Machine Learning in
Asset Pricing and Portfolio Optimization

By

Jiawen Liang

Submitted in fulfilment of the requirements of the Degree of
Doctor of Philosophy in Finance

Adam Smith Business School
College of Social Science
University of Glasgow

December, 2024

Abstract

Machine Learning in Asset Pricing and Portfolio Optimization

Jiawen Liang

In the rapidly evolving field of finance, asset pricing and portfolio optimization are facing challenges due to technological advancements, shifting economic landscapes, regulatory changes, and increased complexity in financial markets. This thesis contains three essays that explore the use of advanced machine learning approaches in financial advising and asset pricing. The first essay improves robo-advisors' performance by combining reinforcement learning (RL) with importance sampling that focuses on rare events, leading to better investment outcomes. The second essay employs inverse optimization to estimate investors' risk aversion under normal and disaster conditions, and then optimizes portfolios based on the learnt risk aversion by deep RL. The third essay proposes a framework for asset pricing that uses neural networks to model nonlinear pricing kernels and includes considerations of environmental, social, and governance (ESG) factors in explaining cross-sectional asset prices. Details of the three essays are summarized below:

Robo-advising under rare disasters

Robo-advisors provide automated portfolio management services to investors, and their growth has been unprecedented in the past few years. However, empirical evidence shows that robo-advisors underperformed during the recent COVID-19 pandemic. This may be because rare disasters are highly unlikely to occur and yet have a huge impact on financial markets. Our study develops a novel computational framework to improve the performance and robustness of robo-advising in the presence of rare disasters. It integrates RL with importance sampling. Instead of sampling the transition probability from a ground-truth probability distribution, we sample it from a proposal distribution, where the event of interest occurs more frequently. The proposed algorithm is validated by data covering the 2008 financial crisis and the COVID-19 pandemic, showing superior performance over benchmarked methods. The estimated quarterly return of the robo-advising portfolio using the optimal policy of the proposed algorithm is 0.512%, significantly higher than both the benchmarked policy and the average quarterly return, which are -0.639% and -14.55%, respectively. This improvement is attributed to targeted learning about rare disasters, enabling robo-advisors to reduce exposure to risky assets. The proposed algorithm is model-free and reduces the variance of value estimates through importance sampling.

Risk aversion and portfolio optimization for robo-advising

We develop a novel framework for learning investors' risk aversion using low-resolution data, a common issue arising from short trajectories recording investors' portfolio choices, particularly during disaster events. Furthermore, the observed portfolio choice is often affected by behavioural biases. Our approach combines online inverse optimization with deep RL to simultaneously estimate risk aversion and determine optimal investment strategies under both normal and disaster states. Utilizing real mutual fund data, we demonstrate that our algorithm's risk aversion estimation converges asymptotically to the optimal risk aversion during the learning process. Critically, based on the learned risk aversion and trained deep RL model, we show that robo-advisors adopting our approach can effectively tailor investment strategies to suit investor risk aversion under varying market conditions, outperforming traditional funds. This highlights the potential for our framework to enhance investment decision-making and better represent investor interests in both stable and volatile market environments.

Nonlinear pricing kernels via neural networks

This study proposes a nonlinear pricing kernel approximated through neural networks, addressing limitations of traditional linear models, which capture linear relationships and are prone to overfitting when applied to the factor zoo. The proposed model specification test examines the validity of the nonlinearity assumption of the pricing kernel. Through optimal neural network selection, our findings reveal that a one-layer neural network significantly reduces quadratic pricing errors, indicating its superior pricing performance compared to deep neural networks. Moreover, the role of ESG variables in asset pricing, particularly within the extensive range of factors, remains underexplored. The significance test designed for neural networks shows that ESG variables are significant in asset pricing.

Contents

List of tables

List of figures

Acknowledgements

I would first like to thank my principal supervisor, Professor Cathy Yi-Hsuan Chen, for her invaluable supervision throughout my PhD journey. Her advice has been crucial in completing this thesis, and her dedication has significantly influenced my professional development. Moreover, I am very grateful to my second supervisor, Dr. Bowei Chen, who has offered support whenever needed. He has not only imparted academic knowledge but also valuable lessons beyond my studies.

I extend my sincere gratitude to the Adam Smith Business School at the University of Glasgow for hosting a variety of events and activities that greatly enriched my research experience. In particular, I am grateful for the Brown Bag Seminars, where I benefited from the valuable feedback and insights on my work and presentations. I also appreciate the organization of the Wards Finance Seminars, which further broadened my understanding and deepened my knowledge in the field of finance.

Last but certainly not least, I am immensely grateful to my beloved parents Weiquan Liang and Ruifen Xu for their unconditional support. I am profoundly thankful to my partner, Wanquan Zhang, who fills my life with happiness. Without their support, completing my PhD would not have been possible.

Author’s declaration

I declare that, except where explicit reference is made to the contribution of others, that this dissertation is the result of my own work and has not been submitted for any other degree at the University of Glasgow or any other institution.

Printed Name: Jiawen Liang

Signature:

Chapter 1

Introduction

The financial sector is currently undergoing significant technological advancements, particularly in the field of financial technology. These developments are transforming traditional financial services and driving innovation in areas such as personalized investment, automated trading and risk management. Integrating advanced technologies such as machine learning and artificial intelligence enables financial institutions to improve efficiency, reduce costs and offer more sophisticated products to their clients. Also, applying machine learning to financial theories helps tackle issues that traditional methods cannot answer.

Robo-advisors play critical roles in modern finance by addressing the growing demand for accessible, transparent and efficient wealth management services. Robo-advisors leverage automated algorithms to provide personalized portfolio management services. These platforms make investment advice more affordable and available to a broader audience. By automating routine investment tasks, robo-advisors help reduce the need for costly human advisors and lower entry barriers for investors. In addition, robo-advisors improve operational efficiency for financial institutions, allowing them to serve a more extensive client base with consistent and scalable services. However, financial institutions should be careful about using robo-advisors to avoid the regulatory risk involved in algorithmic investing.

Asset pricing is fundamental to understanding the behaviours of financial markets. It enables the valuation of financial instruments, explains cross-sectional price variations, discovers pricing risk factors, identifies risk exposures and maintains financial stability. The no-arbitrage opportunity requires the pricing kernel to be positive, ensuring that asset prices reflect the assets' intrinsic values. In addition, factor pricing models identify asset returns and risks to help investors make investment decisions and optimize portfolio performance. Furthermore, asset pricing plays a crucial role in risk management by enabling financial institutions and investors to identify and mitigate risk exposures, thereby preventing strategies that could lead to potential losses. Asset pricing is essential for preventing financial crises, as mispriced assets can accumulate financial imbalances and arise in widespread economic turmoil.

The thesis explores robo-advisory because of the following motivations. Primarily, the focus is on algorithmic investing. Robo-advisors are designed to manage and optimize investment portfolios using automated algorithms to enhance long-term investment performance (Abraham et al., 2019). However, rare disaster events can significantly affect the performance of robo-advisors. According to the Barron's Robo reports, the average normalized return of robo-advising portfolios was -14.55% during COVID-19[1], while it was 5.43% before the pandemic[2]. The distinct difference of performance highlights the vulnerability of automated investment strategies to extreme market conditions. Addressing this issue is essential to ensure that robo-advisors can maintain the robust performance during rare economic disasters.

In addition to portfolio optimization, accurately assessing investors' risk tolerance is necessary to ensure the performance of investment strategies. Currently, robo-advisors rely on ”one-size-fits-all” surveys to determine risk profiles. However, cognitive limitations and behavioural biases may lead to imprecise results as the participants' survey answers do not necessarily reflect their true risk preferences without experiencing real-life situations. Additionally, robo-advisory questionnaires often lack comprehensive financial-related questions to fully understand investors' risk profiles. Therefore, improving risk profiling methods through observing portfolio choices is crucial for enhancing the accuracy of portfolio optimization. This motivation drives the second essay of this thesis, which aims to develop more sophisticated techniques for assessing risk tolerance while simultaneously optimizing portfolio choices.

The motivations for exploring asset pricing in this thesis are outlined below. Firstly, traditional asset pricing models often rely on a linear span of risk factors to explain asset returns. However, this approach is debatable when dealing with nonlinear payoffs, as the linear pricing kernel cannot price the nolienar payoffs (Bansal and Viswanathan, 1993). To address this limitation, neural networks offer a powerful alternative by approximating nonlinear pricing

[1] The Robo Report First Quarter 2020: https://storage.googleapis.com/gcs.backendb.com/wordpress/media/2021/02/2020-Q1-Robo-Report.pdf

[2] The Robo Report Fourth Quarter 2019: https://storage.googleapis.com/gcs.backendb.com/1/2020/11/2019Q4-Robo-Report-and-Robo-Ranking.pdf

kernels, providing a more general and flexible representation of the pricing kernel.

Secondly, a factor zoo poses challenges in terms of overfitting during ordinary least squares (OLS) estimation (Cochrane, 2009; Kozak et al., 2018). The factor zoo consists of a large number of pricing factors that have been found to affect asset prices. The high-dimensional factor models can capture noise rather than true signal, leading to models that perform poorly out-of-sample. Therefore, studying SDFs in high-dimensional settings is crucial for reducing generalization errors and, thus, mitigating overfitting.

Third, existing literature, such as Bolton and Kacperczyk (2021) and Sautner et al. (2023a), has demonstrated the significance of ESG factors in low-dimensional factor pricing models. However, it remains unclear whether ESG factors retain their significance within the extensive factor zoo. Investigating this uncertainty is essential to incorporate relevant risk factors in explaining cross-sectional price variations.

The application of RL to portfolio optimization aligns well with the sequential and dynamic nature of financial markets. In portfolio management, investors make a series of decisions over time, adjusting their asset allocations in response to changing market conditions to maximize returns and minimize risks. RL is designed for such sequential decision-making problems, where an agent learns optimal behaviours through interactions with an environment to achieve the long-term goal (Sutton and Barto, 1998). By receiving feedback in terms of rewards or penalties, RL agents adjust their investment strategies over time on behalf of investors.

The interactions between RL agents and the environment simulate the interactions between robo-advisors, their clients and the financial market. Robo-advisors adjust portfolios based on market movements and client feedback, observing the resulting gains or losses, which helps in refining investment strategies over time. Therefore, RL's capacity for sequential decision-making and learning from interactions makes it highly suitable for the context of portfolio optimization in robo-advising examined in this thesis.

In addition to the feasibility of applying RL to portfolio optimization, the model-free RL that this thesis utilizes offers several advantages over traditional methods. First, model-free RL algorithms learn directly from experience without explicitly defining transition probabilities or reward structures. This is particularly beneficial in environments involving disaster states, where predicting future states is unreliable due to the rare occurrence of these events. Second, model-free RL relies on fewer model assumptions. Unlike classical methods such as stochastic control theory and other analytical approaches, which depend heavily on predefined model assumptions, model-free RL leverages large amounts of financial data with fewer assumptions, improving decision-making in complex financial environments (Hambly et al., 2023). Third, model-free RL is good at handling complex and real-world scenarios where the environment is constantly changing and strategies need to adapt over time. On the other hand, traditional models often oversimplify market dynamics to remain computationally tractable.

Neural networks have distinct advantages over conventional parametric models in asset pricing due to their flexibility. Neural networks do not require a fixed functional form. While neural networks are inherently parametric models with parameters learned during training, they are more flexible than the traditional parametric models that assume fixed functional forms. For example, linear pricing kernel models, as described by Hansen and Jagannathan (1991), assume that the pricing kernel spans mean-variance efficient factors. Even nonlinear pricing kernels discussed by Chapman (1997) and Almeida and Freire (2023) impose specific parametric forms such as polynomial functions or parameter estimations, to manage non-linearity and minimize variance among candidate pricing kernels. In contrast, the flexibility of neural networks allows for a more adaptive approach, capturing complex relationships and interactions between input features and the resulting pricing kernels (Gu et al., 2020; Chen et al., 2023).

Moreover, neural networks are well-suited for handling the high dimensionality of the pricing kernel that makes them surpass the capabilities of conventional linear models. Existing research has identified hundreds of pricing factors which are often referred to as a factor zoo that effectively price cross-sectional assets (Cochrane, 2009; Feng et al., 2020). Traditional methods struggle with the curse of dimensionality, where a limited number of data points relative to the number of dimensions adversely affects model performance. Studies by Gu et al. (2020) and Chen et al. (2023) have successfully employed neural networks to address these high-dimensionality challenges. By utilizing neural networks, the third essay provides a flexible estimation of pricing kernels and deals with the high-dimensional nature of asset pricing factors.

Building upon the links with Finance, research motivations and the advantages of applying machine learning to Finance, I discuss each essay's research questions, methodologies, major findings and contributions in the following. Given that each essay is distinct in its focus and methodology, a separate analysis will provide a clear picture of their objectives and results.

Chapter 3 (the first essay) addresses the challenge of maintaining robo-advisory performance during rare disaster events, such as the 2008 financial crisis and the COVID-19 pandemic. These disaster states are rare but impactful events that are not frequently observed in historical data. The study investigates how robo-advisors can optimize asset allocation to maximize investor utility in these extreme conditions. To achieve this, a novel algorithm, SARSA-IS, is introduced, which integrates the tabular reinforcement learning method SARSA with importance sampling. This integration enhances the algorithm's ability to account for rare disaster states, leading to more stable and reliable investment advice.

The first essay contributes to the robo-advisory literature and industry by incorporating model-free RL with importance sampling, thereby enhancing investment performance under rare disaster events. First, the optimized robo-advising portfolios generated by SARSA-IS result in higher average investor utilities compared to benchmarked policies and traditional investor-only approaches. Additionally, the estimated quarterly return of the optimized portfolio is 0.512%, significantly higher than the returns of -0.639% for benchmarked policies and -14.55% for real robo-advising portfolios during COVID-19. Furthermore, the proposed SARSA-IS algorithm reduces the variance of value estimates by effectively converging the proposal disaster probability through importance sampling.

Chapter 4 explores inverse optimization for estimating investors' risk aversion and optimizing investment strategies using deep RL. Traditional risk aversion estimation methods often update risk aversion without considering distinct states, leading to biased estimations that do not accurately reflect the risk preferences specific to disaster states. This study proposes a framework to update investors' risk aversion via inverse optimization based on their portfolio choices in normal and disaster states, respectively. By leveraging deep RL, the model adapts investment strategies in real time, enhancing portfolio management capabilities. The approach was tested against three types of hybrid mutual funds: aggressive, moderate, and conservative allocation types, which include diversified asset classes such as stocks, bonds, and cash. The results demonstrate that the proposed framework not only provides accurate estimates of risk aversion but also achieves superior performance compared to actual mutual fund outcomes, benchmark strategies, and equal-weighted portfolios over the long term.

The second essay has several crucial contributions. Firstly, it introduces an iterative update algorithm that integrates inverse optimization with deep RL, enabling the simultaneous estimation of investors' risk aversion and the optimization of their portfolios. This dual optimization approach ensures that investment strategies are both personalized and responsive to varying risk preferences across different states. Secondly, the study reveals distinct patterns in risk aversion among different investor types. Aggressive investors exhibit the lowest estimated risk aversion, moderate investors display intermediate levels, and conservative investors have the highest estimated risk aversion. Additionally, the research finds that risk aversion is significantly higher in disaster state spaces compared to normal state spaces, highlighting the importance of context-specific risk assessment. Finally, the optimized portfolios derived from the estimated risk aversion models achieve higher cumulative returns than benchmarked simulated portfolios and existing mutual fund portfolios.

Chapter 5 (the third essay) investigates the complexities of pricing kernels in asset pricing through three crucial research questions. The first question examines the unknown functional form of pricing kernels by employing neural networks, allowing for a nonlinear specification. Secondly, the essay studies whether the nonlinear pricing kernel via neural networks performs better than the linear pricing kernel through the model specification test. Given the validity of the nonlinear assumption, the essay finds the optimal neural network architecture. The third question extends the analysis to the influence of ESG factors on pricing kernels, integrating these into the model to see how they affect asset pricing in today's increasingly sustainability-focused financial market.

This research brings innovations to the estimation of pricing kernels. First, the study incorporates a range of characteristics-based factor portfolios to enhance the pricing ability of the SDF model. Second, a model specification test designed for neural networks sets this work apart from the traditional specification test, allowing for a direct comparison between the linear models and neural networks.

The findings from this essay are significant in several aspects. Empirical tests demonstrate that nonlinear pricing kernels consistently outperform linear models, exhibiting at least three times lower out-of-sample squared pricing errors. The nonlinear SDFs are more correctly specified than linear SDFs, with hypothesis test results of $p = 0.90$ at the 5% significance level. Additionally, a neural network with one hidden layer is identified as the best model, achieving hypothesis test results of $p = 0.99$ at the 5% significance level. Furthermore, ESG factors are highly significant in explaining price variations among the factor zoo, highlighting their growing relevance in asset pricing within the context of sustainable finance. These results challenge the conventional SDF models and indicate the potential of neural network-based approaches in explaining cross-sectional and high-dimensional asset pricing.

This essay makes several important contributions to the field of asset pricing. Firstly, it introduces a novel approach by incorporating a range of characteristics-based factor portfolios to enhance the pricing ability of the SDF model. Secondly, the study develops model selection tailored for neural networks. It includes the model specification test to test the nonlinear specification compared to the linear ones and another hypothesis test to select the best neural network configuration among a set of neural networks. Additionally, by integrating ESG factors into the pricing kernel analysis, the research provides valuable insights into their impact on asset prices, contributing to understanding sustainable finance within high-dimensional factor environments.

Chapter 2

Literature review

2.1 Portfolio choices

The first essay in chapter 3 and the second essay in chapter 4 are related to portfolio choices. We study how robo-advising can utilize RL to improve financial advising in portfolio optimization on behalf of their clients in either normal or disaster states. In particular, these two essays contribute to the literature on robo-advising investment, utilizing importance sampling in the presence of rare disasters, as well as employing inverse optimization to estimate investors' risk aversion.

2.1.1 Mean-variance optimization

This thesis relates to the portfolio choice literature, especially about Markowitz's Mean-Variance Optimization (MVO). The concept of MVO is initially introduced by Markowitz (1952). It remains foundational in the fields of asset pricing and portfolio management, forming the basis of Modern Portfolio Theory. Markowitz's framework advocates for an investment strategy that balances expected returns against risk, quantified as variance, to construct an efficient frontier of optimal portfolios that provide the maximum expected return for a given level of risk. This optimization method accommodates investment objectives such as expected returns and acceptable variance, aiming to satisfy investors' expected utilities. The study of portfolio choices in this thesis is based on the objective function of Markowitz's MVO to achieve optimal asset allocation.

Although MVO is widely adopted, it is not without criticism. A significant drawback is its sensitivity to input assumptions. Slight changes in expected returns can lead to vastly different portfolio recommendations. Additionally, MVO heavily relies on the covariance matrix of returns. Michaud (1989) highlights that Markowitz's MVO maximizes the effects of errors arising from the uncertainty in historical returns used to represent expected returns and the estimation errors of portfolio risk. Furthermore, traditional MVO assumes that returns are normally distributed, which may not hold in real-world scenarios where asset returns can exhibit significant skewness and kurtosis (Konno and Yamazaki, 1991). The thesis sections on robo-advising assume returns are normally distributed, focusing on long-term investment returns rather than short-term trading gains. Using longer periods of historical returns to represent expected returns can mitigate the uncertainty caused by short-term fluctuations.

The relevant literature about robo-advisors is noteworthy. First, extant literature discusses the empirical research on user demographics, the factors influencing the adoption of robo-advisors, and the benefits and drawbacks of their use. An initial focus lies on identifying demographic groups more inclined towards robo-advising. Using proprietary data from a significant Indian robo-advisory firm, Baulkaran and Jain (2023) reveal that robo-advisory services predominantly attract young, male, married professionals who are small investors. Additionally, D'Acunto et al. (2019) demonstrate that robo-advisors significantly enhance performance for investor groups that are less diversified. They outline four primary characteristics of robo-adivosrs including personalization, involvement, discretion and human interaction. They argue their potential to enhance financial decisions and mitigate investment errors such as limited exposure to risky assets, inadequate diversification, and behavioural biases (Campbell, 2006). This thesis also illustrates that robo-advisors are beneficial in making investments on behalf of the investors' risk profiles, as humans are prone to cognitive limitations and behavioural biases, especially when facing disaster events. Moreover, Tao et al. (2021) find that robo-advisors outperform conventional funds in terms of risk-adjusted returns in the US financial market, while Phoon and Koh (2017) explore how robo-advisors pose a threat to traditional human advisors, and Brenner and Meyll (2020) find that robo-advisors effectively substitute human financial advisors. Chapter 4 compares the performance of robo-advisors and human advisors to illustrate whether robo-advising outperforms human advising under disaster states. Second, papers on robo-advising model portfolio choices and the dynamics between robo-advisors and their clients, mostly based on the MVO framework, which is commonly used in practice (Beketov et al., 2018; Dai et al., 2021; Capponi et al., 2022). Our thesis sets the expected utility function from MVO as the reward function. Additionally, we utilize RL to optimize the long-term cumulative utilities of investors.

2.1.2 Rare disasters and importance sampling

Rare disasters, such as financial crises, wars, or pandemics, have dramatic and far-reaching effects on investment portfolios. These events, characterized by sudden and severe market downturns, can cause significant short-term financial loss. Rietz (1988) first incorporates the risk of rare disasters into asset pricing models, showing that even the fear of market crashes can raise equity risk premiums to compensate investors for potentially large losses. Building on this, Barro (2006) demonstrates that low-probability economic disasters, such as the Great Depression and World Wars, further explain the equity premium puzzle by showing that the possibility of extreme losses leads investors to demand higher returns for holding risky assets. More recently, Duchin and Harford (2021) highlight how the COVID-19 pandemic caused unprecedented market disruptions, emphasizing the importance of robust asset allocation during such crises. The dramatic effect of rare disasters on investments underscores the necessity for portfolio strategies to withstand such shocks.

The literature offers various approaches to tackle the challenges posed by rare disasters. Some research focuses on robust portfolio optimization techniques that aim to perform well across a range of scenarios, including rare disasters. Anderson et al. (2003) account for model uncertainty and the potential for extreme events. They highlight how decision-makers, fearing model misspecification, tend to adopt robust strategies that safeguard against worst-case scenarios. Nonetheless, these approaches may lead to overly conservative portfolios that sacrifice returns during normal market conditions. In addition, some studies explore the diversification strategies. Bonaccolto and Paterlini (2020), for example, construct portfolio strategies by aggregating multiple existing methods to improve performance during turbulent periods. However, these solutions often have drawbacks, such as over-reliance on historical data that may not accurately predict future rare events and limited adaptability to sudden market changes. It is due to the low probability but high impact nature of rare disasters, which can lead to the suboptimal investment decisions when such events occur.

Essay 1 in chapter 3 addresses the low occurrence of rare disasters by integrating importance sampling in portfolio optimization for robo-advisors. This essay is among the first to consider portfolio optimization for robo-advisors under rare disasters. By utilizing importance sampling, the approach effectively tackles the challenge of rare events in training data that occur infrequently. Importance sampling allows the model to focus on rare but significant events by adjusting the probability distribution from which samples are drawn, enhancing learning efficiency.

2.1.3 Risk aversion and inverse optimization

Understanding investors' risk aversion is crucial in portfolio optimization, particularly within the MVO framework. Investors' risk aversion directly influences the trade-off between expected return and portfolio variance and determines the optimal asset allocation that maximizes their expected utilities. Accurate estimation of risk aversion is essential for constructing portfolios that align with investors' true aversion.

However, robo-advisors often face challenges in estimating investors' risk aversion. These challenges come from limitations in risk assessment methods and the behavioural biases inherent in investors' decision-making processes. Keffert (2024) demonstrates that robo-advising can mismeasure investors' risk preferences, which leads to decreased clients' utilities when measurement errors occur. They find that higher measurement volatility increases utility losses, particularly when frequent interactions are required. Our thesis contributes to estimating the investors' risk aversion, and maximize their long-term cumulative utilities based on the estimated risk aversion.

