Objective Driven Portfolio Construction Using Reinforcement Learning

Tina Ruiwen Wang
tina_wang@vanguard.com
The Vanguard Group
Canada

Jithin Pradeep
jithin_pradeep@vanguard.com
The Vanguard Group
Canada

Jerry Zikun Chen
jerry_chen@vanguard.com
The Vanguard Group
Canada

ABSTRACT

Recent advancement in reinforcement learning has enabled robust data-driven direct optimization on the investor's objectives without estimating the stock movements as in the traditional two-step approach [8]. Given diverse investment styles, a single trading strategy cannot serve different investor objectives. We propose an objective function formulation to augment the direct optimization approach in AlphaPortfolio (Cong et al. [6]). In addition to simple baseline Sharpe ratio used in AlphaPortfolio, we add three investor's objectives for (i) achieving excess alpha by maximizing the information ratio; (ii) mitigating downside risks through optimizing maximum drawdown-adjusted return; and (iii) reducing transaction costs via restricting the turnover rate. We also introduce four new features: momentum, short-term reversal, drawdown, and maximum drawdown to the framework. Our objective function formulation allows for controlling the trade-off between both maximum drawdown and turnover with respect to realized return, creating flexible trading strategies for various risk appetites. The maximum drawdown efficient frontier curve, derived using a range of values of hyperparameter α, reflects the similar concave relationship as observed in the theoretical study by Chekhlov et al. [5]. To improve the interpretability of the deep neural network and drive insights into traditional factor investment, we further explore the drivers that contribute to the top and bottom performing firms by running regression analysis using Random Forest, which achieves R^2 of approximately 0.8 in producing the same winner scores as our model. Finally, to uncover the balance between profits and diversification, we investigate the impact of the trading size on strategy behaviors.

CCS CONCEPTS

• **Computing methodologies → Reinforcement learning**; **Neural networks**.

KEYWORDS

Portfolio Construction, Optimization, Maximum Drawdown, Asset Allocation

ACM Reference Format:
Tina Ruiwen Wang, Jithin Pradeep, and Jerry Zikun Chen. 2022. Objective Driven Portfolio Construction Using Reinforcement Learning. In *3rd ACM International Conference on AI in Finance (ICAIF '22), November 2–4, 2022, New York, NY, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3533271.3561764

1 INTRODUCTION

Portfolio construction has been a long-existing problem in finance, where there are debates on the objectives for investors with different financial needs [18]. In the conventional Modern Portfolio Theory [20], the question is formulated as a mean-variance optimization problem. In fact, high return is not the sole target, and volatility alone cannot fully reflect the risk exposure [1]. More often, investors tend to focus on other assessment metrics, such as relative performance, downside risks and trading frequencies. Due to the various standards in assessing trade strategies, a certain portfolio composition or strategy that serves the needs of all risk appetites cannot be easily quantified. Furthermore, due to the data complexity and massive scope of the company pool, automatic portfolio construction faces inevitable challenges of high dimensional noisy data [16]. Unlike supervised learning, which requires a labelled target, reinforcement learning (RL) offers the ability to directly optimize a reward signal through interacting with an environment. Hence, portfolio optimizations can be formulated as an RL problem where the reward indicates the financial goal. Recent advances in computing power have enabled the seamless integration of deep learning and RL for practical solutions, which expands the universe of solvable multi-period decision-making problems [29, 30].

We address the diverse risk preferences by extending an existing RL framework, AlphaPortfolio [6] with three objective functions and four additional features. AlphaPortfolio is a transformer-based architecture, which is capable of ingesting large-scale data. Instead of estimating the stock movements and determining the allocation accordingly [8], AlphaPortfolio uses a data-based direct optimization approach to construct large-scale portfolios using Deep Reinforcement Learning (DRL). The model consists of three parts: a Sequence Representation Extraction Module (SREM), a Cross-Asset Attention Network (CAAN), and a Portfolio Generator. The model directly produces action vectors that guide the investing activities, and Sharpe ratio (SR) is maximized. The framework is implemented and tested on zero-investment strategies (long-short). However, in the actual equity market, a margin requirement is usually imposed on long-short positions and the high yields are often grossly inflated and unattainable. In our experiments, we restrict to long-only holdings in all the strategies produced.

We further extend the problem scope by including other objectives, addressing varied risk tolerances and financial goals with

ICAIF '22, November 2–4, 2022, New York, NY, USA

ACM ISBN 978-1-4503-9376-8/22/11…$15.00
https://doi.org/10.1145/3533271.3561764

direct optimization using DRL. First, we introduce information ratio (IR) as the objective function to measure the ratio between excess gains and adjusted risks. IR uses relative returns over a benchmark index rather than nominal returns as in Sharpe ratio. Second, we optimize a linear combination of maximum drawdown (MDD) and Sharpe ratio where a hyper-parameter determines the emphasis on the two components. By altering the hyper-parameter, diverse risk characteristics can be obtained and it also addresses the problem of extreme downside deviations that is not reflected in volatility. Third, the same idea is applied to the turnover rate as a measure of how frequently stock allocation changes. High frequency trading yields larger profits, but it also comes at the cost of transaction activities. By adding the turnover to the objective, we can explicitly regularize trading frequency and transaction costs. Overall, the proposed framework allows investment styles and risk profiles to be adjusted by specifying alternative objective functions and tuning hyper-parameters.

The baseline Sharpe ratio maximization with four additional input features for long-only holdings generates out-of-sample Sharpe ratio over 1.2. The finding is comparable with reduced earnings and volatility when we adopt the information ratio benchmarked on S&P 500. With respect to MDD, the hyper-parameter α allows us to strike a balance between profits and downside risks. We plot an efficient frontier that displays a set of optimal portfolio compositions, which is consistent with findings from Chekhlov et al. [5]. Our proposed method shows an analogous concave positive relationship between return and turnover.

