds, 20% commodities, and 20% volatility index), and Allocation 4 (40% shares, 40% bonds, 10% commodities, and 10% volatility index).

The second set of comparison models are mean-variance optimization (MV) (Markowitz 1952) and MD (Theron and Van Vuuren 2018). We use moving averages with a rolling window of 50 days to estimate the expected returns and covariance matrix. The portfolio weights are updated on a daily basis, and we select weights that maximize Sharpe ratio for MV. The last baseline algorithm is the diversity-weighted portfolio (DWP) from SPT presented by Samo and Vervuurt (2016). The DWP relates portfolio weights to assets' market capitalization, and it has been suggested to be able to outperform the market index with certainty (Fernholz, Karatzas, and Kardaras 2005).

Training Scheme

In this article, we use a single layer of LSTM connectivity, with 64 units, to model the portfolio weights and then to optimize the Sharpe ratio. We purposely keep our network simple to indicate the effectiveness of this end-to-end training pipeline instead of carefully fine-tuning the right hyperparameters. Our input contains close prices and daily returns for each market index, and we take the past 50 days of these observations to form a single input. We are aware that returns can be derived from prices, but keeping returns helps with the evaluation of Equation 7, and we can treat them as momentum features as done by Moskowitz, Ooi, and Pedersen (2012). Because our focus is not on feature selection, we choose these commonly used features in our work. The Adam optimizer (Kingma and Ba 2015) is used for training our network, and the mini-batch size is 64. We take 10% of any training data as a separate validation set to optimize hyperparameters and control overfitting problems. Any hyperparameter optimization is done on the validation set, leaving the test data for the final performance evaluation and ensuring the validity of our results. In general, our training process stops after 100 epochs.

Experimental Results

When reporting the test performance, we include transaction costs and use volatility scaling (Moskowitz, Ooi, and Pedersen 2012; Lim, Zohren, and Roberts 2019; Zhang, Zohren, and Stephen 2020) to scale our positions based on market volatility. We can set our own volatility target and meet the expectations of investors with different risk preferences. Once volatilities are adjusted, our investment performances are mainly driven by strategies instead of being heavily affected by markets. The modified portfolio return can be defined as

$$R_{p,t} = \sum_i \frac{\sigma_{tgt}}{\sigma_{i,t-1}} w_{i,t-1} \cdot r_{i,t} - C \cdot \sum_i \left| \frac{\sigma_{tgt}}{\sigma_{i,t-1}} w_{i,t-1} - \frac{\sigma_{tgt}}{\sigma_{i,t-2}} w_{i,t-2} \right| \quad (7)$$

where σ_{tgt} is the volatility target and $\sigma_{i,t-1}$ is an ex ante volatility estimate of asset i calculated using an exponentially weighted moving standard deviation with a 50-day window on $r_{i,t}$. We use daily changes of the traded value of an asset to represent transaction costs, which is calculated by the second term in Equation 7. C ($= 1\text{bs} = 0.0001$) is the cost rate, and we change it to reflect how our model performs under different transaction costs.

To evaluate the performance of our methods, we use the following metrics: expected return ($E(R)$), standard deviation of return ($\text{Std}(R)$), Sharpe ratio (Sharpe 1994), downside deviation of return ($\text{DD}(R)$) (McNeil, Frey, and Embrechts 2015), and Sortino ratio (Sortino and Price 1994). All of these metrics are annualized, and we also report on maximum drawdown (MDD) (Chekhlov, Uryasev, and Zabarankin 2005), percentage of positive return (% of + Ret), and the ratio between positive and negative return (Ave. P/Ave. L).

Exhibit 3 presents the results of our model (DLS) compared to other baseline algorithms. The top of the exhibit shows the results without using volatility scaling, and we can see that our model achieves the best Sharpe ratio and Sortino ratio, delivering the highest return per risk. However, given the large differences in volatilities, we cannot directly compare expected and cumulative returns for different methods; thus, volatility scaling also helps to make fair comparisons.