The literature on risk preference estimation reveals two main approaches. The first approach is questionnaire-based methods that measure investors' risk preferences directly through surveys. These methods develop questionnaires to form risk-tolerance indices and assess financial risk tolerance (Grable, 2000). Using large-scale representative surveys, Dohmen et al. (2011) discover the determinants of risk attitudes and explain risk-taking behaviours. In practice, robo-advisors mainly use online questionnaires to evaluate investors' risk profiles (Tertilt and Scholz, 2018). However, these questionnaires cannot accurately capture the dynamic and state-dependent nature of risk aversion, leading to mis-alignments between advised portfolios and investors' expectations.

The second approach infers risk aversion from the observed investment behaviours, such as household portfolios or investors' portfolio choices. Bucciol and Miniaci (2011) derive the distribution of risk tolerance from U.S. household samples by analyzing actual investment decisions. Alsabah et al. (2021) learn an investor's risk preference through virtual investors' portfolio choices. Their study inversely estimates a parameter of the risk preference based on a known market model, such as the mean-risk utility model, using observable market and investor information. Following Alsabah et al. (2021), Dong et al. (2022) estimate risk aversion using real Chinese market data. In addition, inverse optimization has emerged as a powerful tool for estimating investors' risk aversion from observed behaviours, particularly when direct measurements are noisy or unreliable. Human investors inevitably exhibit behavioural biases and may make mistakes when making decisions and introduce noise into the observed solution data (Foerster et al., 2017). Additionally, the data collection process might involve measurement errors. Consequently, robo-advisors encounter noisy observations of portfolio choices.

To address this challenge, two inverse optimization methods have been developed for noisy data including batch learning (Aswani et al., 2018) and online learning (Yu et al., 2023). Batch learning infers unknown parameters based on a batch of noisy solutions, while online learning updates risk preferences incrementally using observed decisions as they occur rather than waiting to process all observations at once. Online learning, designed to estimate time-varying unknown parameters, has evolved from batch learning techniques (Dong et al., 2018). Online inverse optimization offers several advantages over its batch counterpart, such as significantly accelerating the learning process while maintaining performance guarantees. The existing literature on online inverse optimization demonstrates that these methods converge at a polynomial time rate and achieve statistical consistency (Yu et al., 2023). However, the current invese optimization methods have not considered the economic states. When facing the disaster events, the risk aversion might change dramatically from either the batch estimation or online estimation.

Essay 2 in chapter 4 integrates online inverse optimization into the robo-advising framework under normal or disaster states, respectively. We assume that risk aversion depends on the the normal state space and disaster state space, respectively. This integration allows robo-advisors to adaptively learn and update the risk aversion parameters according to portfolio choices over time. This approach results in not only more personalized portfolio advisors but also resilience to economic changes. By learning portfolio strategies based on the estimated risk aversion via deep RL, robo-advisors improve the clients' satisfaction and thus enhance the aligned performance between advised portfolios and investors' true risk profiles.

2.2 Asset pricing

In this section, we discuss the related literature to the third essay in chapter 5. This essay contributes to the literature on the development of nonlinear pricing kernels, as well as the role of ESG in asset pricing.

2.2.1 Pricing kernels

The essay is related to estimating nonlinear stochastic discount factors (SDFs), often referred to as pricing kernels. A key strand of research on nonlinear SDFs begins with early works like Bansal and Viswanathan (1993), who are among the first to apply neural networks for estimating pricing kernels. Their approach uses neural networks, albeit constrained by the limitations of smaller datasets and a relatively small number of factors. They use neural networks to address over-identification, but their study is limited in terms of scalability and factor inclusion. Building on this, Chapman (1997) applies polynomial functions to approximate nonlinear pricing kernels, introducing consumption growth as the primary factor. Their approach marks a significant step forward but still falls short in capturing the full complexity of asset pricing by limiting the factors influencing the pricing kernel. Another key development comes from Dittmar (2002), who employs a Taylor series expansion to derive a nonlinear pricing kernel, extending the traditional utility-based models to account for higher-order moments like skewness and kurtosis in asset returns. Although this Taylor expansion allows for modelling nonlinearities, its reliance on utility theory constrains flexibility, and the models assume specific functional forms, which may not capture the full extent of market dynamics. More recently, Chen and Ludvigson (2009) propose habit-based models where they employ sieve estimators to capture the habit formation of consumers, allowing for more flexible non-linear estimations of the pricing kernel. They find that their estimated habit function performs better than standard linear models like the Fama-French three-factor model, but their methodology still focuses on fewer factors compared to the factor zoo available today. In contrast, Almeida and Freire (2023) specify the nonlinear factor pricing models by minimizing the variance among candidate pricing kernels, showcasing the economic implications of nonlinear pricing models. Their work demonstrates the superiority of nonlinear models over linear ones, particularly in pricing cross-sectional returns.

The essay contributes to the body of literature by addressing key limitations in previous studies. First, unlike Bansal and Viswanathan (1993), who rely on small-scale neural networks with only a few factors, we incorporate a large set of factors from the factor zoo, leveraging advancements in machine learning techniques to capture a more comprehensive pricing kernel. Additionally, where Chapman (1997) and Dittmar (2002) rely on polynomial functions and the Taylor series, our approach employs multiple layers of neural networks, allowing for a flexible estimation framework. Our work not only compares different neural network configurations but also ensures that the estimated kernels are both economically meaningful and statistically robust across a broader range of factors. In sum, our research fills a crucial gap by extending the nonlinear pricing kernel representation to accommodate larger datasets and more complex factor structures, offering a more flexible and comprehensive solution compared to the existing methods.

2.2.2 ESG

The role of ESG variables in asset pricing has been a subject of ongoing debate. A central question is whether ESG factors can serve as common risk factors in asset pricing models, similar to traditional risk factors such as the market factor, size and value.

On one side of the debate, several studies affirm the relevance of ESG components as significant pricing factors. For instance, Bennani et al. (2018) treat ESG scores as factors influencing abnormal returns and risk exposures in asset pricing models. Engle et al. (2020) utilize environmental scores from Asset4 to construct green factors, demonstrating that these scores capture important risk characteristics. Maiti (2021) finds that ESG factors derived from Bloomberg data are statistically significant when included in an extended Fama-French five-factor model. Bolton and Kacperczyk (2021) identify a carbon risk factor that independently explains abnormal returns, highlighting the financial materiality of environmental risks. Extant literature has also employed textual analysis methods. Engle et al. (2020), Ardia et al. (2022), Sautner et al. (2023a), and Sautner et al. (2023b) construct measures of climate change exposure through textual analysis, finding that these measures effectively price assets. Moreover, machine learning techniques have been applied to ESG data. For instance, Chen and Liu (2020) assess firms' ESG components through topic modeling, using deep learning to forecast returns and form profitable ESG trading strategies. Lanza et al. (2023) and D'Amato et al. (2022) link specific ESG indicators like CO_2 emissions and waste management to abnormal returns and profitability, respectively, using machine learning methods.

Conversely, some research suggests that ESG factors do not qualify as common risk factors in asset pricing models. Halbritter and Dorfleitner (2015) and Naffa and Fain (2022) argue that ESG variables, despite varying data sources including Asset4 and Bloomberg, do not consistently produce abnormal returns or validate themselves within the Fama-French framework. These studies indicate that the inclusion of ESG factors may not significantly improve the explanatory power of traditional asset pricing models.

2.3 Machine learning

Machine learning has emerged as an important tool in finance, offering sophisticated methods for analyzing complex data, capturing nonlinear relationships and enhancing decision-making processes. Broadly speaking, machine learning methods can be categorized into supervised learning, unsupervised learning, and RL. Each category provides unique capabilities suitable for different financial applications. Neural networks can be applied to each category, depending on the nature of the data. For example, neural networks with labelled datasets belongs to supervised learning, while neural networks can be designed for unsupervised learning with unlabelled datasets. If the state space or action space is large-dimensional, neural networks can also be applied to RL for value function approximation.

2.3.1 Supervised learning

Supervised learning involves training models on labelled datasets to learn the mapping from input features to output targets, allowing predictions on new and unseen data. Techniques such as linear regression, support vector machines (SVMs), decision trees and neural networks are popular in this category.

Supervised learning has been extensively applied to problems where historical data with known outcomes are available. Linear regression has long been fundamental in asset pricing models, such as the Capital Asset Pricing Model (CAPM) and the Fama-French factor model (Fama and French, 1992). However, these traditional models assume linear relationships, which may not capture the complexities of financial markets.

Support vector machines have been employed to financial applications. Kim (2003) applies SVMs to forecast stock price index movements and demonstrates that SVMs outperform other stock market prediction methods such as back-propagation neural networks and case-based reasoning. SVMs can handle high-dimensional data and model nonlinear relationships through kernel functions, which make them suitable for financial prediction tasks where market behaviours are complex.

Decision trees predict outcomes by learning decision rules from data features. Bryzgalova et al. (2019) utilize decision tree to predict cross-sectional stock returns for a range of characteristics. Their characteristics are grouped by the decision tree to form the managed portfolios. Although the pricing model is low-dimensional, it is interpretable, which allows us to understand the characteristics that explain price variations.

Neural networks, particularly deep learning models, have gained popularity in supervised learning tasks due to their ability to model complex, nonlinear relationships. Gu et al. (2020) compare linear methods, decision trees, and neural networks in stock return predictions. Their study demonstrates that neural networks outperform traditional linear models in capturing nonlinear relationships and interactions among variables. Using feed-forward networks, they effectively analyze the relationship between input and output variables, which results in enhancing predictive performance in asset pricing and portfolio management.

These supervised learning techniques differ from traditional statistical methods by their ability to handle larger datasets with more variables, model complex nonlinear relationships, and improve predictive accuracy. Their application in finance has led to advancements in asset return prediction, risk assessment, and the development of more sophisticated investment strategies.

2.3.2 Unsupervised learning

Unsupervised learning involves training models on unlabeled data to discover hidden patterns or intrinsic structures within the data. Unlike supervised learning, unsupervised learning does not rely on predefined output labels. Techniques in this category include clustering algorithms, dimensionality reduction and neural networks designed for unsupervised tasks, such as autoencoders (Goodfellow et al., 2016).

Regarding financial applications, unsupervised learning methods are used to uncover hidden structures in financial data, which can inform investment decisions and risk management. Clustering algorithms have been applied to segment financial markets or group similar assets. For example, Tsai et al. (2015) use clustering to segment bank customers based on transaction history and demographic information.

Dimensionality reduction techniques such as principal component analysis (PCA) help reduce the number of variables while retaining essential information. In asset pricing, instrumental PCA (IPCA) has been used to identify underlying latent factors driving asset returns (Kelly et al., 2019). However, PCA captures only linear relationships. To address this limitation, neural networks like autoencoders can capture nonlinear patterns in data. Gu et al. (2021) extend the instrumented principal component analysis by introducing autoencoders to capture nonlinear relationships between factor loadings and firm characteristics. Autoencoders, serving as nonlinear counterparts to PCA, reduce dimensionality by encoding inputs into a lower-dimensional space and then decoding them back to their original dimensions.

Unsupervised learning methods differ from traditional approaches by not requiring labelled data and by their ability to uncover complex and nonlinear structures. Unsupervised learning provides a data-driven approach to discover insights that may not be evident through conventional analysis.

2.3.3 Reinforcement learning

Reinforcement learning enables autonomous agents to learn to make decisions based on interactions with the environment, aiming to maximize cumulative rewards in the long run (Sutton and Barto, 1998). Unlike supervised learning, which learns from labelled examples, or unsupervised learning, which discovers patterns in data, RL learns optimal actions through trials and errors.

RL has been applied to portfolio management, trading strategies, and robo-advising. Its ability to model sequential decision-making and adapt to changing environments makes it suitable for financial applications where decisions have long-term consequences. For instance, Almahdi and Yang (2017) use RL to learn the optimal trading strategy over time. The proposed RL method is effective when using different transaction costs.

In the context of order execution, Gao and Xu (2022) develop an order scoring model integrated with multi-armed bandit learning from RL to quantify the performance of a limit order before execution. Additionally, Schnaubelt (2022) designs an RL environment to optimize limit order placement, concluding that Proximal Policy Optimization learns superior order strategies compared to other RL algorithms. These studies demonstrate the superiority of RL in handling complex financial tasks.

Neural networks are often employed as function approximators to estimate value functions or policies in high-dimensional spaces. Deep RL, which combines RL with deep neural networks, has been pivotal in advancing RL applications in finance. For example, Deng et al. (2016) apply deep RL to high-frequency trading, where the neural network learns complex trading policies based on vast amounts of market data. Jiang et al. (2017) utilize a model-free algorithm of deep RL for the portfolio management problem, demonstrating that RL can handle the complexities of financial markets and adapt to dynamic environments. Maeda et al. (2020) employ deep RL to optimize stock-trading strategies in agent-based artificial price-order-book simulations, showing the effectiveness of RL in developing trading strategies in simulated environments.

2.3.4 Model selection

Model selection in neural networks involves choosing the optimal architecture and hyperparameters that balance model complexity and predictive performance. It is essential to prevent overfitting and ensure that models generalize well to new data. The performance of neural networks can vary widely based on chosen hyperparameters and random elements like weight initialization and dropout masks due to their highly non-convex loss functions (Li et al., 2018). This variability makes it challenging to compare the performance of various network architectures directly and complicates the model selection process.

Pioneering work by White (1989) introduces a statistical perspective for neural networks by deriving asymptotic distributions for network parameters, allowing for hypothesis testing and model comparison. Building on this foundation, Anders and Korn (1999) apply White's methods along with cross-validation and information criteria such as the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) for model selection in neural networks. They demonstrate that these criteria could be adapted to evaluate neural network models, providing a balance between goodness-of-fit and model complexity.

2.3.5 Model interpretability

While neural networks have demonstrated superior performance in modelling complex financial data, they often face challenges regarding model interpretability. The ”black box” nature of neural networks makes it difficult to understand the underlying decision-making process, which is crucial for regulatory compliance and trust in Finance.

Several studies have discussed interpretability in neural networks. Farrell et al. (2021) propose methods for statistical inference in deep learning models, introducing significance tests for parameters within neural networks. Their approach adapts traditional econometric techniques to the deep learning framework, allowing for hypothesis testing and confidence intervals in high-dimensional settings. This enables researchers to assess the importance of individual variables and interactions, enhancing model interpretability. Similarly, Fallahgoul et al. (2024) apply significance tests to neural networks with various hidden layers in the context of asset pricing. By conducting hypothesis testing on the estimated parameters, they aim to identify the most relevant factors influencing asset returns and improve model transparency. However, they acknowledge the difficulty in selecting the optimal neural network architecture and emphasize the need for systematic model selection procedures. These significance tests designed for neural networks can be applied to increase the transparency of asset pricing models.

Chapter 3

Robo-advising under rare disasters

Abstract

Robo-advisors provide automated portfolio management services to investors, and their growth has been unprecedented in the past few years. However, empirical evidence shows that robo-advisors underperformed during the recent COVID-19 pandemic. This may be because rare disasters are highly unlikely to occur and yet have a huge impact on financial markets. Our study develops a novel computational framework to improve the performance and robustness of robo-advising in the presence of rare disasters. It integrates RL with importance sampling. Instead of sampling the transition probability from a ground-truth probability distribution, we sample it from a proposal distribution, where the event of interest occurs more frequently. The proposed algorithm is validated by data covering the 2008 financial crisis and the COVID-19 pandemic, showing superior performance over benchmarked methods. The estimated quarterly return of the robo-advising portfolio using the optimal policy of the proposed algorithm is 0.512%, significantly higher than both the benchmarked policy and the average quarterly return, which are -0.639% and -14.55%, respectively. This improvement is attributed to targeted learning about rare disasters, enabling robo-advisors to reduce exposure to risky assets. The proposed algorithm is model-free and reduces the variance of value estimates through importance sampling. In addition to methodological contributions, our study contributes to the growing literature on robo-advising by considering rare events.

Keywords: artificial intelligence; reinforcement learning; robo-advising; importance sampling; rare disasters; portfolio management

3.1 Introduction

Portfolio management entails selecting and supervising a collection of investments aimed at achieving an investor's financial goals, such as maximizing expected returns and minimizing risks. Researchers and professionals have extensively explored this area across various disciplines, including economics, finance, operations research, and computer science. *Robo-advisors* are the digital platforms that use algorithms to provide investors with automated portfolio management services (Abraham et al., 2019). Since their inception in 2008, robo-advising has experienced tremendous growth. According to Statista, the value of assets managed by robo-advisors is expected to reach US $2.8 trillion by 2025, with 478 million users.[1]

The COVID-19 pandemic has accelerated the adoption of digital technologies, whereas ensuring the performance of robo-advisors has been challenging during the pandemic. A recent McKinsey survey indicates that the COVID-19 pandemic has driven companies to adopt digital technologies, leading to substantial business transformations[2]. Despite the accelerated adoption of robo-advisors, maintaining their performance during the pandemic remains challenging. The Barron's Robo reports reveal that the average normalized return of robo-advising portfolios was -14.55% during COVID-19[3], while it was 5.43% before the pandemic[4]. The reasons behind the sub-optimal performance of robo-advisors during rare disasters are yet to be understood. One possible explanation is that robo-advising applications are relatively new and have not been thoroughly examined across long economic cycles. In this study, we address this challenge by proposing a disaster-adaptive robo-advising framework, enabling robo-advisors to optimize their investment strategies in the presence of rare disasters.

Asset allocation becomes particularly important in managing investments during rare disasters (Duchin and Harford, 2021). Bonaccolto and Paterlini (2020) construct portfolio

[1] Statista: https://www.statista.com/outlook/dmo/fintech/digital-investment/robo-advisors/worldwide?currency=usd.

[2] McKinsey: https://www.mckinsey.com/business-functions/strategy-and-corporate-finance/our-insights.

[3] The Robo Report First Quarter 2020: https://storage.googleapis.com/gcs.backendb.com/wordpress/media/2021/02/2020-Q1-Robo-Report.pdf

[4] The Robo Report Fourth Quarter 2019: https://storage.googleapis.com/gcs.backendb.com/1/2020/11/2019Q4-Robo-Report-and-Robo-Ranking.pdf

strategies by aggregating multiple existing strategies to improve asset allocation and portfolio performance. We focus on asset allocation for robo-advising, enabling robo-advisors to choose assets that maximize the investor's utility during disaster events. We propose a novel algorithm called *SARSA-IS*, integrating a well-established RL algorithm, *State-Action-Reward-State-Action* (*SARSA*) with *importance sampling* (*IS*). RL is a type of machine learning technique in which autonomous agents learn to make decisions based on the realizations and interaction with the environment, aiming to maximize the cumulative rewards in the long run (Sutton and Barto, 1998). It has been used in Finance to solve optimization problems and improve decision-making (Akbarzadeh et al., 2018; Maeda et al., 2020; Gao and Xu, 2022; Schnaubelt, 2022). Importance sampling allows us to effectively capture rare events, addressing the limitations of existing robo-advising algorithms during the COVID-19 pandemic. For new developments about COVID-19, Khalilpourazari and Hashemi Doulabi (2021) propose a novel hybrid RL approach for modelling and forecasting the COVID-19 pandemic.

The proposed SARSA-IS algorithm is examined with financial data containing the 2008 financial crisis and the recent COVID-19 outbreak, and the key parameters are initiated by the economic and market outlook reports of the NBER and Vanguard (Alsabah et al., 2021). Compared to benchmarked methods, we find that the optimal SARSA-IS policy achieves relatively higher rewards and value estimates in rare disasters. The resulting optimized policy advises a cautious strategy whenever the probability of transition to a disaster state is relatively high. We extensively compare the performance of SARSA-IS, SARSA, and the investor-only approach. For the latter, investors are supposed to invest their assets and maximize their utilities by themselves. They may commit investment mistakes such as behavioural biases and cognitive limitations (Foerster et al., 2017) when dealing with their own investment. We show that during the COVID outbreak, the performance of the proposed robo-advising algorithm is salient, with an estimated quarterly return using the optimal policy of up to 0.512%, compared to -0.639% generated by the benchmark policy and -14.55% by the average quarterly return of existing robo-advising portfolios.

Several pieces of literature about robo-advising are worth mentioning. D'Acunto et al. (2019) show that the performance of portfolios managed by robo-advisors improved significantly for less-diversified investor groups. Capponi et al. (2022), and Dai et al. (2021) investigate the use of the mean-variance approach in robo-advising separately. Tao et al. (2021) find that robo-advisors have better risk-adjusted performance than conventional funds on the US financial market. Phoon and Koh (2017) investigate how the rise of robo-advisors threatened traditional human advisors, and Brenner and Meyll (2020) confirm that robo-advisors have a substitution effect on human financial advisors. Van Staden et al. (2021) point out that robo-advisors require human intervention due to their unstable capability of asset allocation. Alsabah et al. (2021) are perhaps the earliest researchers that use RL to learn investors' risk preferences through multiple interactions between robots and investors. Gan et al. (2021) find that investors, particularly those with higher financial literacy, are inclined to adopt robo-advisors during the crisis. Giudici et al. (2022) combine financial network analysis with portfolio optimization techniques in order to better understand and manage risk in robo-advisory portfolios during crisis times.

Our study contributes to the growing literature on robo-advising by considering the impact of disaster events like the 2008 financial crisis and the COVID-19 pandemic. We develop a computational framework based on RL that can be deployed in real-world robo-advisory systems. Integrating importance sampling enables robo-advisors to perform well and remain reliable during rare events. Thus, we develop a novel application that utilizes machine learning or, more broadly, artificial intelligence (AI) to support portfolio management in robo-advising. Our development yields several outputs, including optimized policy during disasters, reduced variance of value estimates, increased rewards, and value functions.

Emphasizing disaster states in portfolio optimization is economically meaningful due to the profound impacts that disaster states can have on portfolios. Although disaster states occur infrequently, their potential shocks are substantial, aligning with Nicholas (2008)'s Black Swan theory, which illustrates that such rare events own high uncertainty and disastrous impact. The disaster states are typically difficult to predict but can result in significant asset losses when they occur, representing extreme risk. Ignoring the possibility of these events can significantly increase the vulnerability of investment portfolios, enabling them to be ineffective to deal with severe market fluctuations. Furthermore, historical analysis reveals consistent patterns and underlying causes of financial disasters, as emphasized by Reinhart (2009). By incorporating disaster states more frequently through importance sampling, this paper investigates asset allocation strategies that leverage historical disaster events. Utilizing RL to analyze historical data allows for the identification of recurring patterns and mechanisms, thereby enhancing the ability to recognize and respond to such extreme events effectively.

Emphasizing disaster states has significant economic implications for investor behaviours, asset allocation and risk management. From the perspective of investor behaviours, investors react more intensely to potential losses than equivalent gains according to the Prospect theory from Kahneman and Tversky (2013). Thus, exposure to potential extreme risks often leads investors to adjust their asset allocation strategies, seeking safe-haven assets. By increasing the sampling frequency of disaster states within the model, the study can more accurately reflect investor decision-making under extreme conditions.

In terms of asset allocation, incorporating disaster states enables adjustments to the asset portfolio proportions based on investor expectations, risk preferences, and prevailing market conditions during disaster states. This strategic adjustment helps mitigate potential losses, thereby enhancing investors' utility from their portfolios which can be supported by the Model Portfolio Theory (Markowitz, 1952). Regarding risk management, disaster states are typically associated with significant market volatility and exposure to systemic risks. Acharya et al. (2017) emphasize that failure to adequately account for these disaster states in portfolio optimization can result in unforeseen substantial losses, undermining the stability of overall returns. Therefore, employing importance sampling to increase the representation of disaster states allows for a more accurate capture of the potential impacts of these extreme scenarios.