Latest studies focused on incorporating state-of-the-art machine learning techniques in asset allocation, inherits lack of interpretability as a consequence of the black-box mechanism of neural networks [11]. We use a projection procedure to interpret the DRL model output by mapping the allocation vectors to input features. Random Forest has strong statistical reliance with feature importance to offer interpretability by naturally identifying the attributes that contribute the most to the final output. This can be used to assist with factor modeling. Finally, we show that the degree of diversification is highly dependent on the number of stocks traded.

2 RELATED LITERATURE

2.1 Time Series Analysis Using Machine Learning

Historical stock prices and corporate financials have been the deterministic input for tasks like equity price forecasting and asset allocation. Quantitative finance studies examine these data with time series analysis [22]. Traditional time series analysis methods include Autoregressive Integrated Moving Average (ARIMA) [4], and Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models [10].

With rapid advancements in Machine Learning (ML), more research has utilized modern algorithms to predict time series [3]. Recurrent Neural Networks (RNN) - a specific category of Deep Neural Networks (DNN), and their variations like Long Short-Term Memory (LSTM) [7, 25] use internal states to maintain useful information across time, providing the networks with the ability to capture long-term temporal dependencies [14]. In 2017, Transformers, an architecture based on attention mechanisms was introduced that successfully overcomes the vanishing and exploding gradient problem in RNN and LSTM [27]. Since then, many studies employ Transformers in serial data analysis [28] including financial time series predictions [9].

2.2 Data-driven Direct Portfolio Optimization

Traditional portfolio construction involves predicting the returns of each asset and assigning weights using certain optimization rules [8]. However, given the highly stochastic nature of stock market, the errors made in prediction distorts the allocation decision. In the paper AlphaPortfolio by Cong et al. [6], a DRL method is introduced to replace the traditional two-step portfolio construction approaches. The goal is to maximize the out-of-sample Sharpe ratio through a data-driven direct optimization.

The AlphaPortfolio framework detects the auto-correlation across time for individual stock as well as the cross-correlation among stocks. As mentioned in Section 1, it consists of three modules: Sequence Representation Extraction Module (SREM), Cross-Asset Attention Network (CAAN), and Portfolio Generator. SREM feeds the multivariate time series of each stock to a transformer encoder with shared universal parameters. It maps the sequential input into a high-dimensional latent space and retrieves the underlying embeddings. Next, CAAN module applies cross-attention on SREM embeddings to capture the interactions between different stocks. CAAN generates a scalar winner score for each stock. Finally, the Portfolio Generator applies a softmax operator to obtain the allocation proportions that are within $[0, 1]$ and $[-1, 0]$ for long and short positions, respectively. The return for the following month is calculated using the allocation proportion vector and each stock's return.

More specifically, with rewards as monthly return the formulation could be written as: $\max_\theta SharpeRatio\{reward_1, ..., reward_t\}$. The entire architecture acts as a policy network that produces actions, which are one-dimensional vectors representing equity allocation proportions. Model parameters θ are updated by backpropagating the gradients ($\nabla_\theta SharpeRatio$) to optimize the objective function.

3 METHODOLOGY

We propose an objective-driven portfolio construction framework, incorporating three novel objective functions building upon AlphaPortfolio [6] to generate flexible financial trading strategies. The first objective incorporates information ratio benchmarked on S&P 500 Index returns, which is often used as a measure of the portfolio manager's ability to generate excess alpha compared to passively investing in the market index. Second, we incorporate maximum drawdown as part of the objective. In financial studies, portfolio managers and investors care about MDD but usually only use it as an assessment metric [13] or view MDD as a constraint [12] rather than explicitly optimizing it. By adjusting the hyper-parameter in the objective function, level of uncertainty could be altered for different risk appetites. The third objective we consider is the frequency of transactions and associated cost. Deep reinforcement learning models absorb transaction costs in the trading environment, i.e., transaction costs are deducted from profits each time a trading action is performed, which usually requires prior knowledge. Alternatively, the turnover rate is treated as an adjusting factor in the objective. As a result, the relationship between optimal accumulative return and rate of turnover is easy to envision - a positive association is expected, with greater earnings at the cost of higher trading frequency and a more active strategy.

Additionally, supplementary attributes typically useful in traditional factor modelling are added to the input space. The features include momentum, short-term reversal, stock-level drawdown, and stock-level maximum drawdown.

Algorithm 1 Objective-Driven Portfolio (ODP)

```
Require: Trading environment $\mathcal{E}$ with historical data. Learning rate: $lr$. Objective function $J$ of choice: Information Ratio, Maximum Drawdown, and Turnover. If the objective function is MDD or Turnover, hyper-parameter $\alpha$.
Initialize model weights $\theta$ that is used for policy (actor) model $\mu(s, \theta)$.
for episode = 1 to N do
    Acquire the initial state $s_1$ from the environment $\mathcal{E}$.
    for t = 1 to T do
        Select action $b_t$ from $b_t = \pi(s_t)$.
        Interact with the environment $\mathcal{E}$ (with action $b_t$) and get the new observation $s_{t+1}$ and reward $r_t$.
    end for
    Update actor policy by policy gradient:
    if Objective function is Information Ratio then
        $\nabla_\theta J = \nabla_\theta IR(\{r_1, ..., r_T\}) = \frac{\text{mean}(\{r_1,...,r_T\})}{\text{std}(\{r_1,...,r_T\})}$
    else if Objective function is Maximum Drawdown then
        $\nabla_\theta J = \nabla_\theta (\alpha SR(\{r_1, ..., r_T\}) + (1 - \alpha) MDD(\{r_1, ..., r_T\}))$
    else if Objective function is Turnover then
        $\nabla_\theta J = \nabla_\theta (\alpha SR(\{r_1, ..., r_T\}) - (1 - \alpha) Turnover(\{r_1, ..., r_T\}))$
    end if
end for
```

3.1 Data

Data is primarily sourced from the Wharton Research Data Services (WRDS) [1] database, and for the paper we used the following 4 datasets: CRSP Stock File, CRSP Compustat Merged, CRSP Beta Suite, and Financial Ratios (Firm Level data). CRSP Stock contains the Open-High-Low-Close-Volume (OHLCV) information. CRSP Compustat Merged and Financial Ratios datasets include income statements and balance sheets fundamentals, with insightful financial indicators and ratios. CRSP Beta Suite enables the inclusion of stocks' weights and coefficients on various risk factors in conventional factor models, for example, CAPM, or Fama-French 4 Factor Model.