Once volatilities are scaled (shown in the middle of Exhibit 3), DLS delivers the best performance across all evaluation metrics except for a slightly larger drawdown. If we look at the cumulative returns in Exhibit 4,

EXHIBIT 3

Experiment Results for Different Algorithms

	E(R)	Std(R)	Sharpe	DD(R)	Sortino	MDD	% of + Ret	Ave. P Ave. L
Panel A: Novolatility Scaling and $C = 0.01\%$								
Allocation 1	0.282	0.303	0.929	0.136	2.065	0.142	0.479	1.193
Allocation 2	0.249	0.212	1.173	0.095	2.616	0.097	0.483	1.254
Allocation 3	0.228	0.256	0.890	0.116	1.962	0.122	0.476	1.183
Allocation 4	0.152	0.123	1.228	0.052	2.932	0.081	0.505	1.349
MV	0.082	0.108	0.759	0.069	1.192	0.195	0.562	1.199
MD	0.462	0.523	0.882	0.239	1.931	0.273	0.473	1.182
DWP	0.051	0.102	0.493	0.067	0.740	0.179	0.549	1.107
DLS	0.313	0.168	1.858	0.099	3.135	0.102	0.537	1.518
Panel B: Volatility Scaling ($\sigma_{tgt} = 0.10$) and $C = 0.01\%$								
Allocation 1	0.160	0.105	1.526	0.061	2.629	0.111	0.554	1.289
Allocation 2	0.123	0.106	1.146	0.065	1.861	0.127	0.549	1.211
Allocation 3	0.145	0.105	1.383	0.061	2.396	0.105	0.542	1.259
Allocation 4	0.164	0.104	1.579	0.064	2.588	0.112	0.565	1.303
MV	0.112	0.100	1.120	0.063	1.767	0.211	0.561	1.213
MD	0.157	0.106	1.484	0.065	2.414	0.125	0.565	1.297
DWP	0.089	0.109	0.818	0.069	1.291	0.115	0.556	1.148
DLS	0.206	0.105	1.962	0.062	3.322	0.123	0.559	1.375
Panel C: Volatility Scaling ($\sigma_{tgt} = 0.10$) and $C = 0.1\%$								
Allocation 1	0.133	0.105	1.274	0.061	2.172	0.113	0.548	1.236
Allocation 2	0.105	0.107	0.986	0.066	1.590	0.244	0.547	1.179
Allocation 3	0.117	0.105	1.110	0.061	1.903	0.107	0.538	1.203
Allocation 4	0.135	0.104	1.299	0.064	2.108	0.114	0.559	1.244
MV	0.019	0.101	0.191	0.066	0.293	0.324	0.537	1.033
MD	0.095	0.106	0.899	0.066	1.431	0.145	0.549	1.171
DWP	-0.083	0.110	-0.753	0.074	-1.129	0.627	0.508	0.880
DLS	0.148	0.105	1.403	0.063	2.327	0.125	0.547	1.272

DLS shows outstanding performance over the long haul, and the MDD is reasonable, ensuring investors will have the confidence needed to hold through hard times. Furthermore, if we look at the bottom of Exhibit 3, in which a large cost rate ($C = 0.1\%$) is used, our model (DLS) still delivers the best expected return and achieves the highest Sharpe and Sortino ratios.

However, with a higher cost rate, we can see that reallocation strategies work well. In particular, Allocations 3 and 4 achieve results comparable to our method. To investigate why the performance gap diminishes with a higher cost rate, we present the boxplots for annual realized trade returns and accumulated costs for different assets in Exhibit 5. Overall, our model delivers better realized returns than reallocation strategies, but we also accumulate much larger transaction costs because our

positions are adjusted on a daily basis, leading to higher turnover.