The remainder of the paper is organized as follows. Section 3.2 introduces our proposed methodology framework. Section 3.3 describes data and provides an analysis of robo-advisors in rare disasters. Section 3.4 concludes the paper.

3.2 Learning about rare disasters

In this section, we first introduce the mathematical preliminaries of RL and importance sampling in the context of robo-advising. We then discuss the technical details of the proposed algorithm SARSA-IS.

3.2.1 Portfolio allocation

We start with a *Markov decision process* (*MDP*) environment for RL, where robo-advising (also called *agent*) interacts with the environment at a sequence of discrete-time, denoted by $t = 0, 1, 2, ...$, and with a state space that contains different economic scenarios in financial markets, denoted by $\mathcal{S}$. Each state can be specified for portfolio management by economic variables such as market return and volatility. For example, we denote $s_t = 0$ for the *low return volatility state*, which represents the state with the low expected market return and volatility; $s_t = 1$ for the *high return volatility state* that represents the state with the high expected market return and volatility; and $s_t = 2$ for the *disaster state* with the dramatic negative expected market return, notoriously high fluctuation and low probability of occurrence. At time t, the robo-advisor observes some features representing the state $s_t \in \mathcal{S}$, and takes action from an action space $a_t \in \mathcal{A}$. In the next time step $t+1$, it receives the resulting reward r_{t+1} and moves into the next state s_{t+1}.

There are several assumptions. First, the robo-advisors' actions are independent of each other and conditional on the state. Second, each individual's decision has a minor (or trivial) effect on the overall market. Third, the transition probability from state s to state s' is independent of the portfolio choice, denoted by $p(s'|s) = \mathbb{P}\{s_{t+1} = s'|s_t = s\}$ for all $s, s' \in \mathcal{S}$. The third assumption distinguishes the optimal learning policy in the context of robo-advising from the conventional RL setting, where the action is a conditioning variable in the transition probability.

In accordance with the optimal policy, it is important to note that the action taken at state s_t represents the portfolio allocation strategy across N assets. To be more specific, we define the action set $\mathcal{A} = \{a^{(1)}, a^{(2)}, \ldots, a^{(i)}\}$, where each $a^{(i)}$ represents a possible portfolio weight vector within the space $[0,1]^N$. Robo-advisors solve an optimization problem by learning an optimal policy that maps the state s_t to the action $a^{(i)}$. In response to state transitions, robo-advisors are expected to select the optimal action. For the sake of simplification, we limit the portfolio to consist of one risky asset and one risk-free asset. The risky asset can represent the stock market, which exhibit returns and volatility that fluctuate with states. Meanwhile, the risk-free asset serves as a safe haven. We assume that short sales are not permitted. Consequently, the weight vector reduces to a scalar $a \in (0,1)$ for the risky asset and the remaining $1-a$ for the risk-free asset.

Given the portfolio return in the next-period $R_p(s',a)$, we consider the maximization of rewards defined in the mean-variance analysis (Markowitz, 1952), that is

$$r(s,a,s') = \mathbb{E}[R_P(s',a)] - \theta \mathrm{Var}[R_P(s',a)], \tag{3.1}$$

where $\mathbb{E}[\cdot]$ is the expectation operator, $\mathrm{Var}[\cdot]$ is the variance operator, $r(s,a,s')$ is the expected reward for the state-action-next state triple $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, and θ is the risk-aversion parameter representing the client's risk aversion. We assume the risk aversion is given, as they have been extensively studied in existing literature (Bucciol and Miniaci, 2011; Alsabah et al., 2021).

In our method, at each time step t, robo-advisors allocate a fraction a of the investor's wealth to the risky asset and the remaining $(1-a)$ fraction to the risk-free asset under state s. The expected portfolio return is given by $\mathbb{E}[R_P(s',a)] = a\mathbb{E}[R_M(s')] + (1-a)R_F$, where $R_M(s')$ represents the rate of return on the risky asset in the market portfolio in the next state, and R_F denotes the rate of return on the risk-free asset. Consequently, the variance of the portfolio is $\mathrm{Var}[R_P(s',a)] = a^2\sigma_M^2(s')$, where $\sigma_M(s')$ is the standard deviation of the risky asset return in the subsequent state s'.

3.2.2 Reinforcement learning

RL consists of two building blocks: policy evaluation and policy control. The former iteratively estimates the value function, while the latter improves the given policy through *greedy policy improvement* (Sutton and Barto, 1998). In policy evaluation, it is crucial to compute the state value function $V^\pi(s)$ or the state-action value function $Q^\pi(s,a)$ for a given policy. The policy $\pi(a|s) = P(a_t = a|s_t = s)$ represents the probability of actions conditional on the state. The state value function $V^\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s)Q^\pi(s,a)$ represents the total sum of the probability of choosing an action or policy multiplied by the state-action value for each action in a state.

Value functions can be solved using the Bellman equation (Singh and Sutton, 1996). For

$s, s' \in \mathcal{S}$, the state value function can be expressed as:

$$V^{\pi}(s) = \sum_{a \in A} \pi(a|s) \sum_{s' \in S} p(s'|s,a) \Big[r(s,a,s') + \gamma V^{\pi}(s') \Big], \tag{3.2}$$

where $r(s,a,s')$ is the reward received after transitioning from state s to state s' via action a. γ is the discount factor, representing the importance of future rewards relative to immediate rewards.

The state-action value function $Q^{\pi}(s,a)$ can be written as:

$$Q^{\pi}(s,a) = \sum_{s' \in S} p(s'|s,a) \Big[r(s,a,s') + \gamma \sum_{a' \in A} \pi(a'|s') Q^{\pi}(s',a') \Big]. \tag{3.3}$$

Evaluating the Bellman equation is costly, as it requires full information, including $p(s'|s,a)$, $\pi(a'|s')$, and $Q(s',a')$ for all possible states and actions in the next step. Instead, *temporal difference* (*TD*) learning can be used. It is model-free (i.e., requires no knowledge of an MDP) and avoids the need to simulate the entire trajectory until reaching the terminal condition. TD learning employs the bootstrap technique to estimate the value function at the next step, and the value function can be updated as follows:

$$V(s) \leftarrow V(s) + \alpha \Big[r(s,a,s') + \gamma V(s') - V(s) \Big], \tag{3.4}$$

where α is the learning rate.

Policy control aims to find the policy that maximizes the value function $Q(s,a)$ for action $a \in \mathcal{A}$ and $s \in \mathcal{S}$. $Q(s,a)$ is updated by adjusting it towards the estimated optimal future value. It can be achieved using SARSA with the following recursive state-action value function.

$$Q(s,a) \leftarrow Q(s,a) + \alpha \Big[r(s,a,s') + \gamma Q(s',a') - Q(s,a) \Big]. \tag{3.5}$$

3.2.3 Probability of rare disasters

Let $\varepsilon(s) \in [0,1]$ denote the probability of a rare disaster that might occur in state s. We define the two sub-spaces of state, $\mathcal{D}$ as the disaster state space, and its complement state space $\mathcal{D}^c = \mathcal{S} \setminus \mathcal{D}$. Then the transition probability $p(s'|s)$ can be expressed as a mixture of two independent transition probabilities:

$$p(s'|s) = \begin{cases} (1-\varepsilon(s))f(s'|s) & \text{if } s' \in \mathcal{D}^c, \\ \varepsilon(s)g(s'|s) & \text{if } s' \in \mathcal{D}. \end{cases} \tag{3.6}$$

where $f(s'|s)$ is the transition probability that characterizes the normal environment for $s' \in \mathcal{D}^c$, $g(s'|s)$ is the transition probability to disaster events $s' \in \mathcal{D}$, and the disaster probability $\varepsilon(s)$ is also a mixture weight that governs the dominance between two component probability distributions. In this case, the total sum of transition probabilities $\sum_{s'} p(s'|s)$ equals 1, ensuring that the expressions of the transition probability are valid.

In theory, the mixture weight $\varepsilon(s)$ in eq. (3.6) is expected to be small, suppressing the impact of a rare disaster on the transition process. We, therefore, employ a *proposal distribution* $\hat{\varepsilon}(s)$ where the disaster events occur more frequently than $\varepsilon(s)$. The estimated $\hat{\varepsilon}(s)$ varies and is updated during the simulation, and $\hat{\varepsilon}(s)$ (rather than $\varepsilon(s)$) determines a fraction between the unchanged $f(s'|s)$ and $g(s'|s)$. Eventually, $\hat{\varepsilon}(s)$ converges to optimal $\varepsilon^*(s)$ that minimize the variance under an exhaustive simulation which will be proved in Section 3.2.5. The transition probability distribution under the mixture weight governed by $\hat{\varepsilon}(s)$ becomes

$$q(s'|s) = \begin{cases} (1-\hat{\varepsilon}(s))f(s'|s) & \text{if } s' \in \mathcal{D}^c, \\ \hat{\varepsilon}(s)g(s'|s) & \text{if } s' \in \mathcal{D}. \end{cases} \tag{3.7}$$

3.2.4 Optimal policy

We aim to optimize investment strategies for decision-making empowered by robo-advisors through policy control. Not all the investment strategies adopted by robo-advising are the best ones. The optimal policy π^* is declared to be better than or equal to any other policy

π' if its expected reward is larger than or equal to that of π' for all states (Sutton and Barto, 1998). The robo-advisors' objective is to find an optimal policy to maximize the investor's cumulative rewards over a lifetime. The rewards are the investor's utilities in the paper.

The optimal policy is the one that maximizes the state-action value function $Q^{\pi}(s,a)$ in eq. (3.3):

$$\pi^* = \operatorname*{argmax}_{\pi} Q^{\pi}(s,a) \tag{3.8}$$

To search for the optimal policy, one can adopt various TD learning methods such as SARSA or Q-learning (Watkins, 1989). SARSA is an on-policy learning algorithm, meaning it updates $Q^{\pi}(s,a)$ based on the actions taken under the current policy π, and improves π via ε-greedy exploration. In contrast, Q-learning is an off-policy algorithm that updates $Q^{\pi}(s,a)$ using the greedy policy, independent of the agent's actions. We adopt SARSA for its faster convergence and higher learning speed (Wang et al., 2013), which is particularly beneficial for applications involving rare disaster events.

The ε-greedy policy improvement is a strategy to greedily improve the policy by choosing the greedy action with a probability of $1-\varepsilon$, and with a probability of ε, selecting action $a \in A$ randomly. ε decreases as the episodes progress. In the first few episodes, more actions are chosen randomly to ensure exhaustive exploration. As the episode develops, the optimal action is chosen with a higher probability of maintaining exploitation in the following episodes. When the policy converges to the optimal policy, an optimal action that optimizes cumulative rewards for each different state will be chosen with a probability close to one.

Eligibility traces are widely applied to operate step-by-step updates upon observing samples. Instead of updating the value function by looking forward to the entire trajectory, eligibility traces make it possible to propagate the update backwards to the state at the last step along the trajectory.

In our approach, we incorporate replacing eligibility traces into the SARSA algorithm to enhance its learning efficiency. Replacing eligibility traces are proposed to provide a stable and efficient approach of eligibility traces (Singh and Sutton, 1996).

To implement SARSA with replacing eligibility traces, we update the eligibility traces as follows for each state-action pair:

$$e_t(s,a) = \begin{cases} 1 & \text{if } s = s_t \text{ and } a = a_t, \\ \gamma\lambda e_{t-1}(s,a) & \text{otherwise.} \end{cases} \tag{3.9}$$

The decay rate of eligibility traces, denoted as λ, ranges from 0 to 1. A value of $\lambda = 0$ simplifies the algorithm to the standard SARSA update eq. (3.5), whereas $\lambda = 1$ aligns it with Monte-Carlo methods.

At state s, robo-advisors choose an action a according to the ε-greedy policy derived from the state-action value $Q(s,a)$, observe reward $r(s,a,s')$, and transition to the next state s'. The update rule for SARSA with replacing eligibility traces is then given by:

$$Q(s,a) \leftarrow Q(s,a) + \alpha e(s,a)\left[r(s,a,s') + \gamma Q(s',a') - Q(s,a)\right]. \tag{3.10}$$

The technical details of the vanilla SARSA are provided in Appendix 3.A.

3.2.5 Optimal policy with importance sampling

In the presence of rare disasters, the conventional SARSA encounters the problem of a less consistent approximation toward the value function. With such an obstacle, the process is likely to fail to converge within a given period. Hinich (2003) showed that it took at least ten times the average time for the agent to generate sufficient outcomes of the low probability event so that all agents shared the same objective knowledge of distribution. To meet this challenge, we propose SARSA-IS (see an algorithm 1) to draw an alternative sampling probability in the next state, which can accelerate policy learning for rare disasters.

The probability distribution is replaced by a proposal distribution that features a fatter-tailed distribution than the true one. By exploiting the distribution with a higher likelihood of unusual random variables, robo-advisors benefit from effective updating since learning is not constrained by the required sample size or the number of learning episodes. Suppose one believes that important values have a more significant impact on the parameter being estimated than others. In that case, the main idea is to sample the “important region” of the distribution more often and to choose a distribution that encourages the important but unusual samples. By doing so, robo-advisors observe samples in the extreme area of the distribution more often, accelerating learning and policy updating. Importance sampling resolves the issue of excessive learning duration induced by a low probability of occurrence. More importantly, importance sampling is a variance-reduction technique commonly used in rare event simulation (Juneja and Shahabuddin, 2006). If the sampling probability is modified effectively, using samples from the alternative distribution can reduce the variance of estimators.

SARSA is modified by taking the presence of rare disasters into account. To overcome the learning difficulty in the presence of rare disasters, sampling the future realization is undertaken by the proposal sampling distribution q that favours sampling disaster events. Frank et al. (2008) mentioned that the bias and variance of the importance sampling weight can be substantially reduced. To correct samples that are biased due to sampling q rather than p, we have importance sampling weights as:

$$w(s,s') = \frac{p(s'|s)}{q(s'|s)}. \tag{3.11}$$

It should be noted that we assume that robo-advisors’ actions do not affect the market environment. Given that the state space $\mathcal{S}$ is split into two subspaces, and the proposal transition probability in eq. (3.7), the importance sampling weight boils down to

$$w(s,s') = \begin{cases} \varepsilon(s)/\hat{\varepsilon}(s) & \text{if } s' \in \mathcal{D}, \\ (1-\varepsilon(s))/(1-\hat{\varepsilon}(s)) & \text{if } s' \in \mathcal{D}^c. \end{cases} \tag{3.12}$$

where $\varepsilon(s)$ is the true disaster probability and $\hat{\varepsilon}(s)$ is the proposal rare event probability.

The optimal disaster probability can be characterized by choose an alternative sampling distribution that minimizing the variance of state-action value function due to the change of trajectory. The optimal form of disaster probability are as follows.

$$\varepsilon^*(s) = \varepsilon(s) \frac{\sum_{s' \in \mathcal{D}} g(s'|s) \Big[r(s,a,s') + \gamma \sum_{a' \in A} \pi(a'|s') Q^{\pi}(s',a') \Big]}{Q^{\pi}(s,a)}, \tag{3.13}$$

Proof: see Appendix 3.B.

With the observation that the state-action values for the disaster state space $\mathcal{D}$ is

$$Q^{\pi}_{\mathcal{D}}(s,a) = \varepsilon(s) \sum_{s' \in \mathcal{D}} g(s'|s) \Big[r(s,a,s') + \gamma \sum_{a' \in A} \pi(a'|s') Q^{\pi}(s',a') \Big]. \tag{3.14}$$

Eq. (3.13) can be rewritten as

$$\varepsilon^*(s) = \frac{Q^{\pi}_{\mathcal{D}}(s,a)}{Q^{\pi}(s,a)}. \tag{3.15}$$

If states and actions are sampled according to the optimal policy, $\hat{Q}^{\pi}_{\mathcal{D}}(s,a)$ and $\hat{Q}^{\pi}(s,a)$ converge to the optimal and unbiased values, so we also have the estimated disaster probability converges to the optimal disaster probability as follows.

$$\hat{\varepsilon}(s) = \frac{\hat{Q}^{\pi}_{\mathcal{D}}(s,a)}{\hat{Q}^{\pi}(s,a)} \to \varepsilon^*(s). \tag{3.16}$$

In the algorithm 1, we take the absolute value of $\hat{Q}^{\pi}_{\mathcal{D}}(s,a)$ and $\hat{Q}^{\pi}_{\mathcal{D}}(s,a)$ in eq. (3.16), mainly because the expected cumulative rewards of taking state s and action a can be both positive or negative in general cases. A boundary is set for $\hat{\varepsilon}(s)$ to ensure an exhaustive exploration of the optimal value $\varepsilon^*(s)$. Hence, $\hat{\varepsilon}(s) \in (\delta, 1-\delta)$ is bounded. If $\hat{\varepsilon}(s)$ is computed outside the boundary, it takes the value of δ or $1-\delta$ as follows

$$\hat{\varepsilon}(s) \leftarrow \min \left\{ \max \left\{ \delta, \frac{|\hat{Q}^{\pi}_{\mathcal{D}}(s,a)|}{|\hat{Q}^{\pi}(s,a)|} \right\}, 1-\delta \right\}.$$

The state-action value function that over-weights the important samples is

$$\hat{Q}^{\pi}(s,a) \leftarrow \hat{Q}^{\pi}(s,a) + \alpha e(s,a) \Big[w\big(r(s,a,s') + \gamma \hat{Q}^{\pi}(s',a') \big) - \hat{Q}^{\pi}(s,a) \Big], \tag{3.17}$$

where w is the importance sampling weight in eq. (3.12) to correct biases.

We can then estimate Q values using the form of Bellman equation. If $s' \in \mathcal{D}$, then

$$\hat{Q}^{\pi}_{\mathcal{D}}(s,a) \leftarrow (1-\alpha)\hat{Q}^{\pi}_{\mathcal{D}}(s,a) + \alpha\varepsilon(s)\Big(r(s,a,s') + \gamma\hat{Q}^{\pi}(s',a')\Big). \tag{3.18}$$

Proof: see Appendix 3.C.

The eligibility traces of SARSA-IS can be updated as

$$e_t(s,a) \leftarrow \begin{cases} 1 & \text{if } s = s_t \text{ and } a = a_t, \\ \gamma\lambda w e_{t-1}(s,a) & \text{otherwise.} \end{cases} \tag{3.19}$$

In addition to the algorithm 1, there is a flow chart 3.1 for SARSA-IS to understand the algorithm.

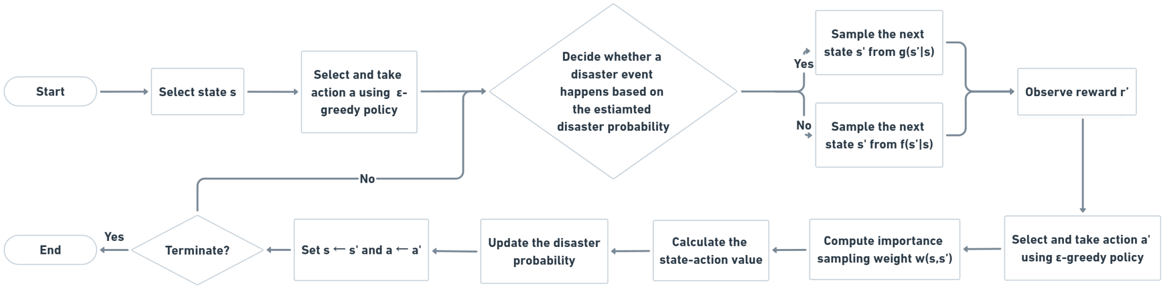

Start
Select state s
Select and take action a using ε-greedy policy
Decide whether a disaster event happens based on the estiamted disaster probability
Yes
Sample the next state s' from g(s'|s)
No
Sample the next state s' from f(s'|s)
Observe reward r'
Select and take action a' using ε-greedy policy
Compute importance sampling weight w(s,s')
Calculate the state-action value
Update the disaster probability
Set s ← s' and a ← a'
Terminate?
No
Yes
End

Fig. 3.1: Flow chart for SARSA-IS algorithm

One can obtain and compare the optimal policy π^* generated by SARSA-IS and the benchmark policy π obtained from SARSA. We propose the *disaster events adaptive importance sampling* (abbreviated as *DEIS*) (see algorithm 2). It improves TD learning through importance sampling in the policy evaluation. We outline the difference between DEIS and SARSA-IS. First, DEIS has the state value function when evaluating the given policy $V^{\pi}(s)$ in the presence of disaster states while SARSA-IS uses a state-action value function $Q^{\pi}(s,a)$. Second, DEIS is designed for evaluating a specific policy, and thus cannot improve the policy as SARSA-IS does. The state value function of DEIS is

$$\hat{V}^{\pi}(s) \leftarrow \hat{V}^{\pi}(s) + \alpha e(s)\Big[w\Big(r(s,a,s') + \gamma\hat{V}(s')\Big) - \hat{V}(s)\Big]. \tag{3.20}$$

Algorithm 1 SARSA with importance sampling (SARSA-IS)

```
Input: Rare event set $\mathcal{D}$, normal event set $\mathcal{D}^c$, true rare-event probabilities $\varepsilon(s)$, transition probabilities $f(s'|s)$, $g(s'|s)$, the learning rate $\alpha$, the discount rate $\gamma$, the greedy parameter $\varepsilon$ and the parameter $\delta > 0$.
Initialize state action values $\hat{Q}^{\pi}(s,a), \hat{Q}^{\pi}_{\mathcal{D}}(s,a), \hat{Q}^{\pi}_{\mathcal{D}^c}(s,a)$ arbitrarily, $\forall s,a$;
Initialize the rare event sampling distribution $\hat{\varepsilon}(s) = 0.5, \forall s$;
Initialize eligibility traces $e(s,a) = 0$;
Select the initial state $s$;
Select an action $a$ using $\varepsilon$-greedy policy;
for each iteration do
    Update eligibility traces $e(s,a)$ in eq. (3.19);
    Take the action $a$;
    Decide whether a disaster event happens based on $\hat{\varepsilon}(s)$. If a disaster event is determined to occur, sample $s'$ from the disaster event transition distribution $g(s'|s)$. Otherwise, sample $s'$ from normal state transition distribution $f(s'|s)$.
    Observe the reward $r'$;
    Select an action $a'$ using $\varepsilon$-greedy policy;
    Compute the weight of importance sampling:
        $w(s,s') = \begin{cases} \varepsilon(s)/\hat{\varepsilon}(s) & \text{if } s' \in \mathcal{D}, \\ (1-\varepsilon(s))/(1-\hat{\varepsilon}(s)) & \text{if } s' \in \mathcal{D}^c; \end{cases}$
    Compute the error:
        $\triangle = w(s,s')\left(r' + \gamma \hat{Q}^{\pi}(s',a')\right) - \hat{Q}^{\pi}(s,a);$
    Update the value estimates:
        $\hat{Q}^{\pi}(s,a) \leftarrow \hat{Q}^{\pi}(s,a) + \alpha e(s,a) \triangle;$
    if $s' \in D$ then
        $\hat{Q}^{\pi}_{\mathcal{D}}(s,a) \leftarrow (1-\alpha)\hat{Q}^{\pi}_{\mathcal{D}}(s,a) + \alpha \varepsilon(s)\left(r' + \gamma \hat{Q}^{\pi}(s',a')\right),$
    else
        $\hat{Q}^{\pi}_{\mathcal{D}}(s,a) \leftarrow \hat{Q}^{\pi}_{\mathcal{D}}(s,a);$
    end if
    Update disaster probabilities:
        $\hat{\varepsilon}(s) \leftarrow \min\left\{ \max\left\{ \delta, \frac{|\hat{Q}^{\pi}_{\mathcal{D}}(s,a)|}{|\hat{Q}^{\pi}(s,a)|} \right\}, 1-\delta \right\};$
    $s \leftarrow s'$, $a \leftarrow a'$;
end for
Output: $r'$, $\pi^*(a|s)$, $Q^{\pi}(s,a)$ and $\varepsilon^*(s)$.
```

The eligibility traces $e_t(s)$ of DEIS can be computed as

$$e_t(s) \leftarrow \begin{cases} 1 & \text{if } s = s_t, \\ \gamma\lambda w e_{t-1}(s) & \text{otherwise.} \end{cases} \tag{3.21}$$

In DEIS, $\hat{\varepsilon}(s)$ can be updated as

$$\hat{\varepsilon}(s) = \frac{|\hat{V}^{\pi}_{\mathcal{D}}(s)|}{|\hat{V}^{\pi}_{\mathcal{D}}(s)| + |\hat{V}^{\pi}_{\mathcal{D}^c}(s)|}. \tag{3.22}$$

The state values $\hat{V}^{\pi}_{\mathcal{D}}(s)$ and $\hat{V}^{\pi}_{\mathcal{D}^c}(s)$ are updated following the forms of Frank et al. (2008). If $s' \in \mathcal{D}$, then

$$\hat{V}^{\pi}_{\mathcal{D}}(s) \leftarrow (1-\alpha)\hat{V}^{\pi}_{\mathcal{D}}(s) + \alpha\varepsilon(s)\Big(r(s,a,s') + \gamma\hat{V}^{\pi}_{\mathcal{D}}(s')\Big), \tag{3.23}$$

Otherwise, then

$$\hat{V}^{\pi}_{\mathcal{D}^c}(s) \leftarrow (1-\alpha)\hat{V}^{\pi}_{\mathcal{D}^c}(s) + \alpha(1-\varepsilon(s))\Big(r(s,a,s') + \gamma\hat{V}^{\pi}_{\mathcal{D}^c}(s')\Big). \tag{3.24}$$

Since $\hat{V}^{\pi}_{\mathcal{D}}(s)$ and $\hat{V}^{\pi}_{\mathcal{D}}(s)$ are updated to the unbiased $V^{\pi}_{\mathcal{D}}(s)$ and $V^{\pi}_{\mathcal{D}^c}(s)$, respectively, disaster probabilities will converge to the optimal, that is $\hat{\varepsilon}(s) \rightarrow \varepsilon^*(s)$. In what it follows, $\hat{V}^{\pi}(s) \rightarrow V^{\pi}(s)$. Hence, we put forward Proposition 3.1 of convergence.