3.1.1 *Data Processing.*

Stock prices and financial data have different release frequencies. Stock price data are available daily, and financial data are obtained annually or quarterly. Daily data is summarized and aggregated, while quarterly and annual data is forward filled in accordance with monthly periodicity. Forward filling is a technique to fill missing future values with previous values without interpolation until a new value is seen. We apply z-score normalization using the training period distribution such that training data is a standard normal distribution. Due to the presence of extreme outliers and distorted distribution, min-max normalization is not adopted here.

For the paper, we have imposed the following restrictions on the data to be included in the final dataset. First, the stock is in the scope of research if stock's record exists in all four listed datasets. Second, to mitigate bias caused by forward filling, stocks that survive more than two years during training phase and at least one year during testing phase are included. Third, extreme values outside 3 standard deviations are removed because they significantly distort the performance of the model. After exclusion, about 1.8 million records remain in the dataset which corresponds to 14 thousand unique stocks during the period of 1971 to 2020.

3.2 Environment

The reinforcement learning environment is established to simulate a trading system. At a specific time, the available knowledge to the agent is the historical company data over the past 12 months, known as the states.

The action $\boldsymbol{b_t}$ is the output of the neural network model described in Section 2.2, denoting investment distribution. Given a notional amount (budget constraint) of $\$K$, the amount allocated in each stock is $K\boldsymbol{b_t}$, and the number of shares would be $K\boldsymbol{b_t}/\boldsymbol{p_{t-1}}$, where $\boldsymbol{p_{t-1}}$ is a vector that records the last month's stock prices. Overall, monthly return, which is the stepwise reward, can be calculated as the dot product between action $\boldsymbol{b_t}$ and the vector representing gain for individual tickers $\boldsymbol{ret_t}$:

$$r_t = \sum_i \frac{Kb_t^{(i)}}{p_{t-1}^{(i)}K}(p_t^{(i)} - p_{t-1}^{(i)}) = \sum_i b_t^{(i)} \frac{(p_t^{(i)} - p_{t-1}^{(i)})}{p_{t-1}^{(i)}} \tag{1}$$

$$= \sum_i b_t^{(i)} ret_t^{(i)} = \boldsymbol{b_t} \cdot \boldsymbol{ret_t} \tag{2}$$

3.3 Extensions

Our major enhancements include the implementation of three innovative objective functions that address various financial aspects in risk and return and supplementary input variables.

3.3.1 *Information Ratio (IR).*

Information ratio and Sharpe ratio (SR) are both calculated by dividing the mean by the standard deviation of returns. The distinction is that SR uses nominal gains and IR takes surplus benchmarked against a particular index. Besides, the denominator consists of the variance of excess returns, namely tracking errors. In our experiment, S&P 500 Index is used as the benchmark:

$$\text{IR} = \frac{\text{mean}(r - r^{S\&P500})}{\text{std}(r - r^{S\&P500})} = \frac{\text{mean}(\{r_t - r_t^{S\&P500}\}_{t=1,...,12})}{\text{std}(\{r_t - r_t^{S\&P500}\}_{t=1,...,12})} \tag{3}$$

where r and $r^{S\&P500}$ denote the return vectors of the portfolio and S&P 500 Index respectively. In comparison,

$$\text{SR} = \frac{\text{mean}(\{r_t\}_{t=1,...,12})}{\text{std}(\{r_t\}_{t=1,...,12})} = \frac{\text{mean}(\{r_1, ..., r_{12}\})}{\text{std}(\{r_1, ..., r_{12}\})} \tag{4}$$

[1] Wharton Research Data Services (WRDS) was used in preparing Objective-Driven Portfolio Construction Using Reinforcement Learning. This service and the data available thereon constitute valuable intellectual property and trade secrets of WRDS and/or its third-party suppliers.

Modification to the environment is that the reward is replaced from r_t to $r_t - r_t^{S\&P500}$, which are the excess returns. In optimization stage, $loss = -\text{IR}$.

3.3.2 Maximum Drawdown (MDD). Maximum drawdown is the largest percentage drop detected within a specified time window - which we set to be 12 months, before the observation point t. MDD refers to the magnitude of the largest loss. Because of liquidity and reserve requirement, MDD is considered more often than volatility. The mathematical formula is demonstrated below:

$$\text{MDD}_t = \frac{trough - peak}{peak} = \min_{j=t-11,\ldots,t} \frac{A_j - \max\limits_{m=t-11,\ldots,j} A_m}{\max\limits_{m=t-11,\ldots,j} p_m} \tag{5}$$

$$= \min_{j=t-11,\ldots,t} \frac{\prod_{k=1}^{j}(1+r_k) - \max\limits_{m=t-11,\ldots,j} \prod_{k=1}^{m}(1+r_k)}{\max\limits_{m=t-11,\ldots,j} \prod_{k=1}^{m}(1+r_k)} \tag{6}$$

where A_j and A_m indicate the accumulative wealth at time j and m. MDD is a negative value with a maximum of 0. MDD reaches 0 only when there have been no losses incurred to the portfolio over the past 12 months.