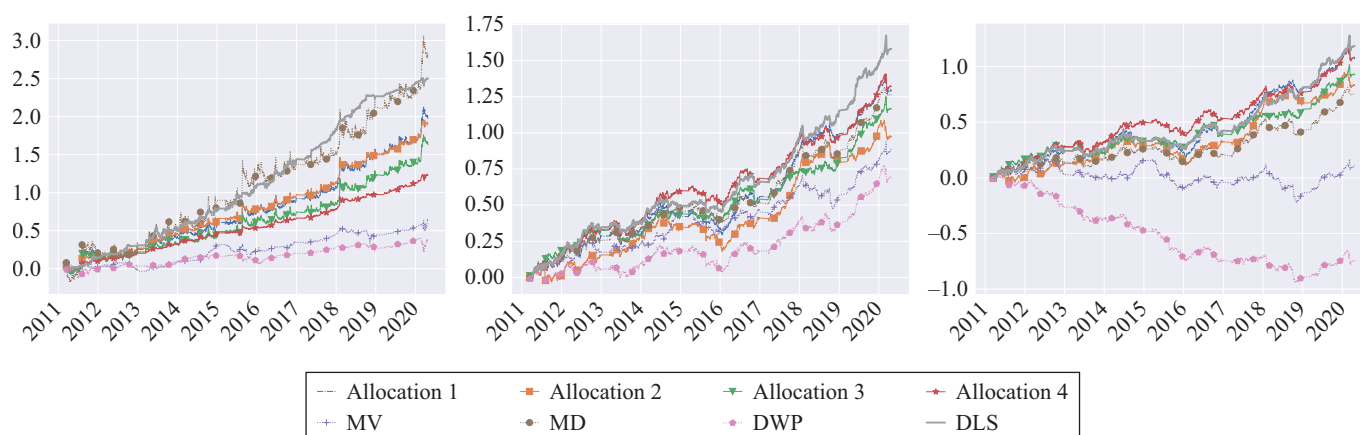
For reallocation strategies, daily position changes are only updated for volatility scaling. Otherwise, we only actively change positions once a year to rebalance and maintain the allocation ratio. As a result, reallocation strategies deliver minimal transaction costs. This analysis aims to indicate the validity of our results and show that our method can work under unfavorable conditions.

Model Performance during 2020 Crisis

Due to the recent COVID-19 pandemic, global stock markets fell dramatically and experienced extreme volatility. The crash started on February 24, 2020, when

EXHIBIT 4

Cumulative Returns (logarithmic scale)



Notes: Left: no volatility scaling and $C = 0.01\%$. Middle: volatility scaling ($\sigma_{igt} = 0.10$) and $C = 0.01\%$. Right: volatility scaling ($\sigma_{igt} = 0.10$) and $C = 0.1\%$.

markets reported their largest one-week declines since the 2008 financial crisis. Later, with an oil price war between Russia and the OPEC countries, markets further dampened and encountered the largest single-day percentage drop since Black Monday in 1987. As of March 2020, we have seen a downturn of at least 25% in the US markets and 30% in most G20 countries. The crisis shattered many investors' confidence and resulted in a great loss of their wealth. However, the crisis also provides a great opportunity to stress test our method and understand how our model performs during the crisis.

To study the model behavior, we plot how our algorithm allocated the assets from January to April 2020 in Exhibit 6. At the beginning of 2020, we can see that our model had a quite diverse holding. However, after a small dip in stock index in early February, we had almost only bonds in our portfolio. There were some equity positions left, but very small positions for volatility and commodity indexes. When the crash started on February 24, our holdings were concentrated in the bond index, which is considered to be a safe asset during the crisis. Interestingly, the bond index also fell at this time (in the middle of March), although it rebounded quite quickly. During the fall in bonds, our original positions did not change much, but the scaled positions decreased greatly for the bond index owing to spiking volatility; therefore, our drawdown was small. Overall, we can see that our model delivers reasonable allocations

during the crisis, and our positions are protected through volatility scaling.

Sensitivity Analysis

To understand how input features affect our decisions, we study the sensitivity analysis presented by Moody and Saffell (2001) for our method. The absolute normalized sensitivity of feature x_i is defined as