Proposition 3.1. *For every state s, with $\varepsilon^*(s) \in (\delta, 1-\delta)$, as $t \rightarrow \infty$, the estimated disaster probability $\hat{\varepsilon}(s)$ converges almost surely to the true disaster probability $\varepsilon^*(s)$.*

$$\hat{\varepsilon}(s) \rightarrow \varepsilon^*(s).$$

Furthermore, the estimate of the value function also converges almost surely to the unbiased value.

$$\hat{V}^{\pi}(s) \rightarrow V^{\pi}(s).$$

Algorithm 2 Disaster event adaptive importance sampling (DEIS)

```
Input: Policy $\pi$, rare event set $\mathcal{D}$, normal event set $\mathcal{D}^c$, true rare-event probabilities $\varepsilon(s)$, transition probabilities $f(s'|s)$, $g(s'|s)$, the learning rate $\alpha$, the discount rate $\gamma$, the greedy parameter $\varepsilon$ and the parameter $\delta > 0$.
Initialize state values $\hat{V}^\pi(s), \hat{V}^\pi_{\mathcal{D}}(s), \hat{V}^\pi_{\mathcal{D}^c}(s)$ arbitrarily, $\forall s$;
Initialize the rare event sampling distribution $\hat{\varepsilon}(s) = 0.5, \forall s$;
Initialize eligibility traces $e(s) = 0$;
Select the initial state $s$;
for each iteration do
    Update eligibility traces $e(s)$ in eq. (3.21);
    Select an action $a \sim \pi$;
    Decide whether a disaster event happens based on $\hat{\varepsilon}(s)$. If a disaster event is determined to occur, sample $s'$ from the disaster event transition distribution $g(s'|s)$. Otherwise, sample $s'$ from normal state transition distribution $f(s'|s)$.
    Observe the reward $r'$;
    Compute the weight of importance sampling:
        $$w(s,s') = \begin{cases} \varepsilon(s)/\hat{\varepsilon}(s) & \text{if } s' \in \mathcal{D}, \\ (1-\varepsilon(s))/(1-\hat{\varepsilon}(s)) & \text{if } s' \in \mathcal{D}^c; \end{cases}$$
    Compute the error:
        $$\triangle = w(s,s')\Big(r' + \gamma \hat{V}^\pi(s')\Big) - \hat{V}^\pi(s);$$
    Update the value estimates:
        $$\hat{V}^\pi(s) \leftarrow \hat{V}^\pi(s) + \alpha e(s) \triangle;$$
    if $s' \in \mathcal{D}$ then
        $\hat{V}^\pi_{\mathcal{D}}(s) \leftarrow (1-\alpha)\hat{V}^\pi_{\mathcal{D}}(s) + \alpha\varepsilon(s)\Big(r' + \gamma\hat{V}^\pi(s')\Big),$
    else
        $\hat{V}^\pi_{\mathcal{D}^c}(s) \leftarrow (1-\alpha)\hat{V}^\pi_{\mathcal{D}^c}(s) + \alpha(1-\varepsilon(s))\Big(r' + \gamma\hat{V}^\pi(s'\Big);$
    end if
    Update the disaster probabilities:
        $$\hat{\varepsilon}(s) \leftarrow \min\left\{\max\left\{\delta, \frac{|\hat{V}^\pi_{\mathcal{D}}(s)|}{|\hat{V}^\pi_{\mathcal{D}}(s)| + |\hat{V}^\pi_{\mathcal{D}^c}(s)|}\right\}, 1-\delta\right\};$$
    $s \leftarrow s'$;
end for
Output: $r'$, $V^\pi(s)$ and $\varepsilon^*(s)$.
```

3.3 Empirical results

This section assesses to what extent robo-advisors surpass the performance of human advisors and the stand-alone investor approach. We first introduce the used dataset, which contains rare disaster events and normal events. We then discuss our analysis of results on robo-advising using the proposed algorithm against the benchmarked methods.

3.3.1 Data

To simplify the analysis without loss of generality, we consider an MDP with three states: high-volatility, low-volatility, and disaster. Our classification of high and low volatility states is based on the financial markets (Alsabah et al., 2021). Data for high and low volatility states are extracted from the NBER and Vanguard's economic and market outlook reports (Alsabah et al., 2021). We collect data of S&P 500 from Yahoo Finance and the periods include the targeted two disaster events: the financial crisis in 2008 and the recent COVID-19 outbreak.[5]

The key parameters of the risky asset are specified for each $s \in \mathcal{S}$. Each state is characterized by a tuple of the first and the second moment of the market returns. The historical monthly returns of S&P 500 in the period from January 2008 to December 2008 are used for the 2008 financial crisis. Regarding the COVID-19 outbreak, the monthly returns of S&P 500 in the first quarter of 2020 are extracted. Finally, we pin down the average returns 1.25%, 0.5%, -3% and -2.75% for the high volatility state, low volatility state, the financial crisis, and the COVID-19 outbreak, respectively. We assign 5% (high volatility), 3% (low volatility), 6% (the 2008 financial crisis), and 13.52% (the COVID-19 pandemic) volatility levels for each considered state. We choose the monthly treasury yield rate which is 0.2% as the proxy for the risk-free rates following Alsabah et al. (2021).[6] Robo-advisors are assumed to make investment decisions every month, so we use monthly data and parameters. Robo-advisors run iteratively from episode to episode. Each episode has 75 time steps (months), the average length of the US business cycle between 1945 and 2020.[7]

[5] Source: Yahoo Finance: https://finance.yahoo.com/quote/5EGSPC/history/

[6] Alsabah et al. (2021) collect the monthly treasury rate as on May 16, 2019 from the US Department of the Treasury: https://home.treasury.gov/policy-issues/financing-the-government/interest-rate-statistics.

[7] We update the business cycle data from the NBER: https://www.nber.org/research/business-cycle-dating.

The set of possible risk aversion parameters for retail investors is $\{2.2, 2.3, \ldots, 8.3\}$ (Alsabah et al., 2021). Lower risk aversion corresponds to a higher portfolio weight allocated to the risky asset. Since this paper does not examine the risk profiles and investment behaviours of investors, we use the average risk aversion value 5.25 from this set as the predefined value for the investor's risk aversion θ.

3.3.2 Analysis of results

There are two simulations that contain high volatility states, low volatility states and disaster states. The difference between the two simulations is that Simulation 1 includes the 2008 financial crisis as the disaster state, while Simulation 2 treats the COVID-19 pandemic as the disaster state.

We begin by examining the optimal policy that maximizes investor utility in the presence of rare disasters. The optimal policy is obtained by implementing policy optimization according to the algorithm 1. In table 3.1, we tabulate the constellation of the optimal action over a state space. Table 3.1 compares the optimal action at each state between SARSA-IS and SARSA at the 500^{th} episode. The policy generated by SARSA serves as a benchmark. This episode is considered to have converged for both simulations in the presence of rare disasters. The optimal action represents the ideal allocation of the risky asset in the portfolio by robo-advisors. For instance, if the robo-advisor employs SARSA-IS in Simulation 1, the most favorable action in response to the disaster state is to decrease the allocation of the risky asset to 0.01, and seek safety by investing the remaining wealth in the risk-free asset under the disaster state. In Simulation 2, the optimal allocation remains at 0.01 in the disaster state. Notably, the allocations are lower than the benchmark in all states. This observation suggests that SARSA-IS produces a conservative policy with regard to rare disasters. These findings indicate that robo-advisors utilizing SARSA-IS can adapt their investment strategies to better account for the impact of rare disasters. By adopting a more conservative approach, robo-advisors are better equipped to protect investors' wealth during periods of extreme market turmoil.

Figure 3.2 shows the average rewards over 500 episodes for SARSA-IS and SARSA, and the investor-only approach, respectively. A stand-alone investor approach assumes that in-

Table 3.1 The estimated optimal action

States	Simulation 1		Simulation 2	
	SARSA-IS	SARSA	SARSA-IS	SARSA
High return volatility state	0.5	0.58	0.24	0.3
Low return volatility state	0.23	0.87	0.59	0.69
Disaster state	0.01	0.12	0.01	0.14

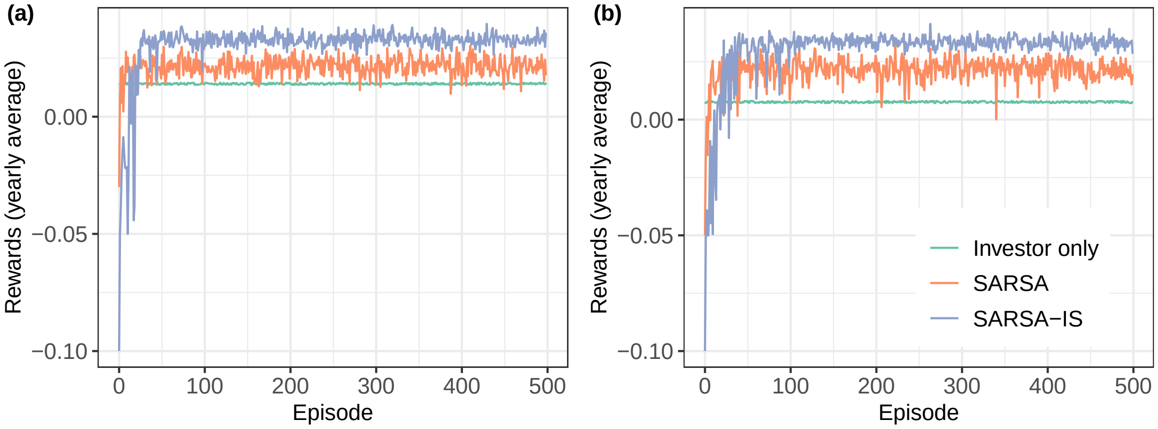


Fig. 3.2: Reward estimation of: (a) Simulation 1; (b) Simulation 2

vestors select a portfolio independently and allocate their wealth to risky assets depending on risk preference. They are supposed to maximize their utilities but are subject to behavioural biases and investment mistakes. On the one hand, we assume that a rational investor acts in accordance with the principle of a mean-variance optimizer (Markowitz, 1952). On the other hand, rationality may not be upheld, as in reality, investors are subject to investment mistakes, information cost (Campbell, 2006), behaviour biases, and cognitive limitations (Foerster et al., 2017). Such behaviour biases lead to a disparity (the investment gap) between the optimization and observed portfolios. Based on the US Survey of Consumer Finances dataset, the gap size is estimated to be around 0.073%. Thus, the investor-only approach considers mean-variance optimization and this disparity gap. The performance of an investor-only approach, as shown in figure 3.2 reflects these two factors.

Several findings merit emphasis. First, SARSA-IS maintains a consistently higher average annual reward than the benchmarked methods in the long run (over 100 episodes). This superior performance is attributed to the adoption of importance sampling techniques, which assign a higher importance weight to the outcomes corresponding to the identified disasters. Furthermore, compared to SARSA, the reward generated by SARSA-IS exhibits lower fluctuations over time. It has been observed that SARSA-IS converges more rapidly than SARSA due to the intensive sampling of disaster events, which enhances the policy improvement process. Subsequently, the gap between SARSA-IS and SARSA remains bounded, with SARSA-IS persistently outperforming both SARSA and the investor-only approach.

In comparison to the robo-advisors, the rewards of the investor-only approach fall significantly short of those achieved by SARSA-IS, demonstrating that stand-alone investors are unable to manage their portfolios as effectively as robo-advisors.

The results in figure 3.2 implicate that the adoption of robo-advisors, particularly those implementing SARSA-IS, can enhance portfolio performance by mitigating the impacts of ambiguity and uncertainty. By utilizing a systematic and data-driven approach, robo-advisors can address the challenges faced by individual investors, resulting in improved investment decision-making and more efficient risk management. Additionally, the reduced fluctuations exhibited by SARSA-IS indicate that this approach may be especially beneficial in handling rare disaster events, potentially offering a more robust solution for investors during turbulent market conditions.

Optimizing the probability of a rare event is a challenging task in robo-advising. According to Proposition 3.1, if the number of learning episodes is sufficiently large, the estimated probability of a rare event converges almost surely to the optimal one, with $\hat{\varepsilon}(s) \to \varepsilon^*(s)$. Figure 3.3 illustrates the estimated probability of a rare event $\hat{\varepsilon}(s)$ for the two targeted disasters. The optimal disaster event probability estimated by SARSA-IS reaches 0.1 after 400 episodes. Prior to converging on an optimal probability, importance sampling is effective at the sampling stage, and SARSA-IS favours the sampling of rare events. As the number of episodes increases, $\hat{\varepsilon}(s)$ converges to the optimal $\varepsilon^*(s)$ at a rate of 0.10. This convergence reconciles with the true probability of the disaster event $\varepsilon(s)$, which is set at 0.10. The implication emphasizes the value of using importance sampling in robo-advising, particularly when estimating the probability of rare disaster events. By effectively sampling rare events and converging to the optimal probability more quickly, SARSA-IS can provide a more accurate and reliable estimation of disaster event probabilities. This, in turn, allows for better risk management, improved investment decision-making, and potentially more robust portfolio performance during times of market turmoil.

We now turn our focus to policy evaluation. In the policy evaluation process, we compare the

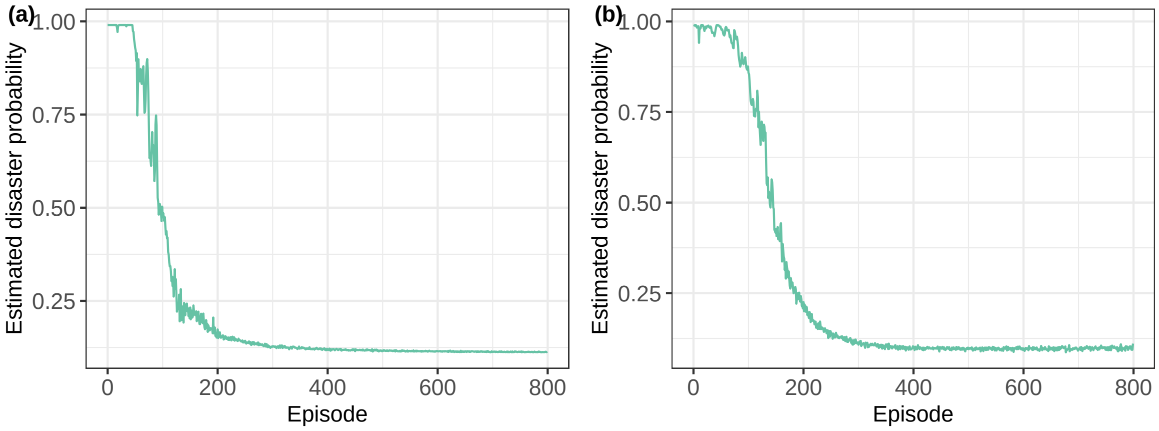


Fig. 3.3: Estimated disaster probability ($\hat{\varepsilon}(s)$) of: (a) Simulation 1; (b) Simulation 2

performance of policies using the algorithm DEIS (Frank et al., 2008). Our approach has at least two advantages. First, regarding rare disasters, enhanced learning by importance sampling overcomes learning bottlenecks caused by the low probability of occurrence. Second, using the proposal distribution substantially reduces the variance of value function estimates.

Figure 3.4 presents value function estimates for Simulation 1 with the 2008 financial crisis, and Simulation 2 with the COVID-19 outbreak. To show that DEIS is a sensible method for policy evaluation, we compare its performance with TD learning. As expected, the latter exhibits a higher fluctuation before convergence, even though it gets close to DEIS by the end. TD learning may take longer to converge to the optimal one, and sometimes, this convergence is unfeasible in practice (Frank et al., 2008; Dann et al., 2014). After a 2000-round of iteration, we conclude that for an evaluated policy π^*, both converge to 2.64 in Simulation 1 and 2.45 in Simulation 2. For the policy π, a convergence value at 1.67 (2008 simulation) and 1.92 (COVID-19 simulation) are presented. Overall, empirically we are able to support $\hat{V}^{\pi}(s) \rightarrow V^{\pi}(s)$ using DEIS, as $t \rightarrow \infty$. This result supports the proposition 3.1. It is worth noting that in the presence of rare disasters, TD learning exhibits higher volatility, resulting in errors in value estimation. To circumvent this risk and the associated cost, the robo-advising industry may consider adopting DEIS for policy evaluation tasks.

With DEIS, the robo-advising industry benefits from precisely estimating investors' utilities. Proper estimations enable robo-advisors to provide customized consulting services and tactical strategies to investors, addressing their unique needs and preferences. Such efforts help alleviate clients' concerns and restore their confidence in investment decision-making (Capponi et al., 2022). The value estimate under policy π^* generated from SARSA-IS consis-

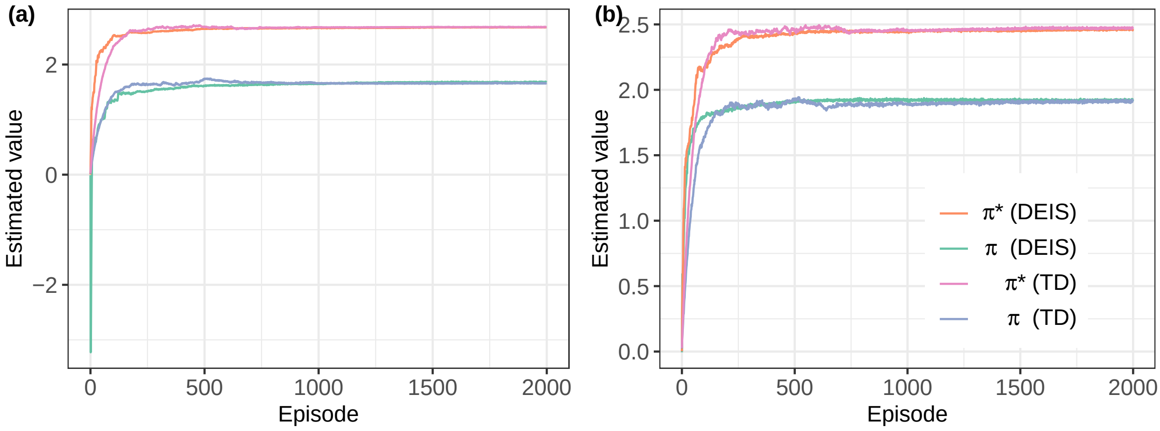


Fig. 3.4: Value estimation of: (a) Simulation 1; (b) Simulation 2. π^* is the optimal policy generated by SARSA-IS in the algorithm 1, while π is the benchmark policy updated by SARSA, π^*(DEIS) and π(DEIS) are the algorithm 2 DEIS following the optimal policy π^*, the benchmark policy π, respectively. π^*(TD) and π(TD) are TD learning following the optimal policy π^*, the benchmark policy π, accordingly.

tently surpasses that of policy π generated from SARSA. Both figures demonstrate that the optimal policy π^* derived from SARSA-IS outperforms the benchmark policy π obtained from SARSA. Moreover, investors can experience higher utility under the desired policy generated from SARSA-IS compared to a policy generated from SARSA.

The implications emphasize the superiority of SARSA-IS in generating policies that maximize investor utility. By providing higher utility, SARSA-IS can help robo-advisors deliver more effective investment strategies, thus fostering better portfolio performance and client satisfaction. In a competitive industry like robo-advising, adopting advanced techniques like SARSA-IS and DEIS may contribute to retaining and attracting clients, ultimately leading to a more sustainable and successful business.

To examine the effectiveness of our proposed solutions when confronted with rare disasters in real-world scenarios, we compare the quarterly portfolio returns following our policies to the actual quarterly returns of “real” robo-advisory portfolios in Q1 2020. The term of “real” robo-advisory portfolios refers to the investment strategies employed by well-known robo-advisors, as detailed in the 2020 First Quarter Robo Report[8]. These portfolios typically utilize diversified algorithmic strategies that allocate assets across a range of investment vehicles, including a mix of risky and risk-free assets, tailored to meet the risk preferences and financial goals of individual investors.

[8] For the data and details of robo-advisory portfolios, please see: https://storage.googleapis.com/gcs.backendb.com/wordpress/media/2021/02/2020-Q1-Robo-Report.pdf

The comparison between our proposed portfolio, which comprises one risky asset and one risk-free asset, and “real” robo-advisory portfolios is justified through the lens of Modern Portfolio Theory (Markowitz, 1952) and the principles underlying robo-advisory services. Although our model utilizes a simplified asset universe, the objective remains consistent with that of the “real” robo-advisors to optimize the risk-return profiles of portfolios based on predefined risk preferences and market conditions. While the “real” robo-advisory portfolios are potentially more complex and diversified, they are fundamentally designed to balance risk and return by allocating assets across various investment vehicles. Our two-asset model captures the essential dynamics of this process and allows us to evaluate the effectiveness of the SARSA-IS policy in mitigating extreme risks. Furthermore, by focusing on the performance during a disaster period, we highlight the robustness of our approach in scenarios where traditional robo-advisory strategies may falter due to insufficient emphasis on rare but impactful events in reality.

As illustrated in figure 3.5, the portfolio return adhering to the policy π^* generated from SARSA-IS significantly exceeds that of the policy π generated from SARSA, as well as the robo-advising portfolio returns from the 2020 First Quarter Robo Report[9] during the onset of the COVID-19 pandemic. Specifically, we observe that the estimated quarterly return utilizing the optimal policy π^* generated from SARSA-IS is 0.512%, representing a remarkable performance when compared to -0.639% (generated by the benchmark policy from SARSA) and -14.55% (the average quarterly return of existing robo-advising portfolios). All returns of existing robo-advising portfolios were obviously negative, while our policy generated from SARSA-IS ensured a positive return during the same disaster period.