With only risk-related metrics, the model cannot generate any valuable income strategy. A complete objective function should take into account both a downside risk component, and an earnings factor. Therefore, we add the Sharpe ratio to the loss not only because SR accounts for volatility as a disparate aspect of uncertainty, and achieves better experimental results. A linear combination between MDD and SR is used to indicate the degree of contribution from both factors.

$$loss = -(\alpha \times \text{SR} + (1-\alpha) \times \text{MDD}) \tag{7}$$

where α is a hyper-parameter and MDD and SR are both derived using monthly portfolio returns. By minimizing the loss, we can construct a portfolio with high SR and limited drawdown.

3.3.3 Turnover Rate. Turnover rate refers to the percentage adjustment in asset allocations in a portfolio at each rebalancing time, which is often taken as a reporting metric rather than an optimization goal. The definition is as below, the factor of 1/2 is to avoid double counting.

$$\text{Turnover}_t = \frac{1}{2}\sum_i |b_{t-1}^{(i)}(1 + ret_t^{(i)}) - b_t^{(i)}| \tag{8}$$

The exact calculation of turnover rate necessitates a massive amount of memory due to the large vector $\boldsymbol{ret_t}$. During the learning process, the expression is simplified to $\frac{1}{2}\sum_i |b_{t-1}^{(i)} - b_t^{(i)}|$. The final reported turnover rate uses the full formula. Similar to MDD, the loss function consists of a linear combination of turnover and SR. The negative sign is added for the turnover part in Equation 9 due to positive turnovers.

$$loss = -(\alpha \times \text{SR} - (1-\alpha) \times \text{Turnover}) \tag{9}$$

3.3.4 Four Additional Features. Four additional attributes - momentum, short-term reversal, drawdown and maximum drawdown, computed from portfolio return are appended to the feature space. Momentum and short-term reversal are financial terms that refer to the ability for a stock price trend to remain constant or revert in the short term respectively. They have been proven to contribute predictive power in portfolio construction problems [17]. Momentum is represented by the return of the stock over the past year and short-term reversal takes the coefficients of up-minus-down (UMD) in Fama-French 4 Factor model, which is regressed on monthly premium of winners over losers in the market. Last but not least, stock-level drawdown and maximum drawdown over one year period are added as additional independent variables.

4 EXPERIMENT PERFORMANCE AND RESULTS

Computation of cross-correlation relies on self-attention, which has a quadratic space complexity with respect to the total number of stocks. Hence, a subsampling is performed such that 2k companies out of the 14k pool are randomly chosen as the new stock universe. Also, to align with the baseline paper [6], the training phase spans from 1971 to 1989. Records beyond 1990 are used for testing. In training, the sample month is randomly selected without replacement throughout the timeframe, the learning is repeated until all months have been exploited, which is called one epoch. Overall, we complete 30 epochs with decaying learning rates. For testing, data are fed in through a chronological order, and out-of-sample (OOS) results are recorded. To maintain model timeliness and curb the impacts of market regime shifts, the model is recalibrated after every 12 steps during testing, that is, the model is fine-tuned every 12 months. The unbiased OOS results are reported using unseen data.

Sharpe ratio, information ratio, maximum drawdown and turnover are four measures that are evaluated to examine the resilience of the network architecture and the variety of portfolios guided by distinct criteria. The SR and IR experiments are designed to produce portfolios that maximize the return-to-risk ratio. In further analysis, we discover the volatility of the output strategy can be controlled by adjusting the hyper-parameter α in MDD objective function. Lastly, the experiments on turnover rate reveal the trade-off between return and trading frequency.

4.1 Sharpe Ratio

We directly evaluate the efficacy of the four added variables mentioned in Section 3.3.4 and compare the model performance to that of Cong et al. [6]. Table 1 presents the outcome of the long-short experiment with Sharpe ratio as the maximization objective. The setting with additional features yields a higher earning but also a bigger risk, and achieves an overall Sharpe ratio of 2.76.

Due to practical restrictions on short-selling activities and overinflated gains using long-short approach, all the other experiments focus on long-only solutions. In other words, the action vector $\boldsymbol{b} = \boldsymbol{b}^+$ with all the entries positive and adding up to 1.

Tabel 2 reports the OOS outcomes with Sharpe ratio as the fundamental objective and short selling banned. The realized return easily surpasses the benchmark of S&P 500 (SR of S&P 500 is 0.54), and the model reaches a Sharpe ratio of 1.2. This improved performance can also be viewed in the graph of accumulative wealth in Figure 1. The strategy, however, has a high level of volatility and the magnitude of drawdown is 33% during the Great Recession. In addition, the actions in testing phase are recorded and grouped by

Table 1: Sharpe Ratio Long-Short Performance

Metric	AlphaPortfolio	Objective-driven model
Annualized Return	17.00%	43.33%
Average Volatility	8.48%	13.55%
Sharpe Ratio	2.00	2.76
Skewness	1.42	0.36
Kurtosis	6.35	1.62
Mean Turnover	0.26	0.51
Max Drawdown	-8%	-25%

Table 2: SR performance

Average Monthly Return	1.87%
Annualized Return	21.61%
Average Volatility	18.64%
Sharpe Ratio	1.20
Skewness	-0.14
Kurtosis	0.68
Mean Turnover	0.21
Max Drawdown	-33.06%

industry as illustrated in Figure 2. The industrial categorization is based on the 10 SIC codes. We noticed that the distribution is relatively steady over time, implying that the strategy is to invest in individual stocks rather than industrial sector trends.