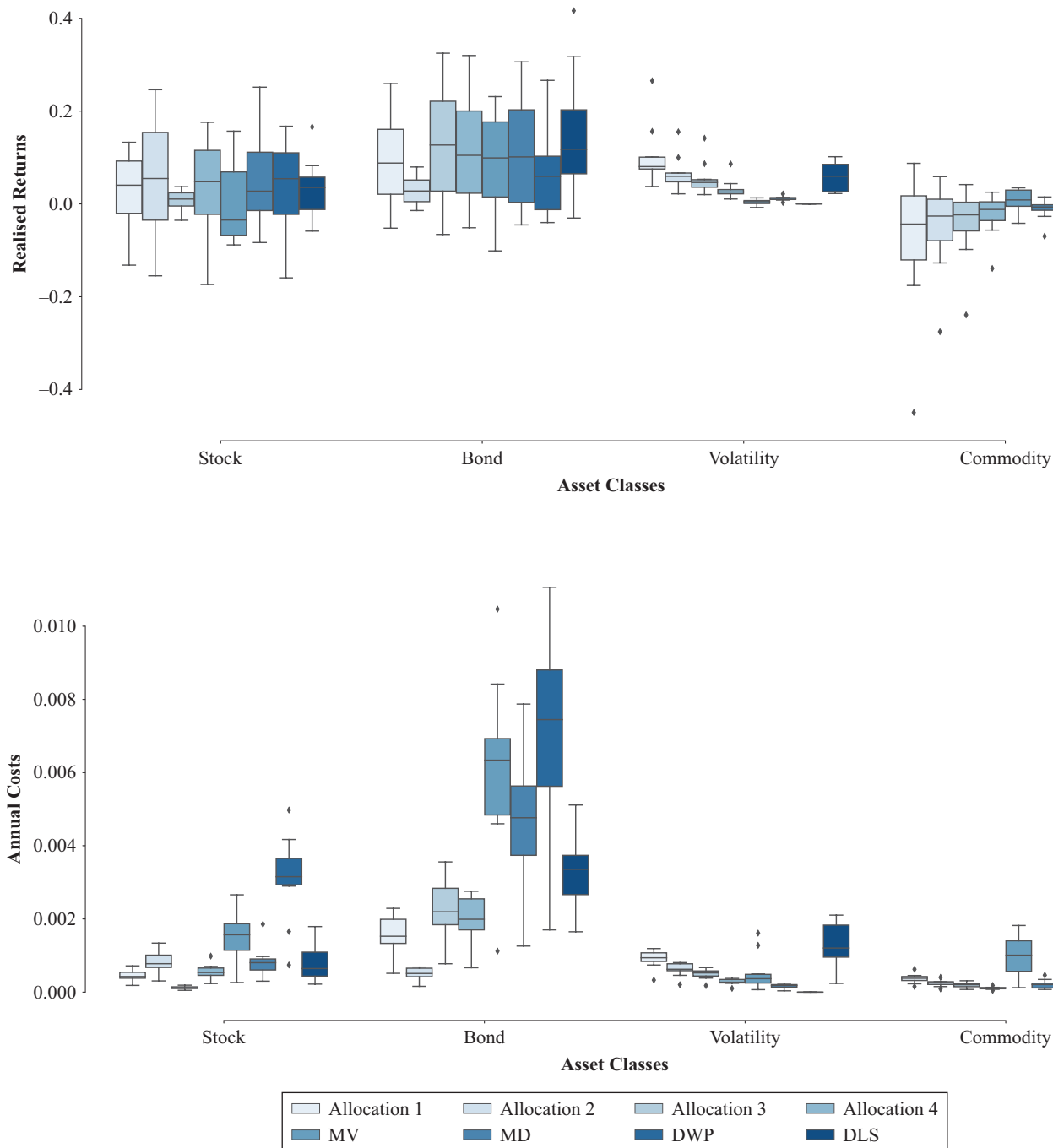
$$S_i = \frac{\frac{dL}{dx_i}}{\max_j \left| \frac{dL}{dx_j} \right|} \quad (8)$$

where L represents the objective function and S_i captures the relative sensitivity for feature x_i compared with other features. We plot the time-varying sensitivities for all features in Exhibit 7. The y -axis indicates the 400 features we have: We use four indexes (each with prices and returns) and take a timeframe of the past 50 observations to form a single input, so there are 400 features in total. The row labeled Sprice represents price features for the stock index, and the bottom of row Sprice is the most recent price for that observation. The same convention is used for all other features.