Employing SARSA-IS, the strategy of robo-advising manifests resilience and stability even when confonted with rare disasters. Figure 3.5 highlights the significant economic loss experienced by the robo-advising industry during 2020 Q1. By implementing policy π^* generated from SARSA-IS, robo-advisors alleviate the dramatic impact of rare disasters. To sum up, our solutions optimize investment strategies employed by robo-advisors and enhance the performance of their portfolios, even under challenging circumstances.

[9] For the data and details of robo-advising portfolios, please see: https://storage.googleapis.com/gcs.backendb.com/wordpress/media/2021/02/2020-Q1-Robo-Report.pdf

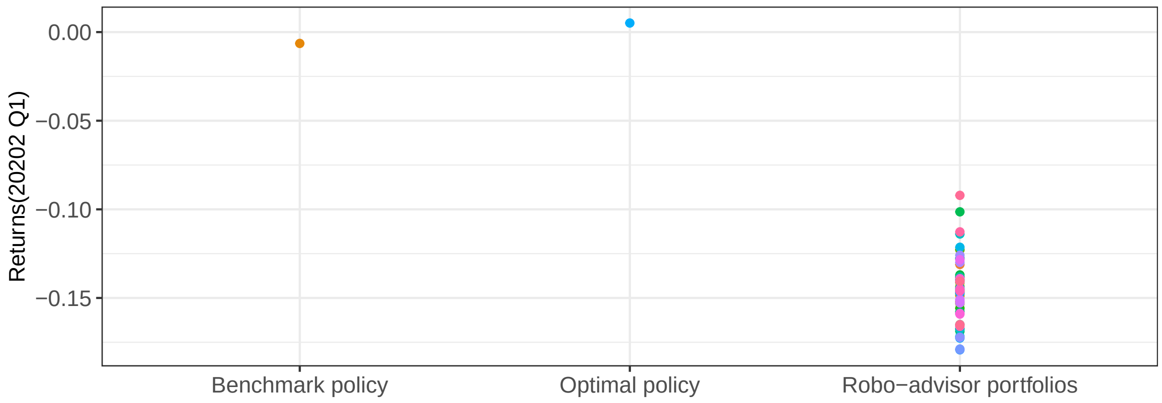


Fig. 3.5: Comparison of portfolio returns. The blue dot represents the optimal SARSA-IS policy, the orange dot is for the benchmark SARSA policy, and the dots aligned along a vertical line depict various ”real” robo-advisory portfolios.

3.4 Conclusion

Robo-advisors have effectively transferred human-involved investment decisions into automated and AI-driven solutions. The advantages and potential of AI-powered portfolio management are substantial. However, during the COVID-19 outbreak, robo-advisors faced challenges in maintaining satisfactory performance, particularly under rare disasters. Empirical evidence indicates that robo-advisors underperformed during the recent pandemic, emphasizing the need for a novel robo-advising algorithm that ensures reliability and robustness.

In this study, we develop a novel framework that enables robo-advisors to learn from rare but emphasized samples and maximize cumulative rewards in portfolio optimization. By considering the mean-variance approach in asset allocation, robo-advisors can assign appropriate asset weights to maximize returns. We propose SARSA-IS to search for the optimal policy and DEIS for policy evaluation under rare disasters. Robo-advisors, empowered by SARSA-IS, significantly improve investors’ utilities and rewards due to the effective identification of optimal policies. Our approach diverges from existing research focusing on robo-advising adoption during COVID-19, contributing to the robo-advising domain by examining the impact of rare disasters and offering pertinent solutions. By employing importance sampling techniques, our algorithm effectively learns from rare disasters. DEIS, combined with importance sampling, exhibits lower variance than traditional TD learning methods in value function estimation. Our study suggests that robo-advisors, proficient in learning investors’ utilities, can provide reliable financial services.

To the best of our knowledge, our research is among the first to analyze robo-advising in the context of rare disaster events. Our work can be specialized in portfolio management related to rare disasters, proving effective in learning from rare but crucial events and solving optimization problems within low-probability state spaces. We classify financial markets into three states for simplicity and ease of interpretation, although our analysis can be extended further. In practice, numerous features can contribute to defining a state. Future research will involve identifying key features for high-dimensional data and expanding the size of states to explore the set of optimal policies in more complex scenarios. We leave this extension for future endeavours.

Appendices to chapter 3

3.A SARSA

For the reader's convenience, this appendix provides the technical details about SARSA in the context of robo-advising.

Algorithm 3 State–Action–Reward–State–Action (SARSA)

```
Input: Rare event set $\mathcal{D}$, normal event set $\mathcal{D}^c$, true rare-event probabilities $\varepsilon(s)$, the learning rate $\alpha$, the discount rate $\gamma$ and the greedy parameter $\varepsilon$.
Initialize state action values $\hat{Q}^{\pi}(s,a)$ arbitrarily, $\forall s,a$;
Initialize eligibility traces $e(s,a) = 0$;
Select the initial state $s$;
Select an action $a$ from $s$ using $\varepsilon$-greedy policy;
for each iteration do
    Take an action $a_t$;
    Sample $s'$ from the transition distribution $p(s'|s)$;
    Observe the reward $r'$;
    Select an action $a'$ from $s'$ using $\varepsilon$-greedy policy;
    Compute the error:
        $\triangle = r' + \gamma\hat{Q}^{\pi}(s',a') - \hat{Q}^{\pi}(s,a)$
    Update the value estimates:
        $\hat{Q}^{\pi}(s,a) \leftarrow \hat{Q}^{\pi}(s,a) + \alpha e(s,a)\triangle$
    Update eligibility traces $e(s,a)$ in eq. (3.19);
    $s \leftarrow s'$, $a \leftarrow a'$
end for
Output: $a$, $r'$ and $\hat{Q}^{\pi}(s,a)$.
```

3.B Proof of the optimal disaster probability function

According to Frank (2009), there is an optimal form of transition probability that enables the zero-variance of Q value under disaster events, given an MDP with states, actions, one-step transition probability and reward function. The optimal transition probability is:

$$p^*(s'|s,a) = \frac{p(s'|s,a)\sum_{a'\in A}\pi(a'|s')\left[r(s,a,s') + \gamma Q^\pi(s',a')\right]}{Q^\pi(s,a)}. \tag{3.25}$$

Developed from eq. (3.25), one can obtain the optimal form of disaster probability as shown in eq. (3.13). That is:

$$\sum_{s'\in S} p^*(s'|s,a) = \sum_{s'\in S} \frac{p(s'|s,a)\left[r(s,a,s') + \gamma \sum_{a'\in A}\pi(a'|s')Q^\pi(s',a')\right]}{Q^\pi(s,a)}.$$

As

$$\sum_{s'\in S} p(s'|s,a) = \varepsilon(s)\sum_{s'\in \mathcal{D}} g(s'|s) + (1-\varepsilon(s))\sum_{s'\in \mathcal{D}^c} f(s'|s,a),$$

we have

$$\varepsilon(s)^*\sum_{s'\in \mathcal{D}} g(s'|s) = \frac{\varepsilon(s)\sum_{s'\in\mathcal{D}} g(s'|s)\sum_{a\in\mathcal{A}}\pi(a'|s')\left[r(s,a,s') + \gamma Q^\pi(s',a')\right]}{Q^\pi(s,a)},$$

for all $s'\in\mathcal{D}$. As $\varepsilon(s)^*$ is a constant for s',

$$\varepsilon(s)^*\sum_{s'\in \mathcal{D}} g(s'|s) = \varepsilon(s)^*.$$

Therefore, we obtain the optimal form of disaster probability, which is:

$$\begin{aligned}\varepsilon(s)^* &= \varepsilon(s)^*\sum_{s'\in \mathcal{D}} g(s'|s)\\ &= \frac{\varepsilon(s)\sum_{s'\in\mathcal{D}} g(s'|s)\left[r(s,a,s') + \gamma\sum_{a'\in A}\pi(a'|s')Q^\pi(s',a')\right]}{Q^\pi(s,a)}.\end{aligned}$$

3.C Proof of the disaster state-action value function

The relationship between $Q^\pi(s,a)$ and $Q^\pi_D(s,a)$ is:

$$Q^\pi_D(s,a) = \varepsilon(s)Q^\pi(s,a).$$

$Q^{\pi}(s,a)$ can be evaluated by the Bellman Q function eq. (3.3). Replace $Q^{\pi}(s,a)$ with $\frac{Q_D^{\pi}(s,a)}{\varepsilon(s)}$ and thus, one have:

$$\frac{\hat{Q}_D^{\pi}(s,a)}{\varepsilon(s)} \leftarrow \frac{\hat{Q}_D^{\pi}(s,a)}{\varepsilon(s)} + \alpha[r(s,a,s') + \gamma\hat{Q}^{\pi}(s',a') - \frac{\hat{Q}_D^{\pi}(s,a)}{\varepsilon(s)}],$$

that can be reformulated as:

$$\hat{Q}_D^{\pi}(s,a) \leftarrow (1-\alpha)\hat{Q}_D^{\pi}(s,a) + \alpha\varepsilon(s)\Big(r(s,a,s') + \gamma\hat{Q}^{\pi}(s',a')\Big).$$

Therefore, we complete the proof of eq. (3.18).

Chapter 4

Risk aversion and portfolio optimization for robo-advising

Abstract

We develop a novel framework for learning investors' risk aversion using low-resolution data, a common issue arising from short trajectories recording investors' portfolio choices, particularly during disaster events. Furthermore, the observed portfolio choice is often affected by behavioural biases. Our approach combines online inverse optimization with deep RL to simultaneously estimate risk aversion and determine optimal investment strategies under both normal and disaster states. Utilizing real mutual fund data, we demonstrate that our algorithm's risk aversion estimation converges asymptotically to the optimal risk aversion during the learning process. Critically, based on the learned risk aversion and trained deep RL model, we show that robo-advisors adopting our approach can effectively tailor investment strategies to suit investor risk aversion under varying market conditions, outperforming traditional funds. This highlights the potential for our framework to enhance investment decision-making and better represent investor interests in both stable and volatile market environments.

Keywords: robo-advising; risk aversion; economic disasters; inverse optimization; deep reinforcement learning

4.1 Introduction

Robo-advising are digital platforms that utilize algorithms to build and manage clients' investments automatically. They have attracted lots of investors' interest because of their substantive benefits, such as being cheaper, more efficient and more transparent than traditional financial services. They are able to diversify investors' portfolios based on economic models, and algorithms. However, research argues that robo-advisors are not capable enough to build portfolios tailored to investors' risk profiles (Tertilt and Scholz, 2018; Alsabah et al., 2021; Dong et al., 2022). Although the importance of risk aversion is well documented, in practice, the assessment on risk aversion tends to be hard due to the subjective nature of risk-taking such as inherent behaviour biases, and investment errors.

A strand of research studies reveals investors' risk aversion in two main areas. The first area is the questionnaire-based methods that measure investors' risk preferences directly from a set of questions. These methods develop questionnaires to construct risk-tolerance indices and measure financial risk tolerance (Grable, 2000). Using large-scale representative surveys, Dohmen et al. (2011) discover the determinants of risk attitudes and explain the risk-taking behaviours. In practice, robo-advisors mainly use online questionnaires to evaluate investors' risk profiles (Tertilt and Scholz, 2018). The second area infers risk aversion from investors' observed investment behaviours such as household portfolios or the investors' portfolio choices. Bucciol and Miniaci (2011) derive the distribution of risk tolerance from the U.S. household samples. Alsabah et al. (2021) learn an investor's risk preference through virtual investor's portfolio choices. Their study inversely estimates a parameter of the risk preference based on the known market model, e.g. mean-risk utility model, as well as the observable information from the market and investors. Following Alsabah et al. (2021), Dong et al. (2022) estimate the risk aversion using Chinese stock market data. Yu et al. (2023) infer an investor's risk preference and expected returns directly from historical portfolio allocation data using inverse optimization on the mean-variance portfolio model.

The existing literature about assessing risk aversion of robo-advisory faces several limitations. Firstly, when using questionnaires to assess investors' risk aversion, cognitive biases may lead to imprecise results as the participants' choices do not necessarily reflect their true risk preferences without experiencing real-life situations (Tertilt and Scholz, 2018). Additionally, robo-advisory questionnaires often lack comprehensive financial-related questions to fully understand investors' risk profiles. Secondly, although studying portfolio choices can be more accurate and comprehensive in revealing risk preferences as they examine observed behaviours, these methods may only asymptotically converge to the true risk preference when data is sufficiently large or representative. This is particularly problematic in rare economic disasters where obtaining comprehensive data is challenging. Third, the current approach of inverse optimization assumes that risk aversion is time-varying, relying on the real-time market data and previously updated risk aversion (Yu et al., 2023). This approach is sensitive to the market movement resulting in an unstable estimator. The evidence is reflected in the empirical results of big jumps in their estimated risk aversion under economic recession.

To solve the aforementioned limitations, we present a novel approach to learning investors' risk aversion and optimizing investment strategies by combining inverse optimization and deep RL in two state spaces including normal states and disaster states. Our framework is developed using several advanced methods, including mean-variance optimization, inverse optimization and deep RL algorithms. Since the risk aversion parameters may not converge to the true one with limited observations of holdings for batch inverse optimization, online inverse optimization updates iteratively risk aversion if the observed holdings are available in the recorded trajectory until convergence. The updated risk aversion will be used to learn the optimal investment by robo-advisors via deep RL continuously. Robo-advisors act as human-acid investment advisors that not only learn the dynamic of economic states but also provide investment strategies tailored to investors' risk aversion. Continuous portfolio optimization after risk aversion is learned, which helps avoid huge potential losses in disaster states and obtain sustainable profits in normal states.

RL is a sub-field of machine learning that focuses on how an agent can learn to make decisions through interactions with the environment. RL improves learning based on experience and errors, where the agent receives feedback on its actions in the form of rewards or penalties (Sutton and Barto, 1998). The agent learns to optimize its policies based on these rewards, maximising the expected cumulative rewards in the long run. Using converging risk aversion values in two states, we test the trained model in the testing set, which covers both disaster states and normal states. Results show the proposed approach successfully beat the real mutual fund performance and benchmark portfolios in the long term.

The framework greatly improves how we estimate risk aversion based on different economic conditions. Traditional methods, which use a fixed measure of risk aversion, don't accurately reflect the real-world fluctuations of the market. Unlike these static models, our framework adjusts to changes, recognizing that people's levels of risk aversion can differ greatly during normal times and in crisis situations. However, trying to keep risk aversion estimates up-to-date can be challenging. They often need to be adjusted due to changes in both market conditions and investor behaviours. Our approach uses inverse optimization to update these estimates in two distinct state spaces that deal with economic changes.

We select hybrid funds to investigate the risk aversion of funds because this category of funds actively invests in diversified asset classes and, therefore, exists a thorough asset allocation process. Through testing on three types of hybrid funds with different allocation strategies, the results illustrate that our proposed algorithms not only successfully estimate risk aversion, but also outperform the benchmarks and real mutual fund performance in investment.

This paper contributes to the revealed preference literature by inferring investors' risk aversion from observed portfolio choices. Our approach is connected to Hansen and Singleton (1982), who estimate preference parameters within a nonlinear rational expectations framework. They utilize aggregate consumption and asset returns to infer the agent's risk preferences. While their focus was on macroeconomic data and aggregate consumption patterns, we extend it to investors' portfolio choices in financial markets. Similarly, Chetty (2006) develops a method to estimate the coefficient of relative risk aversion from labour supply data. His work demonstrates how individual behavioural responses to wage and income changes reveal underlying risk preferences. Koijen and Yogo (2015) present an equilibrium model of institutional demand and asset prices, using stock market data to understand how institutional investors' preferences influence asset prices. Following their research, we use institutional data, which is the mutual fund holdings, to infer risk aversion.

Our work diverges from the revealed preferences' studies by employing mean-variance optimization within an inverse optimization framework to infer risk aversion from observed portfolio choices. Unlike Hansen and Singleton (1982), who utilize aggregate consumption data, and Chetty (2006), who focuses on labor supply, we apply our methodology directly to financial portfolio allocations. While Koijen and Yogo (2015) analyze institutional demand, we focus on institutional holdings in the robo-advisory context. By using mean-variance optimization, we are able to estimate state-dependent risk aversion parameters. The results reveal that investors' risk aversion increases during normal state spaces or disaster state spaces. This finding contributes to the literature by providing empirical evidence of how risk aversion is inferred from financial data adapted to changing market conditions, thereby extending the revealed preference approach to the domain of portfolio optimization and robo-advisory services.

Additionally, our research presents several crucial contributions. Firstly, this study is the pioneering effort to estimate risk aversion and optimize investment concurrently by proposing an iterative update algorithm between online inverse optimization and deep RL. Secondly, we advance the optimization of investment strategies that cater to investors' unique risk aversion preferences. Lastly, our work improves the estimation of risk aversion dependent on normal and disaster state spaces, offering a framework with the potential for extension to a broader range of states.

The study is presented in the following order. Section 4.2 presents the mean-variance optimization model to optimize asset weights conditional on appointed risk aversion, market signals and states. Section 4.3 propose inverse optimization. Section 4.4 discusses the RL methods. Then, section 4.5 outline the data and section 4.6 illustrates results. Finally, section 4.7 concludes the whole study.

4.2 Portfolio selection

4.2.1 Markowitz mean-variance optimization

The proposed methodology starts with finding the optimal portfolio weights, which is the so-called portfolio selection. Investors choose fractions $x = (x_1, x_2, ..., x_n)^\intercal$ to allocate in n assets over a certain period subject to constraints $e^\intercal x = 1$ where e is a vector of ones, corresponding to the number of assets n to ensure that the sum of the portfolio weights x equals to one (Markowitz and Todd, 2000). We consider the mean-variance portfolio optimization (Markowitz, 1952). That is:

$$
\begin{aligned}
\max_{x} \quad & \mu^{\intercal} x - \frac{1}{2}\rho x^{\intercal}\Sigma x, \\
s.t. \quad & x \geq 0, \\
& e^{\intercal} x = 1,
\end{aligned}
\tag{4.1}
$$

where ρ is the risk aversion parameter to balance the expected portfolio return and the portfolio variance term. The expected portfolio return $\mu^{\intercal} x$ is the weighted average of the expected asset returns $\mu \in \mathbb{R}^{n}$. The variance of portfolio returns $x^{\intercal}\Sigma x$ measures the degree of fluctuation due to the actual returns of the portfolio assets, where $\Sigma \in \mathbb{R}^{n\times n}$ is the covariance matrix between assets.

We assume there is no short sell allowed, in equivalent, $x \geq 0$. The model eq. (4.1) is unable to derive analytical solutions due to inequality constraints (Best and Grauer, 1991), but we can use numerical solutions to optimize the optimal weights from the quadratic optimization function as shown in eq. (4.1).

To achieve effective portfolio diversification in asset allocation for hybrid funds, which typically hold hundreds of distinct assets, it is essential to categorize these holdings into broader financial asset classes, including equities, fixed-income, and cash. This approach not only simplifies the analysis but also aligns with the common investment strategies employed by fund managers. The equity asset class encompasses various types of tradable equity assets, such as common shares, and preferred stocks, found in hybrid funds. Research has shown that equities generally provide higher returns over the long term, albeit with increased volatility (Fama and French, 1992). This makes them an attractive option for investors seeking growth potential. Fixed-income assets, which include sovereign bonds, corporate bonds, mortgages, and asset-backed securities, offer more predictable income streams and lower risk compared to equities (Fabozzi and Fabozzi, 2021). As such, they serve as an essential component in a diversified portfolio, particularly for conservative investors or those nearing retirement. The cash asset class comprises cash and cash equivalents held by fund companies, providing liquidity and a buffer against short-term market fluctuations. Holding cash can help investors manage risk and capitalize on investment opportunities as they arise. Moreover, we exclude other asset types, such as futures and swaps, from our analysis as they constitute an insignificant portion of hybrid fund portfolios.

Researchers often utilize proxy indices to represent asset classes for reasons of data availability and simplification. Acquiring data on specific illiquid and short-lived holdings can be challenging throughout the entire study period. Additionally, determining the historical returns of each asset class from a diverse range of component assets with varying characteristics proves difficult, as these encompass numerous fluctuating returns over time. One solution to this issue is the use of proxy assets. Karoui (2013) employs the stock market factor from the Kenneth French library as a proxy for stock class returns and the total return index on 7 to 10-year U.S. Treasury bonds to represent bond returns. Their study assesses the performance of 182 hybrid funds using these proxies. In line with their employment of proxy assets, our study adopts the S&P 500 stock index (SPX) as a proxy for the equity asset class, the FTSE US Broad Investment-Grade Bond Index (USBIG) as a representative for the fixed-income asset class, and the one-month Treasury bill rate as a proxy for the cash asset class. These indices provide a reliable representation of their respective asset classes.

4.2.2 Expected returns and covariance

The expected return of the S&P 500 index is calculated by the Implied Equity Risk Premium (ERP) summing a proxy risk-free rate. Damodaran (2019) provides the implied ERP calcualting from the price of the S&P 500 index, future cash flow and its growth rate[1]. It represents investors' expectations of the stock market. Thus, we treat the ERP as the expected excess returns of S&P 500 index. And its expected returns are the summation to risk-free returns which is the T bond rate (Tbond) from Damodaran (2019)'s measurement. Thus, the expected returns of stock index μ_s is formulated as:

$$\mu_s = \text{ERP} + \text{Tbond}. \tag{4.2}$$

Unlike traditional methods that often rely on historical data, the implied equity risk premium is forward-looking. This estimation method bases itself on the market's current expectations rather than actual future cash flows, enhancing its relevance and responsiveness to market sentiment and macroeconomic changes. Particularly in disaster states, where actual returns are low and infrequent, relying on historical data can lead to significant estimation errors.

[1] For detailed measture, please refer to Damodaran (2019).

Thus, using the implied ERP allows our expected return calculation to dynamically reflect these fluctuations and provide a more accurate, market-sensitive evaluation.

For the expected returns of the bond index, as the bond index is less volatile to the market conditions, such as the presence of economic disasters, we use the historical average of returns of the bond index as its expected returns. The returns of the bond index at time t are calculated as:

$$R_{b,t} = \frac{p_{b,t}}{p_{b,t-1}} - 1, \tag{4.3}$$

where p_t is the bond price at time t. Konno and Kobayashi (1997) add coupons of the individual bond to the bond prices in the calculation of bond returns. However, as the bond index does not pay coupons, we don't include the average coupon in the returns of bonds.

The expected returns of cash assets are equivalent to their actual returns, and these assets are not included in the covariance matrix. As per the definition of risk-free assets (Damodaran, 1999), they exhibit a characteristic of zero return variance, implying that their expected returns correspond to the actual returns of the risk-free asset.

Regarding the covariance, the Ledoit-Wolf shrinkage approach (Ledoit and Wolf, 2004a) is utilized to estimate expected covariance. The Ledoit-Wolf method is particularly beneficial as it moderates extreme coefficients within the covariance matrix, pulling these values towards more central estimates to reduce estimation risk. In our paper, this method specifically addresses the extreme coefficients that arise from financial assets prone to disaster risk, which are typically skewed due to their high exposure to such risks. By applying shrinkage, these distorted covariance values are adjusted to more neutral estimates, effectively mitigating the impact of extreme risks.