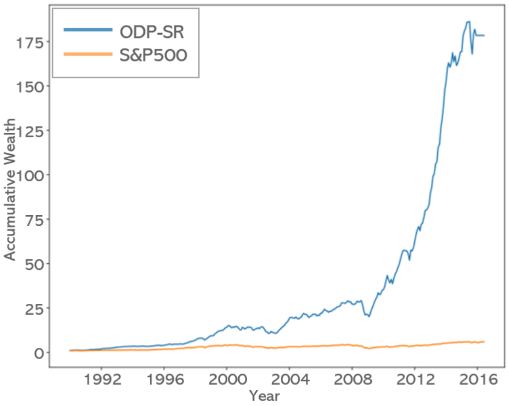


Figure 1: Sharpe Ratio - Accumulative Wealth

4.2 Information Ratio

Although the information ratio serves as a similar purpose as the Sharpe ratio, the capital curve has different shapes and behaviours during the test period, as shown in Figure 3. Moreover, Table 3 presents that the trading policy induces slightly lower SR as a result of reduced return and volatility than in the baseline experiment as well as failure to avoid the extreme losses with MDD of −33%. It is a less attractive choice for risk-seeking investors.

The created portfolio is more inclined to track and follow the benchmark index (S&P 500) as doing so will largely scale down the tracking error variance and the denominator. Consequently, the trading behaviour is closer to that of S&P 500 while maximizing risk-adjusted returns. More analysis should be conducted in future studies to analyze the impact of benchmark selection on IR models.

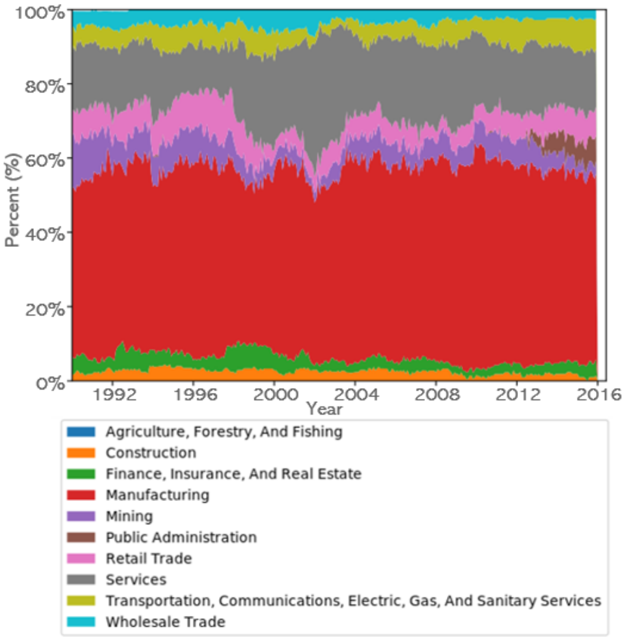


Figure 2: Sharpe Ratio - Action

Table 3: IR performance

Average Monthly Return	1.35%
Annualized Return	16.36%
Average Volatility	13.76%
Sharpe Ratio	1.18
Skewness	0.07
Kurtosis	1.41
Mean Turnover	0.2
Max Drawdown	-33.13%

4.3 Maximum Drawdown

In order to demonstrate the model's adaptability in delivering trading guidance for varying risk appetites, a range of hyper-parameters α from 0% to 100% are tested to highlight the relationship between annualized return and MDD. Table 4 covers the partial list of trials and Figure 4 depicts all experiment findings, where MDDs are displayed in absolute values.

The concave shape of the curve shows that a heavier emphasis on SR yields better earnings but also a more severe loss and drastic maximum drawdown, and vice versa. The magnitude of α implies the degree of emphasis on Sharpe ratio compared to MDD. Classical finance research has also verified this efficient frontier with mathematical programming as shown in Chekhlov et al. [5]. This

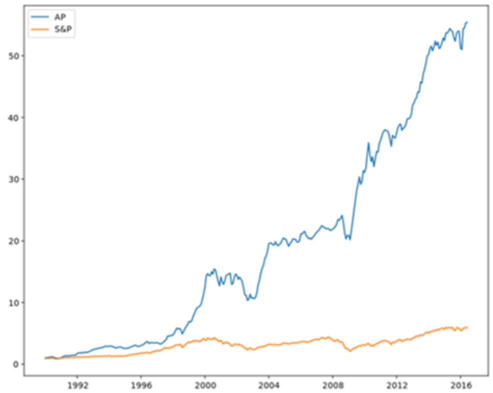


Figure 3: Information Ratio - Accumulative Wealth

Table 4: MDD Performance

α	Absolute MDD	Annualized Return
10%	15.33%	10.98%
30%	17.65%	16.53%
50%	23.61%	17.70%
70%	28.81%	20.44%
90%	30.82%	19.52%
100%	33.06%	21.61%

suggests that our data-based result is consistent with financial theory. Although actual points do not form a perfect curve, the general trend follows the shape of the frontier. In conclusion, the direct optimization setup can be used to adapt strategies by changing the α value.

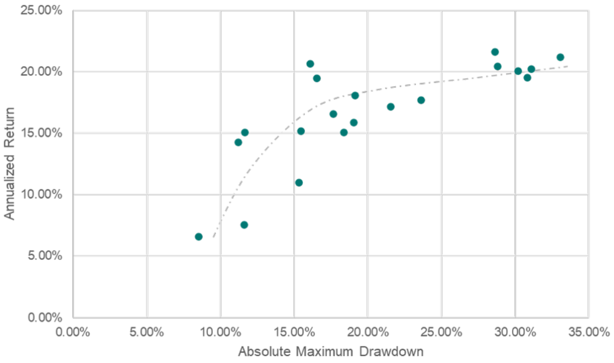


Figure 4: Maximum Drawdown

4.4 Turnover Rate

Similarly, we run multiple trials on the hyper-parameter α to adjust turnover rate as a regularization factor. As expected, we find the concave curve that exhibits the set of efficient portfolios of the return-turnover spectrum in Figure 5. Sample points are laid out in Table 5. The low α range allows us to receive a substantial gain in profits in exchange for sacrifice in trading frequencies; but, as the mean turnover rate increases, the benefit diminishes. For each fixed α, other hyper-parameters such as learning rates are tuned to obtain the most effective model. The same conclusion is drawn that by reducing the magnitude of α to control SR, the strategy becomes more passive, with a lower reward.