The importance of features varies over time, but the most recent features always make the biggest

EXHIBIT 5

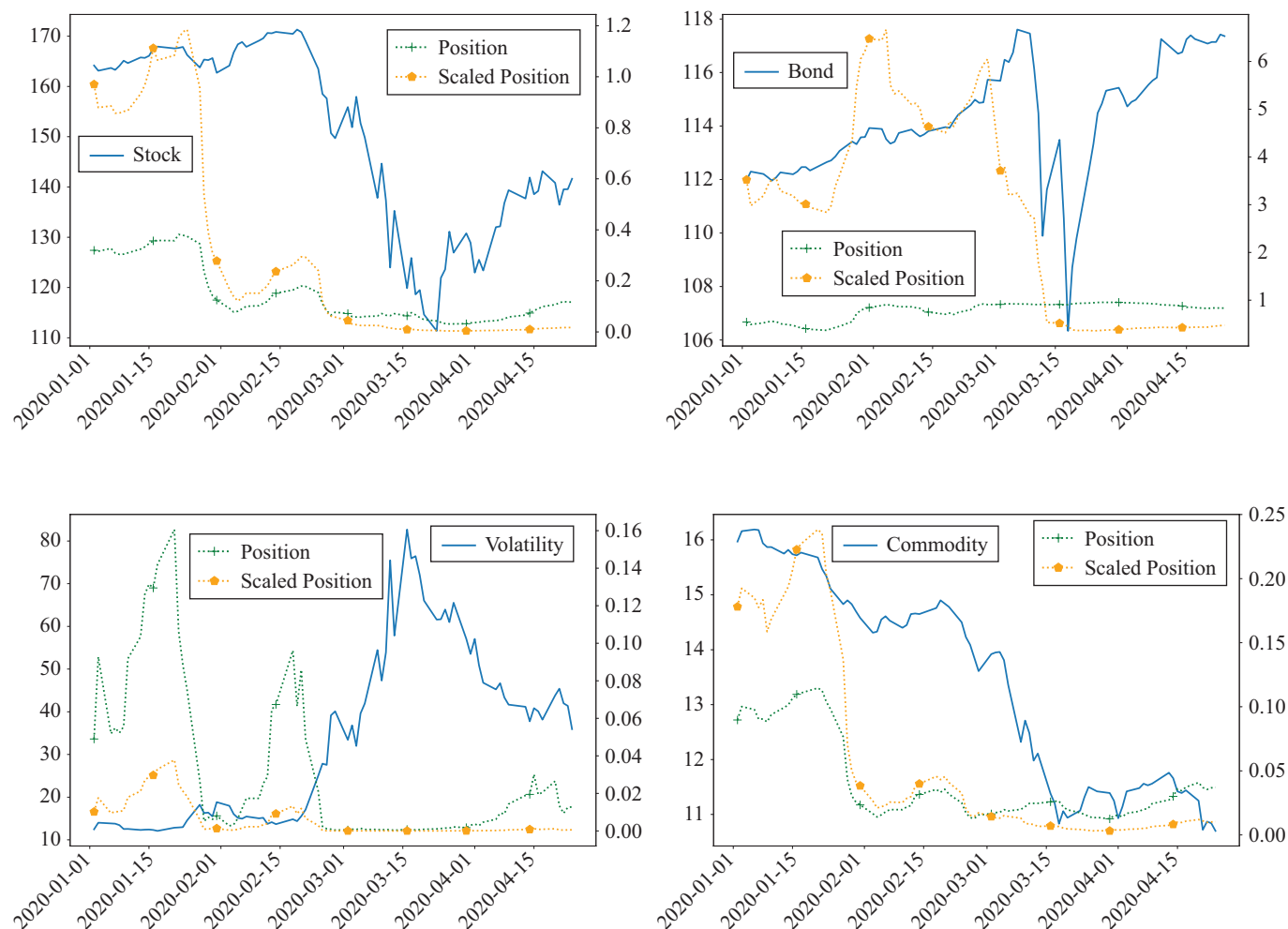
Boxplot for Annual Realized Trade Returns (top) and Annual Accumulated Costs for Different Assets with Volatility Scaling (bottom)



Note: $\sigma_{\text{tgt}} = 0.10$ and $C = 0.01\%$.

EXHIBIT 6

Shifts of Portfolio Weights for Our Model (DLS) during the COVID-19 Crisis with Volatility Scaling ($\sigma_{igt} = 0.10$)



contributions; as we can see, the bottom of each feature row has the greatest weight. This observation meets our understanding because, for time-series, recent observations carry more information. The further from the current observation point, the less importance features show, and we can adjust features used based on this observation (eg, using a small lookback window).