The foundation of this method is a sample covariance matrix, denoted as $\Sigma_{sample} = \mathbb{E}[(R - \mathbb{E}[R])(R - \mathbb{E}[R])']$, where R is a vector of historical returns for the stock index, bond index, and risk-free assets[2]. This matrix is adjusted towards a structured estimator, Σ_{struct}, defined

[2] The variance of the risk-free asset and its covariance to the stock index and the bond index should be set to zero, due to its risk-free attribute.

by a constant variance across all variables. This structured estimator is formulated as follows:

$$\Sigma_{struct} = \frac{\mathrm{tr}(\Sigma_{sample})}{n} I_n, \tag{4.4}$$

where $\mathrm{tr}(\Sigma_{sample})$ represents the trace of Σ_{sample} which is the sum of the diagonal elements, and I_n is an identity matrix of dimension $n \times n$. This formulation provides a simple and stable baseline of covariance variance matrix by setting all variables in the structured estimator with the same variance. The optimal shrinkage constant, denoted as δ, is determined by minimizing the expected quadratic loss between the actual covariance matrix and the shrinkage estimator $\hat{\Sigma}$ (Ledoit and Wolf, 2004b). The shrinkage covariance matrix Σ is expressed as a linear combination of two extremes that are the sample covariance matrix and the structured covariance matrix. The function of the shrinkage covariance matrix is:

$$\Sigma(\delta) = \delta \Sigma_{struct} + (1-\delta)\Sigma_{sample}. \tag{4.5}$$

4.3 Inverse optimization

4.3.1 Forward optimization

Before defining the inverse optimization problem, we generalize the mean-variance optimization eq. (4.1) to the form of forward optimization to understand the process of forward decision-making. In forward optimization, the decision maker minimizes an objective function $f(x,u,\rho)$ given a constraint function $g(x,u,\rho)$ by responding with the optimal solution x. The formula of the forward optimization problem (FOP) is:

$$\mathrm{FOP}(u,\rho) := \min_{x}\{f(x,u,\rho) | g(x,u,\rho) \leq 0\}, \tag{4.6}$$

where $x \in \mathbb{R}^n$ is the decision variable which is portfolio holdings, and $u \in \mathbb{R}^n$ is the external input variable, such as the asset prices or historical returns. $\rho \in \mathbb{R}$ is the risk aversion parameter. The feasible set to $\mathrm{FOP}(u,\rho)$ is:

$$\mathcal{X}(u,\rho) := \{x \in \mathbb{R}^n : g(x,u,\rho) \leq 0\}. \tag{4.7}$$

The optimal solution set to FOP(u,ρ) is:

$$\mathcal{X}^{\text{opt}}(u,\rho) := \underset{x}{\operatorname{argmin}}\{f(x,u,\rho) | x \in \mathcal{X}(u,\rho)\}. \tag{4.8}$$

We have the following assumptions to ensure that FOP is a convex optimization problem.

Assumption 4.1. *$f(x,u,\rho)$ are continuous in x,u,ρ, and convex in x for the fixed u,ρ. $g(x)$ is continuous and convex in x.*

When dealing with concave optimization problems, a common approach to solving them is by converting them into convex optimization problems eq. (4.6), which are typically more tractable due to their well-behaved properties in the field of inverse optimization. We take the opposite of the concave objective function, as the opposite concave function is the convex function. Given the mean-variance optimization problem eq. (4.1), the objective function can be reformulated to focus on the minimization of a linear combination of risk and negative expected returns. The objective function which is also the primal problem becomes:

$$\begin{aligned} \min_{x} \quad & \frac{1}{2}\rho x^{\mathsf{T}}\Sigma x - \mu^{\mathsf{T}} x, \\ s.t. \quad & x \geq 0, \\ & e^{\mathsf{T}} x = 1. \end{aligned} \tag{4.9}$$

Eq. (4.9) is deemed as a convex optimization problem, as it satisfies the assumption 4.1.

In the context of solving eq. (4.9) for the optimal x, the dual problem is derived to facilitate finding this optimal solution. The Lagrangian of the problem is given by:

$$L(x,\lambda,\nu) = \frac{1}{2}\rho x^{\mathsf{T}}\Sigma x - \mu^{\mathsf{T}} x - \lambda^{\mathsf{T}} x + \nu(1 - e^{\mathsf{T}} x), \tag{4.10}$$

where $\lambda \in \mathbb{R}^n$ is the vector of Lagrange multipliers associated with the non-negativity constraints $x \geq 0$, and $\nu \in \mathbb{R}$ is the Lagrange multiplier for the equality constraint $e^{\mathsf{T}} x = 1$. The saddle-point formulation of eq. (4.9) can be written as:

$$\min_{x} \max_{\lambda,\nu} L(x,\lambda,\nu). \tag{4.11}$$

The dual function is obtained by minimizing the Lagrangian with respect to x,

$$D(\lambda,\nu) = \min_x L(x,\lambda,\nu). \tag{4.12}$$

Thus, the dual problem is:

$$\begin{aligned}
\max_{\lambda,\nu} \quad & D(\lambda,\nu), \\
s.t. \quad & \nabla_x L(x,\lambda,\nu) = 0, \\
& \lambda \geq 0.
\end{aligned} \tag{4.13}$$

By controlling the Lagrange multipliers and setting the partial derivatives $\nabla_x L(x,\lambda,\nu) = 0$, $L(x,\lambda,\nu)$ is minimized with respect to x. We further derive the function of the zero partial derivatives as follows:

$$\begin{aligned}
\nabla_x L(x,\lambda,\nu) &= \rho\Sigma x - \mu - \lambda - \nu e \\
&= 0.
\end{aligned} \tag{4.14}$$

Through these calculations, one can estimate the appropriate λ and ν values, substituting back into eq. (4.13) to ultimately determine the optimal x.

Slater's condition assures that there exists at least one point x within the interior of the feasible set of the dual problem such that $x \geq 0$ and $e^\intercal x = 1$, there is a strong duality between the primal problem and dual problem. This strong duality confirms that the optimal solutions to both problems coincide, meaning that any optimal solution $x \in \mathcal{X}^{\text{opt}}(u,\rho)$ must satisfy the Karush-Kuhn-Tucher (KKT) conditions (Bertsekas, 1997). They are:

$$\begin{aligned}
&\rho\Sigma x - \mu - \lambda - \nu e = 0, && \text{(Stationarity)} \\
&x \geq 0, \quad e^\top x = 1, && \text{(Primal feasibility)} \\
&\lambda \geq 0, && \text{(Dual feasibility)} \\
&\lambda_i x_i = 0 \quad \text{for all } i. && \text{(Complementary slackness)}
\end{aligned}$$

The stationarity condition implies that at the optimal solution, the gradient of the Lagrangian function (x,λ,ν) with respect to the decision variable x must be zero. Essentially, the gradient of the objective function is exactly balanced by a linear combination of the gradients of the constraint functions, weighted by their respective Lagrange multipliers.

According to primal feasibility, the solution x must satisfy the original constraints of the primal problem, namely $x \geq 0$ and $e^\top x = 1$. For the dual feasibility, the Lagrange multipliers associated with the inequality constraints must be non-negative $\lambda \geq 0$.

Complementary Slackness is a critical component of the KKT conditions linking primal problem constraints with dual problem multipliers. Specifically, for each constraint i,

$$\lambda_i x_i = 0. \tag{4.15}$$

It indicates that if $x_i > 0$, λ_i must be zero. Conversely, if $\lambda_i > 0$, $x_i = 0$ must hold exactly.

To address the nonlinearity introduced by the complementary slackness condition $\lambda_i x_i = 0$, we can transform it into a set of linear constraints using a large constant $M > 0$ and binary variables $z_i \in \{0, 1\}$ (Yu et al., 2023). That is:

$$\begin{aligned} &\lambda_i \leq M z_i, \\ &x_i \leq M(1 - z_i). \end{aligned}$$

These constraints enable the effectiveness of complementary slackness. If $z_i = 0$, $\lambda_i \leq 0$ forces $\lambda_i = 0$ due to dual feasibility $\lambda_i \geq 0$, and $x_i \leq M$ allows x_i to be positive. If $z_i = 1$, $\lambda_i \leq M$ allows λ_i to be positive, and $x_i \leq 0$ forces $x_i = 0$ due to the primal feasibility $x_i \geq 0$.

4.3.2 Inverse optimization

After outlining the convex optimization and its Lagrangian of mean-variance optimization eq. (4.1), we turn to inverse optimization to inversely estimate risk aversion from the mean-variance optimization as the risk aversion ρ is unknown. Inverse optimization refers to the inference of unknown parameters from an optimization problem based on the knowledge of the optimal solutions. If the observed solution belongs to the optimal solution set $\mathcal{X}^{\text{opt}}(u, \rho)$ of the FOP(u, ρ), inverse optimization finds the optimal ρ in the feasible solution set such that the aggregate fit of the FOP(u, ρ) is optimized to the observed solutions. It is inverse feasible since ρ can be optimized directly by reformulating x with observed solutions in the FOP(u,ρ) model.

However, if the inverse feasibility is not available since the observed data is not the optimal solution to the FOP(u,ρ), we can use data-driven inverse optimization to estimate ρ (Chan et al., 2023). It is usually the case with noisy solution data where the observable solution data are noisy, as human investors inevitably incur behavioural biases and may make mistakes when making decisions (Foerster et al., 2017). Some other noises come from measurement errors during the data collection process. Thus, robo-advisors observe noisy solutions within the rational boundary instead of theoretically optimal solutions. Two developed inverse optimization methods with noisy data are batch learning (Aswani et al., 2018) and online learning (Yu et al., 2023). The former research infers unknown parameters based on a batch knowledge of noisy solutions, while the latter updates risk preferences using concurrent observed resolutions step by step rather than only updating after catching all observations.

The methodologies presented in Aswani et al. (2018) and Yu et al. (2023) are well-suited for our framework, as they are capable of inferring risk aversion coefficients through either batch estimation or an online updating rule. Given the compatibility of online learning with our proposed deep RL framework, which also employs step-by-step updates, we opt for the online inverse optimization approach to estimate risk aversion coefficients. Subsequently, we determine the optimal investment strategies using deep RL.

Online learning, designed to estimate time-varying unknown parameters, has evolved from batch learning techniques (Dong et al., 2018). Online inverse optimization offers several advantages over its batch counterpart, such as significantly accelerating the learning process while maintaining performance guarantees. The existing literature about online inverse optimization demonstrates that these methods converge at a polynomial time rate and achieve statistical consistency. Consequently, it can asymptotically attain the best possible prediction errors while learning parameters with high accuracy and robustness. By incorporating the online learning approach within the context of estimating risk aversion coefficients, we can effectively and efficiently adapt to changing market conditions. Furthermore, the seamless integration with our deep RL framework allows for the simultaneous identification of optimal investment strategies, thus providing a comprehensive and robust solution to the portfolio optimization problem.

Inverse optimization finds the risk aversion parameter ρ if we know all the other variables, including investors' optimal investment solutions and financial market signals. Since we are unable to directly observe the optimal weight vector $x \in \mathbb{R}^n$ because of the noises, such as the behavioural biases and investment mistakes, we employ a variable $y \in \mathbb{R}^n$ to represent the observable portfolio weight vector in reality. In the inverse problem, minimising the distance between observable noisy solutions y and the optimal solutions x can control noises. We use $||\cdot||_p$ to denote the l_p norm. Thus, we define the loss function under each optimal solution set as the minimum predicted distance between observed solutions y_t and the optimal solutions x.

$$l(y_t, u_t; \rho_t) = \min_{x \in \mathcal{X}^{\text{opt}}(u,\rho)} ||y_t - x||_2^2, \tag{4.16}$$

where x is subject to a specific risk aversion parameter ρ according to eq. (4.9). ρ are unknown, but they can be estimated in a pre-defined finite risk aversion set Ψ.

There are several state spaces depending on market situations. For example, we define $S = \{S_1, S_2, \ldots, S_J\}$, where each S_j represents a different kind of state space used to classify a set of similar economic scenarios. This paper sets $J = 2$, with S_1 representing the space of normal states and S_2 representing the space of disaster states where disaster events happen. For the risk aversion ρ_{S_j}, we assume that the true risk aversion is invariant within the state space S_j.

The proposed framework is different from the context of online IOP, in which risk aversion is time-varying at all times. The proposed framework adjusts online inverse optimization to make it suitable for updating risk aversion under different economic state spaces S_j. The optimization updates ρ_{S_j} once receiving the signal that corresponds to S_j and noise solution (u_t, y_t) at time t. For $s_t \in S_j$, the risk aversion under S_j is updated as follows:

$$\hat{\rho}_{S_j,t} = \underset{\rho \in \Psi}{\operatorname{argmin}} \frac{1}{2} ||\rho - \hat{\rho}_{S_j,t-1}||_2^2 + \frac{\eta}{\sqrt{t}} l(y_t, \mu_t; \rho), \tag{4.17}$$

where $\frac{\eta}{\sqrt{t}}$ is a learning rate. $\hat{\rho}_{S_j,t-1}$ is the risk aversion value updated at time $t-1$. $\hat{\rho}_{S_j,t}$ is updated based on the distance from the last updated value $\hat{\rho}_{S_j,t-1}$ and the predicted loss between estimated optimal actions, subject to the estimated risk aversion and the observed actions. The estimated risk aversion $\hat{\rho}_{S_j,t}$ at time t is only formed when investors notice market signals μ_t and also make decisions y_t, as the risk aversion is inherent and dependent under the different state space S_j. After updating the risk aversion under S_j, the estimate is used to update the next estimated risk aversion under the same S_j.

Estimating risk aversion in eq. (4.17) presents challenges due to the high dimensionality of the solutions, which leads to computational complexity, classified as NP-hard (Aswani et al., 2018). Also, even though converting to the dual problem, the unknown parameter ρ prevents it from estimating the solution and dual variables. Instead, we can find the optimal solution by utilizing the KKT conditions of the Markowitz mean-variance optimization eq. (4.9), as detailed in section 4.3.1. The feasibility of eq. (4.9) can be obtained easily, thus there exits solutions to eq. (4.9). Such a solution that satisfies the KKT conditions is the optimal solution. Thus, by incorporating the KKT conditions into the constraint set of the inverse optimization problem eq. (4.17), the method finds the optimal x. Then, we estimate risk aversion $\hat{\rho}$ bounded in Ψ that minimizes the objective function eq. (4.17). Thus, the online updating function and its constraints are:

$$\begin{aligned}
\min_{\rho,x} \quad & \frac{1}{2}||\rho - \rho_{S_j,t-1}||_2^2 + \frac{\eta}{\sqrt{t}}||y_t - x||_2^2, \\
\text{s.t.} \quad & \rho\Sigma x - \mu - \lambda - \nu e = 0, \\
& 0 \leq x \leq M(1-z), \\
& e^\top x = 1, \\
& 0 \leq \lambda \leq Mz, \\
& z \in \{0,1\}^n.
\end{aligned} \tag{4.18}$$

4.4 Reinforcement learning

4.4.1 Preliminary settings

RL is formulated within the framework of a Markov decision process (MDP) environment. An MDP is a mathematical model that formalizes the problem of decision-making in an environment where the outcomes of actions are probabilistic and dependent only on the current state of the environment. An MDP consists of a set of states, actions, transition probabilities, and reward functions. By formulating a problem as an MDP in RL, robo-advisors (also called agents) interact with the environment at a sequence of discrete time, denoted by $t \in T$, and with states over time in a state space that contains different economic scenarios in financial markets, denoted by $s_t \in \mathcal{S}_j$ for $j = \{1,2\}$ with $j = 1$ for the normal state space and $j = 2$ for the disaster state space. At time t, robo-advisors observe some features representing either the normal state or disaster state and take investment actions from an action space $a_t \in \mathcal{A}$ according to their investment policies $\pi(a|s)$. In the next time step $t+1$, the agent transits to the next states according to the transition probability $P(s_{t+1}|s_t)$. It receives the resulting reward r_{t+1} and moves itself into the next state s_{t+1}. We consider rewards r_t defined in the mean-variance analysis as defined in section 4.2.

Our framework employs model-free RL, which operates without the need to know transition probabilities. This approach directly learns the policy $\pi(a|s)$ and optimizes the value function $V(s)$ through interactions with the real-time financial market environment. Model-free methods are beneficial in environments that involve dramatic changes, such as under disaster states, where predicting future states can be unreliable due to the rare occurrence of these transitions. In contrast, model-based approaches rely on knowing the model of transition probabilities and rewards. They calculate expected rewards for each possible next state using transition probabilities and reward functions before selecting actions at the current state. Overall, model-free RL, by focusing on decision-making solely through direct interactions with the financial market, offers a more straightforward approach than model-based approaches.

There are two key assumptions underlying our framework. First, the actions of robo-advisors are assumed to be independent of each other and are based solely on the current state and their individual risk aversion. Second, it is assumed that the impact of any single individual's decision on the overall market is minimal, thus suggesting that the actions taken by robo-advisors do not significantly influence the transition to subsequent economic states. In our simulations, even though agents are actively making investment decisions, they still interact with a realistic economic environment, unchanged with the simulated decisions.

Each state can be featured by a range of indicators. In economics and finance, these features can be macroeconomic indicators and technical indicators. The inflation rates, as the monthly percentage change of the consumer price index (CPI) are used as the macroeconomic indicators. The equation for calculating inflation rates is $\mathrm{INF} = \frac{\mathrm{CPI}_t - \mathrm{CPI}_{t-1}}{\mathrm{CPI}_{t-1}} \times 100$. Moreover, the unemployment rate (UR) plays an important role in representing states, since a change of states often parallels a significant change in the unemployment rate. In terms of technical analysis, various indicators are employed to represent states and make informed investment decisions through deep RL. These indicators include simple moving average (SMA) and relative strength index (RSI). A simple moving average (SMA) is an arithmetic moving average calculated by adding the asset prices from a number of periods and then dividing the total amount by the number of periods. That is, $\mathrm{SMA}_t = \frac{1}{n} \sum_{i=0}^{n-1} p_{t-i}$, where p_{t-i} is the asset price at time $t-i$. SMA smoothes out random fluctuations in past data, thereby providing a view of trends. Relative strength index (RSI) is a momentum oscillator that measures the speed and change of price movements on a scale of 0 to 100. The equation for the m-month RSI is $\mathrm{RSI}_m = 100 - \frac{100}{1+\mathrm{RS}_m}$ where RS_m is the average of m-month gains divided by the average of m-month losses. Deep RL allows these indicators as input to feature the specific states over time, and updates the parameters from the input.

One unknown parameter value ρ is updated from online inverse optimization discussed in section 4.3 and used as a parameter in the reward function in the process of learning investment decisions. The algorithm to simultaneously estimate risk aversion values and make the investment is shown in the algorithm 4.

4.4.2 Policy gradient

Policy gradient is a class of RL algorithms that directly optimize policy to maximize the expected long-term reward instead of ing the value functions of rewards. Unlike value-based methods that approximate and use a value function to compute a procedure, policy gradient methods explicitly outline a policy function with its own parameters to find an optimal policy. The basic idea behind policy-gradient methods is to update the parameters of the policy function in the direction of the gradient of its objective function that measures the long-term expected rewards following a parameterized policy. The parameters are typically updated by stochastic gradient ascent.

Policy-gradient methods benefit in converging to a locally optimal policy. Value-based methods prescribe a near-optimal policy since they rely on a function approximation to estimate the value function, which can introduce errors and approximation bias into the estimates. They may succeed in a good approximation of value function rather than guaranteeing convergence in a near-optimal of the policy (Sutton et al., 1999). This is because any minor change in the estimated value of an action can change the estimated policy in value-based methods. In addition, the optimal policy from the policy gradient is stochastic, selecting a set of actions with specific probabilities. Value-based methods find the deterministic policy of a specific action, which is not realistic in practice (Konda and Tsitsiklis, 1999).

The setting for policy-based RL is described within the standard RL framework, as detailed in section 4.4.1. Policy gradient methods suppose that the policy $\pi_\theta(a|s)$ is differentiable with respect to its parameters θ, such that the partial derivative $\frac{\partial \pi_\theta(a|s)}{\partial \theta}$ exists.

The policy value function is formulated as:

$$J(\theta) = \sum_{s \in \mathcal{S}} d^{\pi_\theta}(s) \sum_{a \in \mathcal{A}} \pi_\theta(a|s) Q^{\pi_\theta}(s,a), \tag{4.19}$$

where θ is a policy network parameter, and $d^{\pi_\theta}(s)$ represents the stationary distribution of states under policy π within the Markov chain framework, and $Q^{\pi_\theta}(s,a)$ is the state-action value function. Although the Q-value function calculates the expected cumulative future rewards, in model-free RL, these values are estimated directly from real interactions with the environment without relying on explicit models or simulations of future states.

A policy gradient theorem articulates the gradient of $J(\theta)$ as follows:

Theorem 4.1. *For any MDP, and for any differentiable policy $\pi_\theta(a|s)$, whether in the average-reward or start-state formulations, the policy gradient is given by:*

$$\nabla_\theta J(\theta) = \mathbb{E}\pi_\theta[\nabla_\theta \log \pi_\theta(a|s) Q^{\pi_\theta}(s,a)]. \tag{4.20}$$

See proof in Appendix 4.A.

The policy parameter vector is updated according to the direction of $\nabla_\theta J(\theta)$. Considering the step size of updating α, the change of policy parameter is $\Delta\theta = \alpha \nabla_\theta J(\theta)$. In practice, calculating the exact gradient involves computing an expectation over all possible states and actions, which can be computationally expensive. A single sample is commonly used to approximate the expectation. In other words, the parameter vector θ is updated using the gradient evaluated at a single sampled state and action, multiplied by the step size α. The update rule is therefore:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(a|s) Q^{\pi_\theta}(s,a). \tag{4.21}$$

4.4.3 Function approximation

Since the $Q^{\pi_\theta}(s,a)$ in policy gradient theorem 4.1 is unknown, it must be estimated. The tabular methods are one way to estimate Q value $Q^{\pi_\theta}(s,a)$. Nevertheless, in many tasks, the state space is relatively large, leading to the difficulty of using tabular RL methods no matter in finding the optimal policy or value function due to the memory limits, time and data needed to fill tabular accurately. On the other hand, function approximation is a generalization used to solve large-scale problems, as it takes samples from a desired function such as a value function or policy function. It represents and approximates value functions or policies in an MDP. Using function approximation, RL algorithms can be applied to a wide range of problems, including high-dimensional control tasks and decision-making problems with large action spaces.

Let $f_w(s,a)$ be an approximator to $Q^{\pi_\theta}(s,a)$ following $\pi_\theta(a|s)$. The update value of parameter vector w is proportional to the differential estimation errors between the true value $Q^{\pi_\theta}(s,a)$ and the approximated $f_w(s,a)$. Therefore, $\Delta w \propto \nabla_w[\frac{1}{2}(f_w(s,a) - Q^{\pi_\theta}(s,a))^2] \propto (f_w(s,a) - Q^{\pi_\theta}(s,a))\nabla_w f_w(s,a)$.

We should choose function approximation carefully to avoid introducing any bias to follow the exact policy gradient. Sutton et al. (1999) illustrate that the policy gradient can be exact to the approximated policy gradient with the two conditions. First, w minimizes the mean square error $\varepsilon = \mathbb{E}_{\pi_\theta}[(f_w(s,a) - Q^{\pi}_{\theta}(s,a))^2]$. Second, w is compatible to the policy. That is:

$$\nabla_w f_w(s,a) = \nabla_\theta \log \pi_\theta(a|s). \tag{4.22}$$

The policy gradient is exact to the approximated policy gradient as shown below:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(a|s) f_w(s,a)]. \tag{4.23}$$

The proof of eq. (4.23) is shown in Appendix 4.B. Moreover, policy iterations with function approximation in eq. (4.23) converge to a locally optimal policy (Sutton et al., 1999).

4.4.4 Actor-Critic

Although we control for unbiased approximate in function approximation, a drawback of the policy-based methods is that they may introduce a large variance due to the use of a stochastic policy gradient. Moreover, they do not involve learning from older information, as a new gradient is estimated using the current trajectory, which means past information is not explicitly stored or used during the learning process. Konda and Tsitsiklis (1999) propose Actor-Critic algorithms to combine the advantages of value-based and policy-based methods. The state-action Q value function is estimated using a critic network, that is, $Q^\pi(s,a) \approx f_w(s,a) = Q_w(s,a)$.