Table 5: Turnover Performance

α	Turnover Rate	Annualized Return
10%	0.1390	15.84%
30%	0.1428	13.64%
50%	0.1601	18.63%
70%	0.1989	16.94%
90%	0.1994	19.27%
100%	0.2174	21.61%

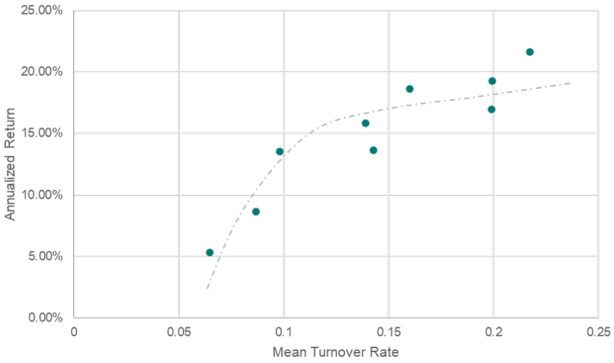


Figure 5: Turnover Rate

4.5 Interpretations

We regress on the winner scores and mimic the outcome of the neural network using several fundamental models (Lasso Regression, Elastic Net, and Random Forest) for model interpretation. Because these simple projection models cannot directly handle serial data, instead of feeding them into transformer architecture, each sequential feature is split into 12 parts that represent the lagging effect over the past year along with a quadratic term to capture any nonlinear trend. In all, there are $55 \times 12 \times 2 = 1320$ features used in the interpretation task as the independent variables.

Lasso regression and Elastic Net are statistical variations of linear regression that perform regularization and feature selection. Absolute shrinkage (L1-Norm) is adopted in Lasso and a linear combination of absolute and quadratic shrinkage (L1 and L2 penalties) is utilized in Elastic Net. Random Forest is a non-linear ensemble method that computes the average outcome of independent decision trees. Both linear techniques (Lasso, Elastic Net) extract the level of significance of driving factors by examining the p-values, while the random forest regressor considers feature importance based on weighted node impurity. There is strong multicollinearity

Table 6: Interpretation Regression R^2

method	Top R^2	Bottom R^2
Lasso Regression	0.12	0.01
Elastic Net	0.17	0.01
Random Forest	0.78	0.82
Linear Regression w/ top 50 features selected by RF	0.18	0.20

Table 7: Interpretation Top Features

Top	Bottom
TotoalAsset_11	BookToMarket_2
TotoalAsset_2	Beta_1
TotoalAsset_12	Spread_5
Turnover_7	Return_3
Spread_2^2	Return_3^2
Spread_2	MarketCap_9
Idol_vol_12	delta_BookValue_12^2
Beta_1	delta_PI2A_11
Spread_4^2	Beta_3^2
Drawdown_1^2	Beta_7

among the 12 lagging components created, which causes a logical concern in linear regression analysis. To resolve this, in Lasso and Elastic Net approaches, if a pair of independent variables has a correlation over 0.9, one of them is removed to help ensure the matrix is not singular.

The response variable is the testing winner scores generated from CAAN ranging from -1 to 1 as described in Section 2.2. A high winner score close to 1 signals that the stock is assigned a large fraction in the portfolio, whilst a low winner score implies a small weight or none-selection (not traded) in the next month’s trading decision.

To assess the influential attributes contributing to the distinct behaviours of leading and unsatisfactory performers, the complete collection is truncated into top ($1 >$ winner score ≥ 0) and bottom set ($-1 <$ winner score < 0) with separate models trained on them. The R^2’s are reported in Table 6 below.

The R^2 statistic measures how well the explanatory characteristics address the variation in the dependent variable. Random Forest justifies substantially more variance than the linear mechanisms, whereas Lasso and Elastic Net fail significantly for the loser half, yielding an R^2 of 0.01. We run simple linear regression with the top 50 characteristics picked by Random Forest, to further validate its potency, which delivers higher R^2 than Lasso Regression and Elastic Net. This proves that the non-linear variable ranking using Random Forest is superior to feature elimination with regularization. Therefore, we deploy Random Forest to unfold the DRL’s behaviour since it reliably simulates the response of the neural network. The inadequacy of linear methods indirectly reflects the depth and complexity of the DRL system.

Table 7 shows the top 10 important features uncovered by Random Forest for the top and bottom stocks respectively. The suffix denotes the time lag; for example, _n is the value of the variable from $12 - n$ months ago, and the ^2 marks the quadratic term. The divergence between the groups of chosen drivers is obvious for the superior and inferior scores. Total asset, turnover, stock spread, idiosyncratic volatility, beta, and drawdown stand out for the leading records and play a critical role in deciding which stocks should appear in the portfolio. This entails that equity and market related indicators as well as the company capitalization are crucial in detecting profitable prospects. On the other hand, book-to-market, beta, spread, monthly gain, market capitalization, and change in property are valuable in spotting red flags and filtering out unhealthy businesses, which suggests that bad business operations are often reflected by fundamentals in annual reports.

4.6 S&P 500 Stock Sharpe Ratio Experiments

Apart from the experiments conducted on all equities, we trade exclusively on S&P 500 tickers and run lateral comparisons to align our trading universe with the benchmark. The AI strategy’s profitability can be realized and examined precisely. Due to the dynamic nature of S&P 500, we collect recent data from 1993 to 2020 and mark the first half (1993-2006) as training and 2007-2020 as test sample. In the standard trial, 10% of the pool, which is 50 stocks are traded at each time, and Sharpe ratio is the objective function. The accumulative wealth is plotted in Figure 6.