CONCLUSION

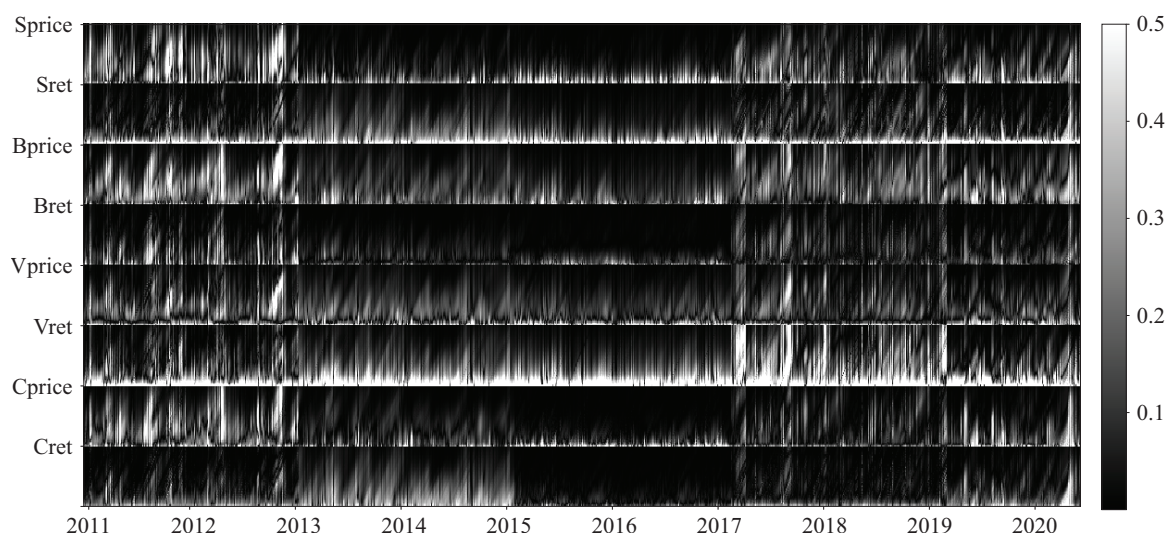
In this article, we adopt deep learning models to directly optimize a portfolio's Sharpe ratio. This pipeline bypasses the traditional forecasting step and allows us to optimize portfolio weights by updating model parameters through gradient ascent. Instead of using individual

assets, we focus on ETFs of market indexes to form a portfolio. Doing this substantially reduces the scope of possible assets from which to choose, and these indexes have shown robust correlations. In this article, four market indexes have been used to form a portfolio.

We compare our method with a wide range of popular algorithms, including reallocation strategies, classical MV, MD, and the SPT model. Our testing period is from 2011 to April 2020 and includes the recent crisis due to COVID-19. The results show that our model delivers the best performance, and a detailed study of our model performance during the crisis shows the rationality and practicability of our method. A sensitivity analysis is included to understand how input features contribute to outputs, and the observations meet our

EXHIBIT 7

Sensitivity Analysis for Input Features over Time



econometric understanding, showing the most recent features are most relevant.

In subsequent continuation of this work, we aim to study portfolio performance under different objective functions. Given the flexible framework of our approach, we can maximize the Sortino ratio or even the diversification degree of a portfolio as long as objective functions are differentiable. We further note that the volatility estimates used for scaling are lagged estimates that do not necessarily represent current market volatilities. We consider another extension to this work to thus adapt the network architecture to infer (future) volatility estimates as a part of the training process.

ACKNOWLEDGMENTS

The authors would like to thank members of Machine Learning Research Group at the University of Oxford for their useful comments. We are most grateful to the Oxford-Man Institute of Quantitative Finance for support and data access.

REFERENCES

Bertoluzzo, F., and M. Corazza. 2012. "Testing Different Reinforcement Learning Configurations for Financial Trading: Introduction and Applications." *Procedia Economics and Finance* 3: 68–77.

Chekhlov, A., S. Uryasev, and M. Zabarankin. 2005. "Draw-down Measure in Portfolio Optimization." *International Journal of Theoretical and Applied Finance* 8 (01): 13–58.

Choueifaty, Y., and Y. Coignard. 2008. "Toward Maximum Diversification." *The Journal of Portfolio Management* 35 (1): 40–51.