The parameterized Q value function $Q_w(s,a)$ is approximated using a linear function:

$$Q_w(s,a) = \phi(s,a)^\intercal w, \tag{4.24}$$

where $\phi(s,a)$ is a feature vector corresponding to the state-action pair (s,a), and w is the parameter vector of the critic network. According to the updated $Q_w(s,a)$, the policy is thereby parameterized using an actor network. The approximate policy gradient becomes $\nabla_\theta J(\theta) = \mathbb{E}_\pi[\nabla_\theta \log \pi_\theta(a|s) Q_w(s,a)]$. Using the parameter update equation eq. (4.21), the change in the policy parameter θ is:

$$\triangle\theta = \alpha \nabla_\theta \log \pi_\theta(a|s) Q_w(s,a). \tag{4.25}$$

Actor-critic algorithms can effectively reduce variance with the effort of a critic to evaluate how good a policy $\pi_\theta(a|s)$ is for current parameter θ. Then, the actor updates policy parameters θ in the direction suggested by the critic. They propose a projection operator to ensure that the policy parameters remain within the range of admissible values during the learning process. Therefore, the critic-parameterized approximation is directly prescribed by the actor rather than they are chosen independently.

4.4.5 A2C

The Advantage Actor-Critic algorithm (A2C) combines the actor-critic architecture with the advantage function to reduce the variance from plain actor-critic algorithms (Mnih et al., 2016). The advantage function is used to estimate the quality of the policy. That means it estimates the advantage of action a in state s. The advantage function is defined as:

$$A^{\pi_\theta}(s,a) = Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s). \tag{4.26}$$

The policy gradient becomes:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(a|s) A^{\pi_\theta}(s,a)]. \tag{4.27}$$

The advantage function value can be approximated by estimating state value $V_w(s) \approx V^{\pi_\theta}(s)$ and state-action value as $Q_w(s,a) \approx Q^{\pi_\theta}(s,a)$.

In the critic process, the A2C algorithm updates the parameters of the value network w to minimize the temporal-difference (TD) error of the state value function $V_w(s)$. The TD error is defined as:

$$\delta_w = r + \gamma V_w(s') - V_w(s). \tag{4.28}$$

This TD error δ_w serves as an estimate of the advantage function $A_w(s,a)$:

$$\delta_w \approx A_w(s,a). \tag{4.29}$$

The update of w is as follows:

$$w \leftarrow w + \beta \delta w \nabla_w V_w(s), \tag{4.30}$$

where β is the step size in the policy network.

In the actor process, the Q value is replaced by $A_w(s,a)$. The policy parameter θ are developed from eq. (4.21) to the below function:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(a|s) A_w(s,a). \tag{4.31}$$

The proposed algorithm to update risk aversion and learn investment decisions are presented in algo 4.

Algorithm 4 Update risk aversion and optimize policy simultaneously

```
**Input:** State spaces $S_j$ where $j = \{1,2\}$, risk aversion set $\Psi$, portfolio holdings $y$ and asset signals including price of asset class index, expected stock returns.
Initialized values $\hat{\rho}_{S_j,t}$ at $t = 0$.
Start at $s_t \in S_j$ for $t = 0$.
**for** t = 0,1,...,T **do**
    Sample $a_t \sim \pi_\theta(a_t|s_t,\hat{\rho}_{S_j,t})$.
    Observe rewards $r(s_t,a_t,\hat{\rho}_{S_j,t})$.
    Transit to next state $s_{t+1}$ according to the actual market environment.
    Update value network: $w_{t+1} \leftarrow w_t + \beta\, \delta_{w_t} \nabla_{w_t} V_{w_t}(s_t)$.
    Update policy network: $\theta_{t+1} \leftarrow \theta_t + \alpha \nabla_{\theta_t} \log \pi_{\theta_t}(a_t|s_t,\hat{\rho}_{S_j,t}) A_{w_t}(s_t,a_t)$.
    Update $\hat{\rho}_{S_j,t+1} = \operatorname*{argmin}_{\rho \in \Psi} \frac{1}{2}||\rho - \hat{\rho}_{S_j,t}||_2^2 + \frac{\eta}{\sqrt{T}}||y_{t+1} - x||_2^2$.
**end for**
$\rho_{S_j} \leftarrow \hat{\rho}_{S_j,T}$.
**Output:** $\rho_{S_j}$, $\pi^\star(a|s,\rho_{S_j})$.
```

4.5 Data

The paper establishes the criteria for classifying the financial market into normal and disaster states. The classification of states refers to the business cycle data provided by the National Bureau of Economic Research (NBER)[3]. According to the NBER, expansions occur from the trough to the peak of a business cycle, while recessions appear from the peak to the trough. We adopt the NBER's classification, designating disaster states as periods of economic contraction and the remaining time periods as normal states. This classification is justified for the reason that the two disaster states retrieved from the NBER business cycles as we can see in table 4.5.1 coincide with the periods of economic turmoils resulting from the 2008 financial crisis and the COVID-19 pandemic. The algorithm starts in January 2008 as

[3] Business cycles data source: https://www.nber.org/research/data/us-business-cycle-expansions-and-contractions.

the starting point and transitions to the subsequent month, February 2008, as the next state, continuing to progress monthly in this manner.

Table 4.5.1: Summary of state dates

State	Date	Month	Quarter
Normal state space (S_1)	2009.07 (2019Q3) - 2020.02 (2019Q4) 2020.05 (2020Q3) - 2022.12 (2022Q4)	158	50
Disaster state space (S_2)	2008.01 (2008Q1) - 2009.06 (2009Q2) 2020.03 (2020Q1) - 2020.04 (2022Q2)	20	7

The data, spanning from January 2008 to December 2022, is divided into two subsets, including a training set covering January 2008 to December 2019, and a testing set encompassing January 2020 to December 2022, as illustrated in figure 4.5.1. The proposed framework estimates risk aversion and trains the deep RL model using the training set. In contrast, the testing set is designated to evaluate the performance of the trained model. To assess the effectiveness of the proposed algorithm, it is crucial to examine its impact on the testing set.

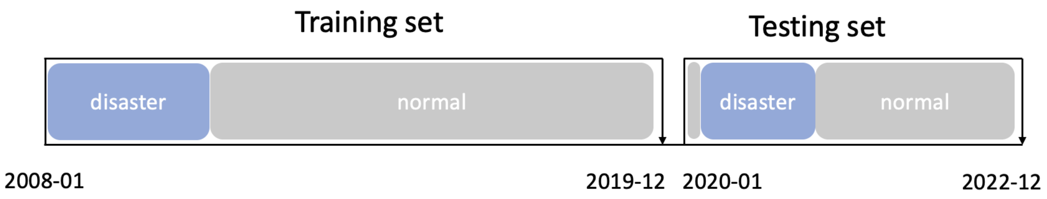


Fig. 4.5.1: Training set and testing set for deep RL. Blue blocks represent disaster states, while grey blocks represent normal states.

In addition, we tune the hyperparameters using three-fold cross-validation on the training set. The hyperparameter set includes the learning rate η and a sufficiently large constant M in the online updating equation (eq. (4.18)), as well as the learning rates α for the policy network and β for the value network. Three-fold cross-validation involves splitting the training set into three equal folds. For each iteration, two folds are used to train the model, and the remaining fold is used for validation. This process is repeated three times, with each fold serving as the validation set once. Consequently, each set of hyperparameters is evaluated through three separate runs. We select the optimal hyperparameters based on the smallest average loss across the three validation sets.

By performing three-fold cross-validation on the training set, we effectively adjust hyperparameters during the model training process, ensuring their robustness and generalization across different data splits. It is crucial that the entire hyperparameter tuning and model training process is conducted solely based on the training set, with the test set completely unused during this process. This ensures the independence and reliability of the test set in the evaluation process.

Regarding the data collection, the required variables and their corresponding data sources are outlined below. For the stock asset class, we use the S&P 500 index (SPX) as a common proxy because of its representativeness for U.S. stock markets. As for the bond asset class, we select the FTSE US Broad Investment-Grade Bond Index (USBIG) as a representative index. The USBIG index encompasses a wide range of bonds including US Treasury, government-sponsored, collateralized, and corporate debt, making it reliable for representing the investment-grade bond market. The prices of SPX and USBIG, as well as holding weights of stocks, bonds, and cash asset classes for mutual funds are collected in Datastream. Also, net asset values (NAV) are gathered to construct mutual funds' returns from Datastream. The expected stock returns are collected from Damodaran (2019)[4]. We choose the one-month treasury bill as a proxy for the risk-free asset (cash) and obtain the treasury yields from the Federal Reserve Economic Data (FRED) database [5]. For deep RL investment, we incorporate technical indicators along with macroeconomic indicators, including monthly inflation and unemployment rates, to present the state features. The latter two variables are collected from the Bureau of Labor Statistics (BLS) [6].

Table 4.5.2 presents the summary statistics of data for the stock index, bond index, risk-free rate, inflation rate and unemployment rate. There is a 180-month data collection from January 2008 to December 2022. The monthly expected stock returns obtained from ERP have the same mean as the historical stock returns of S&P 500, with an average of 0.65%. Bond returns exhibit an average of 0.23% over the dataset. The average risk-free rate is 0.05%, with an average inflation rate of 0.19% and an average unemployment rate of 6.29%.

[4] Data are sourced from his website https://pages.stern.nyu.edu/~adamodar/New_Home_Page/home.htm. From September 2008 onwards, expected stock returns are provided monthly, calculated as the sum of the equity risk premium and Treasury bond yield. For the period from January 2008 to September 2008, we use the expected stock returns in 2007 to avoid forward-looking bias. To align with the frequency of other variables, we convert these annual expected returns into monthly returns by dividing by 12.

[5] FRED data source: https://fred.stlouisfed.org.

[6] BLS data source: https://www.bls.gov/bls/newsrels.htm.

Table 4.5.2: Summary statistics of data. Monthly stock prices (P_s), monthly bond prices (P_b), monthly expected stock returns (μ_s), monthly stock returns (R_s), monthly bond returns (R_b), cash returns as risk-free rate (R_f), monthly inflation rate (INF) and monthly unemployment rate (UR). Returns, inflation rate and unemployment rate are measured in %.

	P_s	P_b	μ_s	R_s	R_b	R_f	INF	UR
count	180	180	180	180	180	180	180	180
mean	2228.74	1573.98	0.65	0.65	0.23	0.05	0.19	6.29
std	1027.84	226.52	0.08	4.71	1.17	0.07	0.33	2.26
min	735.09	1116.76	0.43	-16.94	-4.38	0.00	-1.80	3.50
25%	1341.56	1439.00	0.63	-1.77	-0.31	0.00	0.00	4.30
50%	2061.43	1589.40	0.67	1.24	0.15	0.01	0.20	5.80
75%	2851.18	1685.38	0.70	3.56	0.95	0.08	0.30	8.20
max	4766.18	1975.23	0.89	12.68	4.53	0.34	1.20	14.70

The standard deviation reveals the degree of variability around the mean, with the stock returns showing a relatively high standard deviation of 4.71%, indicating greater volatility compared to bond returns, which have a standard deviation of 1.17%. This suggests that while stock returns offer higher returns, they come with significantly more risk. In contrast, the risk-free rate (R_f) and inflation rate (INF) exhibit lower standard deviations (0.07% and 0.33%, respectively), reflecting their more stable nature over the period.

The minimum and maximum values show the extreme observations in the dataset. For instance, stock returns reached a low of -16.94% and a high of 12.68%, further highlighting the high volatility and risk associated with equity investments during this period. Bond returns, on the other hand, had a smaller range, with a minimum of -4.38% and a maximum of 4.53%.

The lower (25%), median (50%), and upper (75%) quartiles provide further insight into the distribution of the data to know where the majority of observations lie. For stock returns (R_s), the 25th percentile is -1.77%, the median is 1.24%, and the 75th percentile is 3.56%. This indicates that while extreme values exist, the majority of stock returns fall within a moderate range. Similarly, for the unemployment rate (UR), the 25th percentile is 4.30% and the 75th percentile is 8.20%, showing a concentration of observations within this range, with the labour market fluctuating around the 5.80% median. The quantiles provide us with an understanding of both the central tendency and spread of the data.

The work selects six hybrid funds listed in table 4.5.3 according to three types of investment strategies for hybrid funds, including aggressive, moderate and conservative allocations. Investment strategy types determine the allocation of stock and bonds. Aggressive hybrid funds have a higher allocation to stocks, typically $65-80\%$, and a lower allocation to bonds, typically $20-35\%$. The moderate investment strategy allocates $50/50$ to stocks and bonds asset class. Conservative hybrid funds typically have a higher allocation to bonds with $65-80\%$ holding weights and a lower allocation of $20-35\%$ to stocks.

These fund types are designed for investors with different risk tolerance levels and the expectation of returns. Aggressive funds are suitable for investors who expect higher returns and bear higher risk in the long term. In comparison, conservative funds are designed for investors willing to accept lower returns in exchange for a lower level of risk. Moderate funds are a good choice to balance the potential for higher returns with a moderate level of risk. We carefully select two representative mutual funds for each investment type to study their asset allocation and portfolio strategies.

In addition to the performance of mutual funds themselves, we also simulate two other investment strategies for comparison: investment-type benchmark portfolios and the equal-weighted (EW) strategy. On average, aggressive investment portfolios allocate 75% of assets to the stock index, 20% to the bond index, and 5%to cash. Moderate asset allocation portfolios typically hold 50% in the stock index, 45% in the bond index, and 5% in cash. Conservative portfolios consist of 20% in the stock index, 70% in the bond index, and 10% in cash. The EW strategy allocates the same holdings to the stock index, the bond index and cash.

Table 4.5.3: Summary of selected hybrid funds

Investment strategy	**Fund name**	**Ticker**
Aggresive allocation	T Rowe Price Spectrum Moderate Growth Allocation Fund	TRSGX
	American Century One Choice Portfolio	AOVIX
Moderate allocation	Fidelity Balanced Fund	FBALX
	Invesco Equity and Income Fund	ACEIX
Conservative allocation	Vanguard LifeStrategy Conserva	VSCGX
	Chartwell Income Fund	BERIX

Figure 4.5.2 illustrates the NAV of the mutual funds, representing the per-share value of each fund's net assets. The aggressive mutual fund TRSGX exhibits the highest appraisal with the

largest NAV among the funds in our sample. In contrast, the NAV of ACEIX, a fund with moderate allocations, is the smallest. Despite the heterogeneity in asset allocation strategies across the six mutual funds under examination, figure 4.5.2 shows that their NAVs follow business cycle patterns. All funds experienced NAV decreases during the 2008 financial crisis and the COVID-19 pandemic periods.

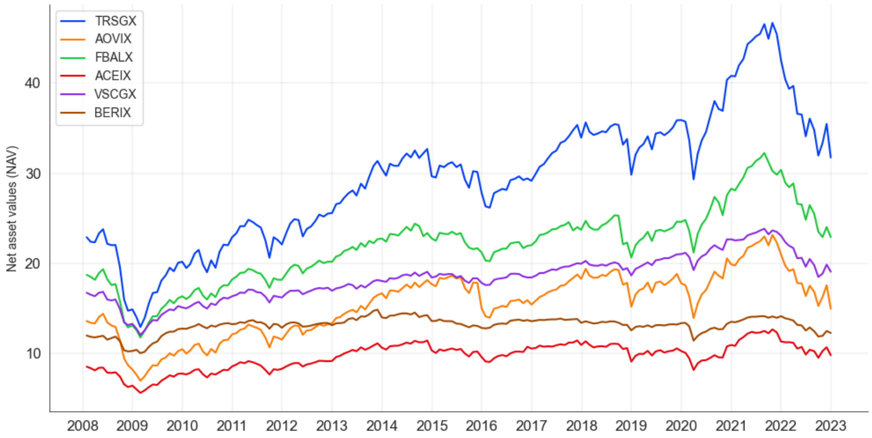


Fig. 4.5.2: NAVs of mutual funds

Except for asset prices and returns, the risk aversion parameter is required to find the optimal weight vector of assets in mean-variance optimization. We set the possible risk aversion in the set $\Psi = \{0.1, 0.2, ..., 9.9, 10.0\}$. The risk aversion set Ψ is defined following the empirical estimation from Bucciol and Miniaci (2011). The asset weight vector will be optimized under each possible risk aversion ρ in the set Ψ. Then, the risk aversion values under Ψ are estimated to be the one that minimizes the predicted loss between observed holdings and the optimal holdings from mean-variance optimization.

The initialization of risk aversion values is based on the assumption that investors are more conservative during disaster states, and investors with more conservative types exhibit greater risk aversion. We set the initialized risk aversion the same for the same strategy type of mutual funds. The risk aversion is then estimated in the inverse optimization given observed portfolio holdings. As shown in table 4.5.4, the initialized risk aversion values for normal states $s_t \in S_1$ are relatively smaller compared to the values in the risk aversion set, while the risk aversion for disaster states $s_t \in S_2$ is 1.5 higher than in normal states. For mutual funds with moderate allocations, the initialized risk aversion lies within the mid-range of the risk aversion set, with 4.0 in normal states and 5.5 in disaster states. Furthermore, risk aversion values are initialized at 7.0 for conservative funds in normal states and 8.5 for disaster states.

Table 4.5.4: Initialize the risk aversion values for mutual funds. $\hat{\rho}_{S_1,0}$ are initialized risk aversion under normal states, while $\hat{\rho}_{S_2,0}$ are initialized values under disaster states. The mutual funds with the same investment strategy are initialized with the same risk aversion value.

Investment strategy	**Ticker**	$\hat{\rho}_{S_1,0}$	$\hat{\rho}_{S_2,0}$
Aggressive	TRSGX	1.0	2.5
	AOVIX	1.0	2.5
Moderate	FBALX	4.0	5.5
	ACEIX	4.0	5.5
Conservative	VSCGX	7.0	8.5
	BERIX	7.0	8.5

4.6 Result

4.6.1 Estimate risk aversion

The proposed framework infers investors' risk aversion by applying online inverse optimization to observed mutual fund holdings, aligning with the revealed preference approach in economics. By studying observable portfolio choices, we infer the implied risk aversion parameters that rationalize investors' decisions in different market states. This methodology extends the work of Chetty (2006), who estimates the coefficient of relative risk aversion from labour supply data, demonstrating how individual behavioural responses reveal underlying preferences. Similarly, we analyze portfolio allocations to infer risk aversion, contributing to the literature by focusing on investment choices rather than consumption or labour decisions.

Figure 4.6.1 presents the estimated risk aversion values for each fund with online inverse optimization. The risk aversion values of all funds are varied at the beginning of the learning process, but they asymptotically converge to accurate values by minimizing the predicted loss in eq. (4.18).

Among aggressive funds, TRSGX and AOVIX update higher values of risk aversion in disaster states than in normal states. For TRSGX, risk aversion converges to 3.3 under disaster states and 0.5 under normal states, while for AOVIX, it converges to 2.3 under disaster states

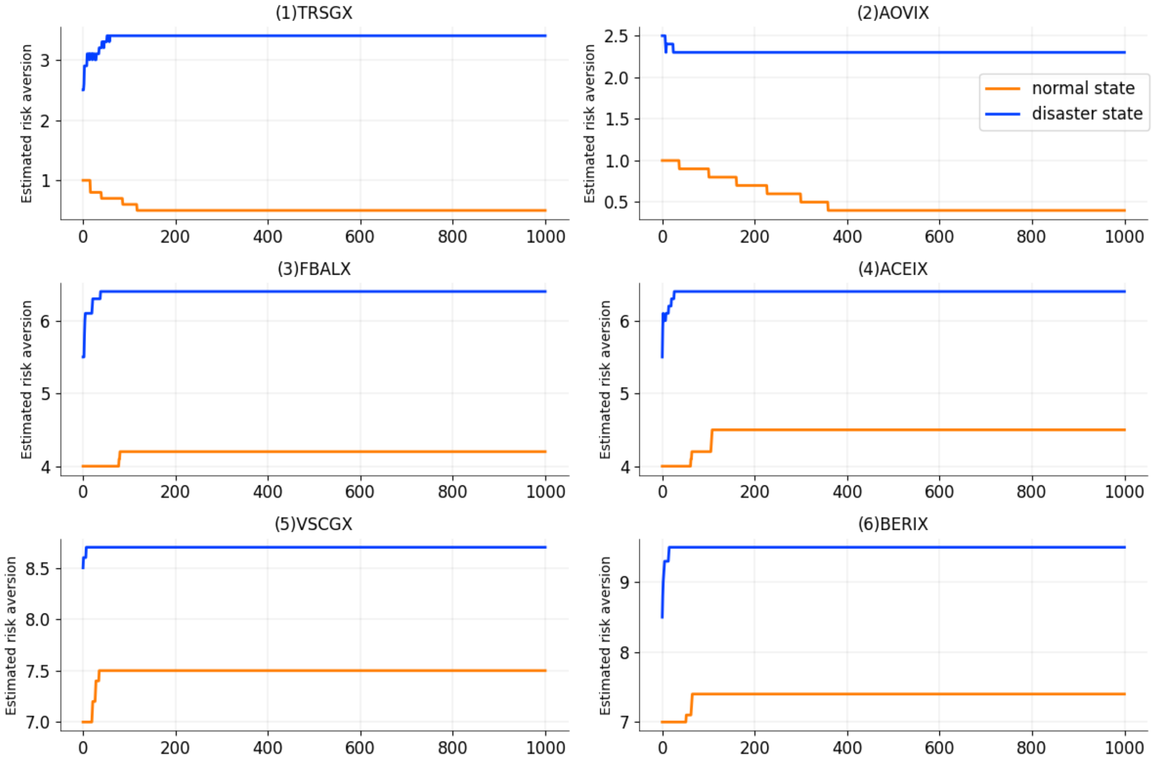


Fig. 4.6.1: Estimated risk aversion values. (1) TRSGX and (2) AOVIX on the first row are hybrid funds with aggressive allocation types, (3) FBALX and (4) ACEIX on the second row are moderate-invested funds, and (5) VSCGX and (6) BERIX on the third row are conservative funds.

and 0.4 under normal states. These funds have relatively low values of risk aversion compared to moderate and conservative funds from (3) to (6) shown in figure 4.6.1. Notably, the risk aversion values in normal states nearly reach the lower bound of 0.1, indicating that aggressive mutual funds can tolerate a relatively high risk of investment under normal states. The convergence value reflects the fact that aggressive funds can hold a larger amount of risky assets in line with their higher tolerance for risk. The results also reveals that aggressive funds respond the most to disasters, as the gap between their risk aversion under disaster states and normal states is the largest among moderate and conservative funds. This implies that investors who are more risk tolerant and willing to accept higher risk should consider investing in aggressive funds, but they should be aware of the increased risk in disaster states.

On the other hand, conservative funds have the largest risk aversion values. VSCGX risk aversion converges to 8.7 under disaster states while 7.5 under normal states. BERIX converges to 9.5 under disaster states while 7.6 under normal states. Conservative hybrid funds nearly touch the upper cap of 10.0, which indicates that they have a large risk aversion when making investments. The conservative risk aversion values indicate that this kind of fund has a substantial aversion to risk and prefers riskless assets when pursuing returns. In line with the large risk aversion, they only allocate a small number of risky assets.

FBALX and ACEIX moderate hybrid funds converge to a range of middle risk aversion values between aggressive and conservative mutual funds in figure 4.6.1. Specifically, for FBALX, the risk aversion values converge to 6.5 under disaster states and 4.2 under normal states, while for ACEIX, they converge to 6.0 under disaster states and 4.6 under normal states. Funds with moderate investment strategies hold a middle range of risky assets and exhibit a moderate level of risk aversion. These findings suggest that moderate funds may be a suitable investment option for investors who seek a balance between risk and returns.