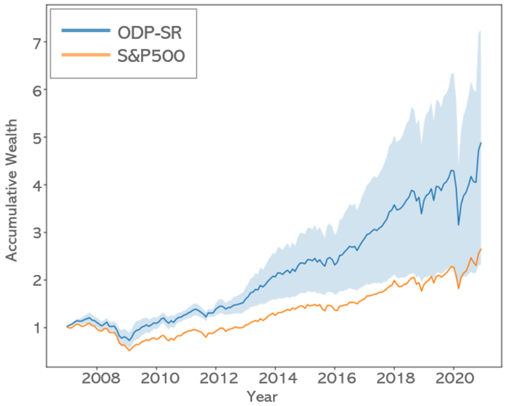
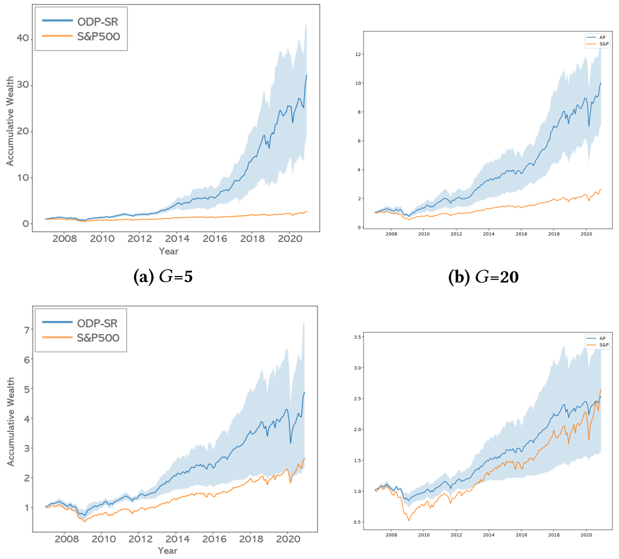


Figure 6: S&P 500, Sharpe Ratio

Due to a more restrictive trading space, we are able to perform parallel computing and construct a confidence interval. In the early phase of the testing period, the algorithm constantly outperforms S&P 500 Index; however, in later years, the lower bound is close to the index. On average, AI strategy exceeds long-and-hold, implying that the model is able to identify and allocate money to rewarding stocks even with the same pool of instruments. Over the last few years, applications and studies focusing on trading strategy creation have surged, particularly with the integration of advanced machine learning algorithms [19, 26]. This has intensified the competition in the zero-sum trading game, which has made capturing and extracting the profits harder. Furthermore, the model is mainly trained on data from 2007, and the stock market data distribution has shifted drastically since then. Hence, the model adapts better to training



(a) G**=5**

(b) G**=20**

(c) G**=50**

(d) G**=100**

Figure 7: Accumulative Wealth with different G

set than recent data. In conclusion, the shrinkage in profit in the later part of the horizon is due to both extended testing window as well as increasing market competitions.

4.7 Effect of Number of Stocks Traded

We test the effect of changing the number of stocks to trade G on the model performance. The value of G denotes the number of non-zero entries of the action vector as an output of Portfolio Generator. The comparison result is shown in Figure 7. With an increase in G, the model generates a lower level of wealth, which can be attributed to the trade-off between profitability and diversification. With smaller G, the model concentrates the allocation on a few top-performing stocks; with larger G, the portfolio distributes the capital on a more diverse collection of equities. Diversification cannot ensure a profit or protect against loss in a declining market. It is a strategy used to help mitigate risk.

5 DISCUSSION AND CONCLUSION

In this paper, we demonstrate that the direct policy optimization approach is flexible and effective in tackling portfolio construction problems and extend the reinforcement learning framework with multiple quantitative objective functions to include information ratio, maximum drawdown, and turnover rate. The stock market possesses many properties of a Markov Decision Process (MDP); hence, the RL policy successfully learns with historical data and adapts to unseen situations.

The experiments have depicted an apparent trade-off between realized gains and downside risks. Also, there is a positive but diminishing correlation between earnings and trading frequencies. The efficient frontiers demonstrate the feasibility to create strategies with specific risk appetites and financial constraints. It also expands on the possibility that data-driven direct optimization using RL could be explored further on other financial objectives with objective function formulation.

Deep neural networks are subject to critiques of inadequacy of human-friendly explanations and absence of closed-form formula. As an attempt to identify the underlying rationale of the complex model, we take the original responses of the neural network to regress on the restructured features. Random Forest, a nonlinear technique, achieves R^2 of roughly 0.8 in producing the same winner scores as our model. Additionally, separate driving factors are found significant for high and low ranking stocks as shown in Table 7. Lastly, we show the trade-off between yields and diversification through changing number of stocks in the portfolio.

The current direct optimization approach uses deterministic policy search as the RL update mechanism. Further improvements in terms of RL framework include the application of more advanced and stable algorithms such as PPO [24], A3C [21], etc. Distributional RL [2] can be implemented to improve model stability. Risk-aware RL [15] and optimization within a Wasserstein Ball [23] could also be adopted to enhance the robustness against model errors by incorporating an economically sound performance criterion.

ACKNOWLEDGMENTS

We are grateful to Haoran Wang, Komal Teru, and Prof. Sebastian Jaimungal for excellent research assistance, and all reviewers for helpful comments.

This material is provided for informational purposes only and is not intended to be investment advice or a recommendation to take any particular investment action. All investing is subject to risk, including the possible loss of the money you invest.