Cont, R., and D. Nitiens. 1999. "Statistical Properties of Financial Time Series." <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.42.5867>.

Fernholz, D., and I. Karatzas. 2010. "On Optimal Arbitrage." *The Annals of Applied Probability* 20 (4): 1179–1204.

———. 2011. "Optimal Arbitrage under Model Uncertainty." *The Annals of Applied Probability* 21 (6): 2191–2225.

Fernholz, E. R. "Stochastic Portfolio Theory." In *Stochastic Portfolio Theory*, pp. 1–24. New York: Springer, 2002.

Fernholz, R. "Portfolio Generating Functions." In *Quantitative Analysis in Financial Markets: Collected Papers of the New York University Mathematical Finance Seminar*, pp. 344–367. World Scientific, 1999.

Fernholz, R., and I. Karatzas. 2009. "Stochastic Portfolio Theory: An Overview." *Handbook of Numerical Analysis* 15: 89–167.

- Fernholz, R., I. Karatzas, and C. Kardaras. 2005. "Diversity and Relative Arbitrage in Equity Markets." *Finance and Stochastics* 9 (1): 1–27.
- Gastineau, G. L. "Exchange-Traded Funds." In *Handbook of Finance 1*, pp. 633–642. Hoboken: Wiley, 2008.
- Goodfellow, I., Y. Bengio, and A. Courville. *Deep Learning*. Cambridge, MA: MIT Press, 2016.
- Harvey, C. R., E. Hoyle, R. Korgaonkar, S. Rattray, M. Sargaison, and O. Van Hemert. 2018. "The Impact of Volatility Targeting." *The Journal of Portfolio Management* 45 (1): 14–33.
- Hochreiter, S., and J. Schmidhuber. 1997. "Long Short-Term Memory." *Neural Computation* 9 (8): 1735–1780.
- Huang, C. Y. 2018. "Financial Trading as a Game: A Deep Reinforcement Learning Approach." *arXiv* 1807.02787.
- Jensen, J. L. W. V. 1906. "Sur les fonctions convexes et les inégalités entre les valeurs moyennes." *Acta Mathematica* 30: 175–193.
- Kingma, D. P., and J. Ba. "Adam: A Method for Stochastic Optimization." *Proceedings of the International Conference on Learning Representations*, 2015.
- Krizhevsky, A., I. Sutskever, and G. E. Hinton. "Imagenet Classification with Deep Convolutional Neural Networks." In *Advances in Neural Information Processing Systems*, pp. 1097–1105. Cambridge, MA: MIT Press, 2012.
- LeCun, Y., Y. Bengio, and G. Hinton. 2015. "Deep Learning." *Nature* 521 (7553): 436–444.
- Lim, B., S. Zohren, and S. Roberts. 2019. "Enhancing Time-Series Momentum Strategies Using Deep Neural Networks." *The Journal of Financial Data Science* 1 (4): 19–38.
- Markowitz, H. 1952. "Portfolio Selection." *The Journal of Finance* 7 (1): 77–91.
- McNeil, A. J., R. Frey, and P. Embrechts. *Quantitative Risk Management: Concepts, Techniques and Tools—Revised Edition*. Princeton, NJ: Princeton University Press, 2015.
- Mnih, V., K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. "Playing Atari with Deep Reinforcement Learning." NIPS Deep Learning Workshop, 2013.
- Molina, G. 2016. "Stock Trading with Recurrent Reinforcement Learning (RRL)." *CS229*, nd Web 15.
- Moody, J., and M. Saffell. 2001. "Learning to Trade via Direct Reinforcement." *IEEE Transactions on Neural Networks* 12 (4): 875–889.
- Moody, J., L. Wu, Y. Liao, and M. Saffell. 1998. "Performance Functions and Reinforcement Learning for Trading Systems and Portfolios." *Journal of Forecasting* 17 (5–6): 441–470.
- Moskowitz, T. J., Y. H. Ooi, and L. H. Pedersen. 2012. "Time Series Momentum." *Journal of Financial Economics* 104 (2): 228–250.
- Ruf, J. 2013. "Hedging under Arbitrage." *Mathematical Finance: An International Journal of Mathematics, Statistics and Financial Economics* 23 (2): 297–317.
- Samo, Y.-L. K., and A. Vervuurt. "Stochastic Portfolio Theory: A Machine Learning Perspective." In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pp. 657–665. Arlington, VA: AUAI Press, 2016.
- Sharpe, W. F. 1994. "The Sharpe Ratio." *The Journal of Portfolio Management* 21 (1): 49–58.
- Sortino, F. A., and L. N. Price. 1994. "Performance Measurement in a Downside Risk Framework." *The Journal of Investing* 3 (3): 59–64.
- Sutton, R. S., and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 2018.
- Theron, L., and G. Van Vuuren. 2018. "The Maximum Diversification Investment Strategy: A Portfolio Performance Comparison." *Cogent Economics & Finance* 6 (1): 1427533.
- Tsantekidis, A., N. Passalis, A. Tefas, J. Kannianen, M. Gabbouj, and A. Iosifidis. "Using Deep Learning to Detect Price Change Indications in Financial Markets." In *2017 25th European Signal Processing Conference (EUSIPCO)*, pp. 2511–2515. New York: IEEE, 2017.
- Wild, R. *Index Investing for Dummies*. Hoboken: John Wiley & Sons, 2008.
- Williams, R. J. 1992. "Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning." *Machine Learning* 8 (3–4): 229–256.

Wong, T.-K. L. 2015. "Optimization of Relative Arbitrage." *Annals of Finance* 11 (3–4): 345–382.

Zhang, Z., S. Zohren, and S. Roberts. 2019a. "DeepLOB: Deep Convolutional Neural Networks for Limit Order Books." *IEEE Transactions on Signal Processing* 67 (11): 3001–3012.

———. 2019b. "Extending Deep Learning Models for Limit Order Books to Quantile Regression." *Proceedings of Time Series Workshop of the 36th International Conference on Machine Learning*, Long Beach, California, PMLR 97, 2019.

Zhang, Z., S. Zohren, and R. Stephen. 2020. "Deep Reinforcement Learning for Trading." *The Journal of Financial Data Science* 2 (2): 25–40.

Zivot, E. *Introduction to Computational Finance and Financial Econometrics*. Boca Raton: Chapman & Hall CRC, 2017.

To order reprints of this article, please contact David Rowe at d.rowe@pageantmedia.com or 646-891-2157.

ADDITIONAL READING

Enhancing Time-Series Momentum Strategies Using Deep Neural Networks

BRYAN LIM, STEFAN ZOHREN, AND STEPHEN ROBERTS

The Journal of Financial Data Science

<https://jfds.pm-research.com/content/1/4/19>

ABSTRACT: Although time-series momentum is a well-studied phenomenon in finance, common strategies require the explicit definition of both a trend estimator and a position sizing rule. In this article, the authors introduce deep momentum networks—a hybrid approach that injects deep learning-based trading rules into the volatility scaling

framework of time-series momentum. The model also simultaneously learns both trend estimation and position sizing in a data-driven manner, with networks directly trained by optimizing the Sharpe ratio of the signal. Backtesting on a portfolio of 88 continuous futures contracts, the authors demonstrate that the Sharpe-optimized long short-term memory improved traditional methods by more than two times in the absence of transactions costs and continued outperforming when considering transaction costs up to 2–3 bps. To account for more illiquid assets, the authors also propose a turnover regularization term that trains the network to factor in costs at run-time.

Deep Reinforcement Learning for Trading

ZIHAO ZHANG, STEFAN ZOHREN, AND STEPHEN ROBERTS

The Journal of Financial Data Science

<https://jfds.pm-research.com/content/2/2/25>

ABSTRACT: In this article, the authors adopt deep reinforcement learning algorithms to design trading strategies for continuous futures contracts. Both discrete and continuous action spaces are considered, and volatility scaling is incorporated to create reward functions that scale trade positions based on market volatility. They test their algorithms on 50 very liquid futures contracts from 2011 to 2019 and investigate how performance varies across different asset classes, including commodities, equity indexes, fixed income, and foreign exchange markets. They compare their algorithms against classical time-series momentum strategies and show that their method outperforms such baseline models, delivering positive profits despite heavy transaction costs. The experiments show that the proposed algorithms can follow large market trends without changing positions and can also scale down, or hold, through consolidation periods.