REFERENCES

[1] Andrew Ang, Joseph Chen, and Yuhang Xing. 2006. Downside Risk. *The Review of Financial Studies* 19, 4 (2006), 1191–1239. http://www.jstor.org/stable/4123472

[2] Marc G. Bellemare, Will Dabney, and Rémi Munos. 2017. A Distributional Perspective on Reinforcement Learning. https://doi.org/10.48550/ARXIV.1707.06887

[3] Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Aël Le Borgne. 2013. *Machine Learning Strategies for Time Series Forecasting*. Springer Berlin Heidelberg, Berlin, Heidelberg, 62–77. https://doi.org/10.1007/978-3-642-36318-4_3

[4] G. E. P. Box and G. M. Jenkins. 1968. Some Recent Advances in Forecasting and Control. *Journal of the Royal Statistical Society Series C* 17, 2 (June 1968), 91–109. https://doi.org/10.2307/2985674

[5] Alexei Chekhlov, Stanislav Uryasev, and Michael Zabarankin. 2003. Portfolio Optimization with Drawdown Constraints.

[6] L. Cong, Ke Tang, Jingyuan Wang, and Y. Zhang. 2020. AlphaPortfolio: Direct Construction Through Reinforcement Learning and Interpretable AI. *Capital Markets: Asset Pricing & Valuation eJournal* (2020).

[7] J.T. Connor, R.D. Martin, and L.E. Atlas. 1994. Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks* 5, 2 (1994), 240–254. https://doi.org/10.1109/72.279188

[8] Victor DeMiguel, Lorenzo Garlappi, and Raman Uppal. 2009. Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy? *The review of Financial studies* 22, 5 (2009), 1915–1953.

[9] Qianggang Ding, Sifan Wu, Hao Sun, Jiadong Guo, and Jian Guo. 2020. Hierarchical Multi-Scale Gaussian Transformer for Stock Movement Prediction.. In *IJCAI*. 4640–4646.

[10] Robert Engle. 2001. GARCH 101: The Use of ARCH/GARCH Models in Applied Econometrics. *Journal of Economic Perspectives* 15, 4 (Fall 2001), 157–168. https://ideas.repec.org/a/aea/jecper/v15y2001i4p157-168.html

[11] Fenglei Fan, Jinjun Xiong, Mengzhou Li, and Ge Wang. 2020. On Interpretability of Artificial Neural Networks: A Survey. https://doi.org/10.48550/ARXIV.2001.02522

[12] Carmine De Franco, Johann Nicolle, and Huyên Pham. 2020. Discrete-time portfolio optimization under maximum drawdown constraint with partial information and deep learning resolution. arXiv:2010.15779 [cs.CL]

[13] Afan Hasan, Oya Kalıpsız, and Selim Akyokus. 2020. Modeling Traders' Behavior with Deep Learning and Machine Learning Methods: Evidence from BIST 100 Index. *Complexity* (2020), 1–21. https://doi.org/10.1155/2020/8285149

[14] Yuxiu Hua, Zhifeng Zhao, Rongpeng Li, Xianfu Chen, Zhiming Liu, and Honggang Zhang. 2019. Deep Learning with Long Short-Term Memory for Time Series Prediction. *IEEE Communications Magazine* 57, 6 (2019), 114–119. https://doi.org/10.1109/MCOM.2019.1800155

[15] Sebastian Jaimungal, Silvana Pesenti, Ye Sheng Wang, and Hariom Tatsat. 2021. Robust Risk-Aware Reinforcement Learning. *SIAM J. Financial Mathematics, Forthcoming. Available at https://arxiv.org/abs/2108.10403* (2021).

[16] N.J. Jobst, M.D. Horniman, C.A. Lucas, and G. Mitra. 2001. Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints. *Quantitative Finance* 1, 5 (may 2001), 489–501. https://doi.org/10.1088/1469-7688/1/5/301

[17] Bryan T. Kelly, Tobias J. Moskowitz, and Seth Pruitt. 2021. Understanding momentum and reversal. *Journal of Financial Economics* 140, 3 (2021), 726–743. https://doi.org/10.1016/j.jfineco.2020.06.024

[18] Tsong-Yue Lai. 1991. Portfolio selection with skewness: a multiple-objective approach. *Review of Quantitative Finance and Accounting* 1, 3 (1991), 293–305.

[19] Yilin Ma, Ruizhu Han, and Weizhong Wang. 2021. Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems with Applications* 165 (2021), 113973. https://doi.org/10.1016/j.eswa.2020.113973

[20] Harry Markowitz. 1952. Portfolio Selection. *The Journal of Finance* 7, 1 (March 1952), 77–91. https://doi.org/10.2307/2975974

[21] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous Methods for Deep Reinforcement Learning. arXiv:1602.01783 [cs.LG]

[22] Andrew J Patton. 2009. Copula–based models for financial time series. In *Handbook of financial time series*. Springer, 767–785.

[23] Silvana Pesenti and Sebastian Jaimungal. 2020. Portfolio Optimisation within a Wasserstein Ball. *Available at https://arxiv.org/abs/2012.04500* (2020).

[24] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347 [cs.LG]

[25] Ralf C. Staudemeyer and Eric Rothstein Morris. 2019. Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks. *arXiv preprint* arXiv:1909.09586 (2019).

[26] Thibaut Théate and Damien Ernst. 2021. An application of deep reinforcement learning to algorithmic trading. *Expert Systems with Applications* 173 (jul 2021), 114632. https://doi.org/10.1016/j.eswa.2021.114632

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *arXiv preprint* arXiv:1706.03762 (2017).

[28] Neo Wu, Bradley Green, Xue Ben, and Shawn O'Banion. 2020. Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case. arXiv:2001.08317 [cs.LG]

[29] Hongyang Yang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid. 2020. Deep reinforcement learning for automated stock trading: An ensemble strategy. *Available at SSRN* (2020).

[30] Pengfei Yu and Xuesong Yan. 2020. Stock price prediction based on deep neural networks. *Neural Computing and Applications* 32 (03 2020). https://doi.org/10.1007/s00521-019-04212-x