Importantly, our results reveal that investors' risk aversion increases during disaster states across all fund types. This state-dependent risk aversion suggests that investors become more risk-averse in response to adverse market conditions, which is consistent with the Prospect Theory (Kahneman and Tversky, 2013). According to the Prospect Theory, investors react more intensely to potential losses than equivalent gains. During market downturns, increased uncertainty and the fear of losses lead investors to adjust their portfolios towards safer assets. These behaviours reflect the higher risk aversion of investors during disaster states.

The findings contribute to asset pricing literature. We extend the revealed preference approach in asset pricing by empirically demonstrating state-dependent risk aversion inferred from portfolio choices. This aligns with the work of Hansen and Singleton (1982), Chetty (2006) and Koijen and Yogo (2015), who emphasize the importance of preferences in explaining market dynamics. Our results provide micro-level evidence of how individual risk aversion adapts to different economic states.

4.6.2 Deep RL investment

This study evaluates the investment performance of the trained A2C model during the testing period. When assessing the rewards of the deep RL approach, we compare the performance against the respective investment types of the funds and the EW benchmark portfolios.

For the A2C deep RL investment, we set the rewards to be the investors' utilities, as represented in the mean-variance optimization equation eq. (4.9). The deep RL agent selects actions based on the policy $\pi(a|\rho_{S_j}, S_j)$, which is conditional on the well-converged risk aversion and states. Subsequently, the agent receives rewards that correspond to the utilities associated with the learned policy. The analysis highlights the ability of A2C to maximize the rewards in the long run. Figure 4.6.2 highlights the potential benefits of employing the A2C. Despite the initial underperformance for some funds, the A2C algorithm demonstrates superior long-term rewards compared to traditional investment types and the EW benchmark.

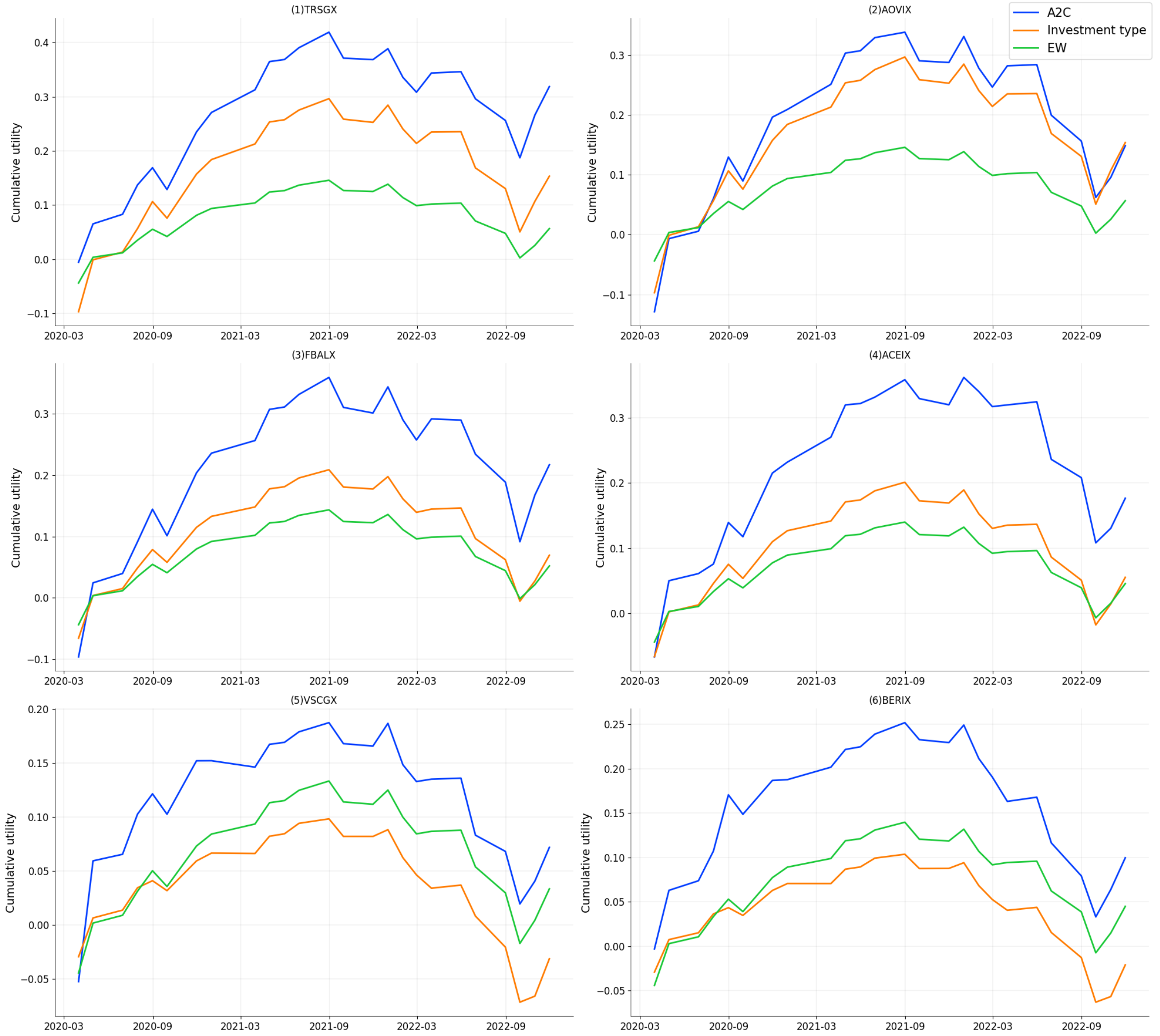


Fig. 4.6.2: Cumulative rewards. The cumulative rewards are the sum of rewards over time. Investment types of (1) TRSGX and (2) AOVIX are aggressive allocation types that allocate 75% of the asset to stocks, 20% to bonds, while 5% to cash. (3) FBALX and (4) ACEIX investment type is moderate allocation with 50% stocks, 45% bonds and 5% cash. Conservative investment types for (5) VSCGX and (6) BERIX include stocks (20%), bonds (70%) and cash (10%) holdings.

The performance of the A2C optimal investment strategy is worth examining, along with the benchmarked investment type allocation strategy and the EW benchmark. The cumulative returns illustrated in figure 4.6.3 represent the returns over time, reflecting the profit potential of various portfolios. For the moderate fund (4) ACEIX, the A2C algorithm demonstrates exceptional performance compared to both the moderate investment and the EW benchmark portfolios throughout the entire testing period. Conversely, while the A2C approach generates substantially higher cumulative returns for the conservative fund (6) BERIX before August 2022, its investment strategy appears to align more closely with the EW portfolio afterwards. Given that the EW portfolios allocate one-third of their weight to each of the three asset classes, this implies that the A2C algorithm also invests conservatively, assigning a smaller portion to risky assets to accommodate the risk aversion of conservative investors. For the remaining funds, the A2C algorithm yields higher cumulative returns in the long run despite experiencing some initial challenges similar to the other two benchmarks. This demonstrates the potential benefits of employing the A2C algorithm for investment decision-making across a variety of funds, as it tends to outperform traditional benchmarks out-of-sample.

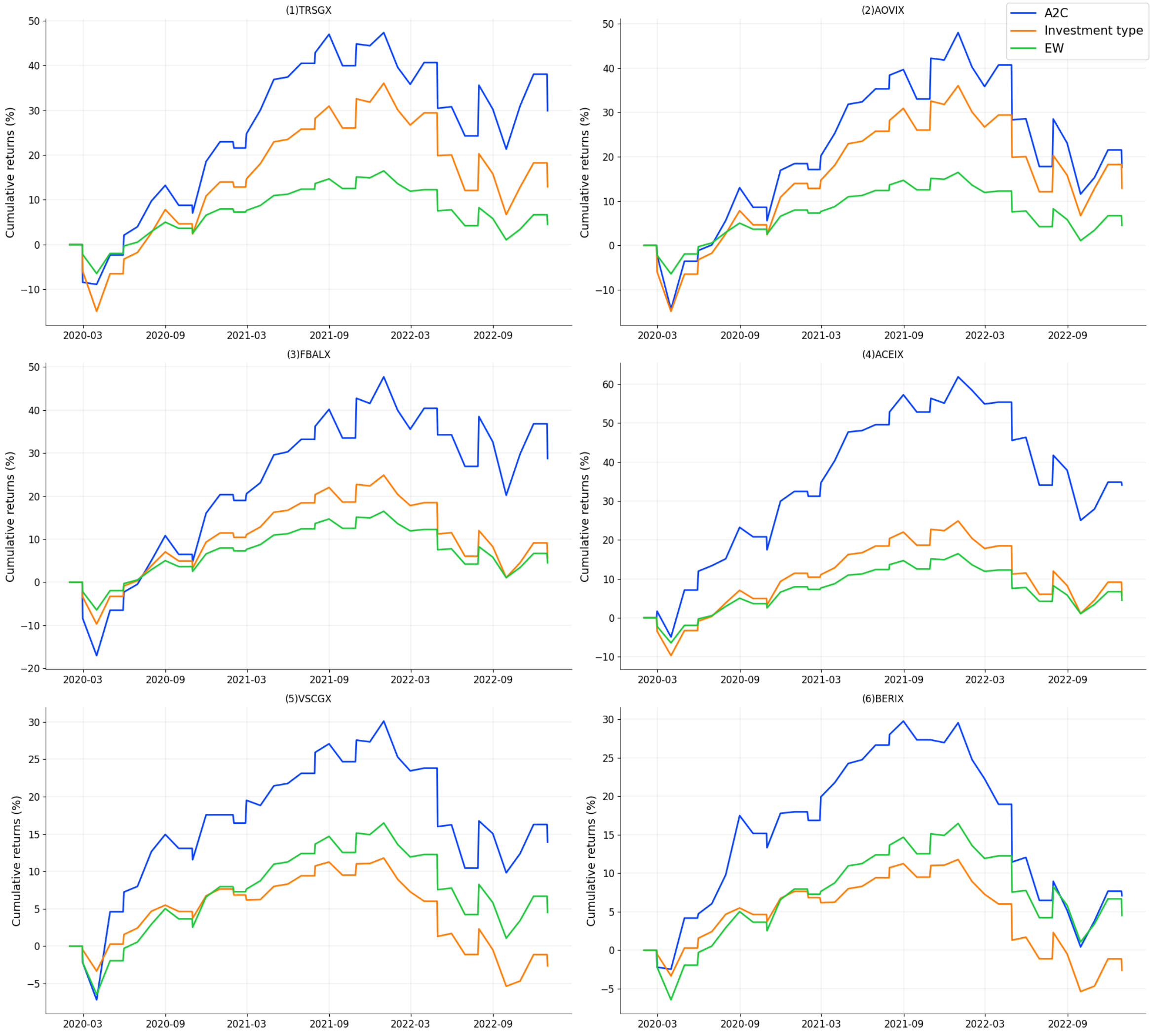


Fig. 4.6.3: Cumulative returns.

In figure 4.6.4, the performance comparison extends beyond the scope of simulated benchmark results by contrasting the proposed A2C optimal strategies with the actual real-world performance of the corresponding mutual funds. The performance of mutual funds is analysed from the NAV of mutual funds. At the onset of the investment period during disaster states, the A2C demonstrates superior performance compared to the actual mutual funds, with the exception of the moderate fund (3) FBALX, which exhibits equivalent performance. These results suggest that A2C effectively mitigates the adverse impact of disaster states on portfolio returns for mutual funds, serving as a buffer during turbulent market conditions. Furthermore, the A2C performance during the normal states surpasses that of the actual mutual funds. This result implies that the A2C algorithm has a superior out-of-sample performance compared to the actual performance of mutual funds across various market states.

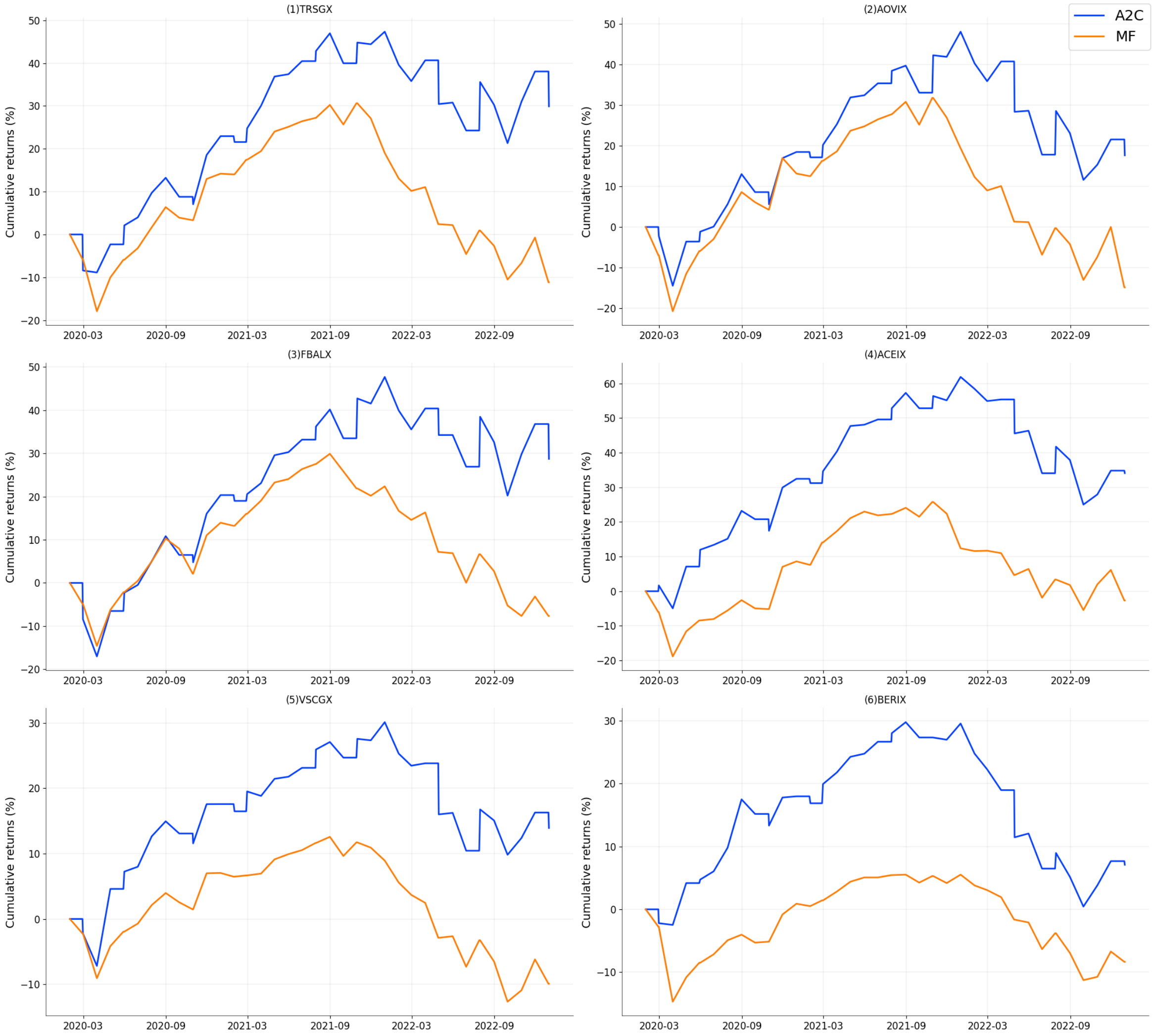


Fig. 4.6.4: Cumulative returns: A2C v.s. mutual funds.

This paper analyzes the out-of-sample results of the A2C optimal strategies for each fund, the benchmark strategies and the EW strategies. As shown in the table 4.6.1 as an example, we present key performance metrics such as average monthly returns (mean), annual returns, standard deviation (std), minimum (min) and maximum returns (max), and quantile returns (25%, 50%, 75%). It also includes risk measures like maximum drawdown, Sharpe ratios and value at risk (VaR). Maximum drawdown represents the largest peak-to-trough decline in the returns during an out-of-sample period, highlighting the magnitude of potential losses. Annualized Sharpe ratios evaluate risk-adjusted returns by comparing expected excess returns to standard deviations. A higher Sharpe ratio indicates better returns per unit of risk. VaR quantifies the maximum expected loss over an out-of-sample time horizon at a given confidence level of 97.5%, providing a measure of downside risk.

In table 4.6.1, the A2C strategies for TRSGX (A2C_TRSGX) and AOVIX (A2C_AOVIX) consistently outperform the mutual funds, aggressive benchmark, and equal-weighted benchmark. The A2C_TRSGX strategy achieves an annual return of 5.72%, surpassing the mutual fund TRSGX's annual return of 5.66%. It also has a higher Sharpe ratio (0.37) compared to TRSGX's Sharpe ratio (0.36). Additionally, the A2C strategies exhibit smaller maximum drawdowns, emphasizing their superior performance and better risk management. The value at risk (VaR) values for A2C strategies also indicate more favourable downside risk characteristics. Specifically, A2C_TRSGX has a VaR of -0.09, the same as TRSGX (-0.09), but with a lower maximum drawdown, indicating a more controlled risk profile. A2C_AOVIX shows a slightly higher VaR of -0.10 compared to AOVIX's -0.12, suggesting that while A2C_AOVIX takes on slightly higher risk than A2C_TRSGX, it still performs better in downside risk control than the traditional AOVIX fund. These results highlight the effectiveness of the A2C approach in enhancing returns and providing greater resilience against market volatility compared to traditional mutual funds and benchmarks during normal states.

Table 4.6.1: Aggressive funds performance under normal states. (1) A2C_TRSGX, (2) A2C_AOVIX, (3) TRSGX, (4) AOVIX, (5) Aggressive, (6) EW.

	(1)	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**
count	33	33	33	33	33	33
mean	0.58%	-0.25%	0.58%	-0.35%	0.37%	0.10%
std	4.87%	4.70%	5.02%	5.79%	4.31%	2.17%
min	-8.41%	-10.52%	-9.34%	-14.85%	-7.87%	-4.50%
25%	-3.92%	-3.50%	-3.14%	-3.94%	-2.95%	-1.48%
50%	1.67%	0.62%	0.91%	0.00%	1.52%	0.30%
75%	3.58%	3.00%	4.24%	3.34%	3.21%	1.36%
max	10.75%	9.32%	10.75%	12.17%	8.28%	3.94%
annual return	5.72%	-4.16%	5.66%	-6.05%	3.43%	0.90%
annual volatility	16.86%	16.29%	17.39%	20.05%	14.94%	7.53%
max drawdown	-0.18	-0.32	-0.25	-0.35	-0.22	-0.13
Sharpe ratio	0.37	-0.23	0.36	-0.25	0.25	0.06
value at risk	-0.09	-0.10	-0.09	-0.12	-0.08	-0.04

In table 4.6.2, which shows the performance of aggressive funds under disaster states, the A2C strategy for TRSGX (A2C_TRSGX) notably outperforms the mutual funds and benchmarks. A2C_TRSGX achieves an impressive average monthly return of 3.74% and an annual return of 54.50%, significantly higher than the other funds. It also boasts a superior Sharpe ratio of 3.26, indicating excellent risk-adjusted performance and resilience during market downturns. In terms of downside risk, the A2C_TRSGX also has a relatively low VaR of -0.04, which is much smaller than TRSGX's VaR of -0.24, reflecting better control of extreme losses at a 95% confidence level. The performance of A2C_AOVIX is also better than its traditional strategy.

The unusually lower absolute maximum drawdown during disaster states than normal states can be attributed to the deep RL model's adaptive risk management. During disaster periods, the model recognizes increased market volatility and uncertainty, making it to adopt a more conservative strategy. Moreover, the performance are improved based on the market situation in the same period. For example, the benchmark portfolios of A2C_TRSGX are TRSGX, the aggressive type, and EW, since these portfolios are evaluated using the same market data.

Table 4.6.2: Aggressive funds performance under disaster states. (1) A2C_TRSGX, (2) A2C_AOVIX, (3) TRSGX, (4) AOVIX, (5) Aggressive, (6) EW.

	(1)	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**
count	3	3	3	3	3	3
mean	3.74%	0.43%	0.90%	1.04%	1.29%	0.72%
std	3.92%	11.81%	12.68%	13.97%	9.86%	4.65%
min	-0.51%	-12.89%	-12.51%	-14.79%	-9.49%	-4.34%
25%	2.01%	-4.16%	-4.99%	-4.25%	-2.99%	-1.32%
50%	4.53%	4.57%	2.53%	6.30%	3.50%	1.69%
75%	5.87%	7.10%	7.61%	8.96%	6.68%	3.25%
max	7.20%	9.62%	12.68%	11.62%	9.86%	4.81%
annual return	54.50%	-0.59%	4.39%	4.47%	12.21%	8.04%
annual volatility	13.57%	40.92%	43.91%	48.39%	34.16%	16.10%
max drawdown	-0.01	-0.13	-0.13	-0.15	-0.09	-0.04
Sharpe ratio	3.26	0.11	0.23	0.24	0.43	0.49
value at risk	-0.04	-0.23	-0.24	-0.27	-0.18	-0.09

The performance statistics in table 4.6.3 highlight the mixed results of the A2C strategies for moderate-risk funds under normal market conditions. A2C_FBALX delivers an average monthly return of 0.69%, slightly higher than the mutual fund FBALX's return of 0.68%. A2C_ACEIX also outperforms with a mean return of 0.10% compared to ACEIX's -0.24%. The A2C_FBALX strategy also outperforms its benchmark in annual return, achieving 7.42%

compared to FBALX's 7.06%, indicating its ability to deliver slightly better returns. In terms of risk, the annual volatility of A2C_FBALX is notably higher at 17.59% compared to FBALX's 14.38%, which reflects greater variability in returns, while A2C_ACEIX exhibits lower volatility at 14.34% compared to ACEIX's 16.26%. The Sharpe ratio of A2C_FBALX is higher (0.52) than FBALX's 0.43, suggesting that its risk-adjusted return is as favourable. Similarly, A2C_ACEIX has a higher Sharpe ratio of 0.03 compared to ACEIX with a negative Sharpe ratio of -0.25.

Table 4.6.3: Moderate funds performance under normal states. (1) A2C_FBALX, (2) A2C_ACEIX, (3) FBALX, (4) ACEIX, (5) Moderate, (6) EW.

	(1)	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**
count	33	33	33	33	33	33
mean	0.69%	0.10%	0.68%	-0.24%	0.14%	0.10%
std	5.08%	4.14%	4.15%	4.69%	3.21%	2.17%
min	-9.34%	-7.86%	-9.34%	-8.31%	-6.63%	-4.50%
25%	-3.92%	-3.14%	-1.97%	-2.08%	-2.15%	-1.48%
50%	1.31%	0.00%	1.02%	0.09%	0.58%	0.30%
75%	4.36%	2.61%	2.61%	2.87%	2.05%	1.36%
max	10.75%	8.72%	10.63%	12.85%	5.85%	3.94%
annual return	7.42%	-0.05%	7.06%	-3.84%	1.05%	0.90%
annual volatility	17.59%	14.34%	14.38%	16.26%	11.11%	7.53%
max drawdown	-0.19	-0.29	-0.23	-0.25	-0.19	-0.13
Sharpe ratio	0.52	0.03	0.43	-0.25	0.08	0.06
value at risk	-0.08	-0.09	-0.09	-0.09	-0.06	-0.04

Table 4.6.4 shows the comparison of the A2C optimal strategies for each moderate fund with the mutual funds, the benchmark strategies, and the EW strategy during disaster states. The A2C strategy for ACEIX (A2C_ACEIX) outperforms the mutual fund ACEIX, achieving an annual return of 11.84% compared to ACEIX's -9.75%. It also has a positive Sharpe ratio of 0.41 versus ACEIX's negative -0.13, indicating better risk-adjusted performance. However, for FBALX, the mutual fund itself performs better than its A2C strategy, with a higher annual return of 47.04% and a Sharpe ratio of 1.26 compared to A2C_FBALX's 29.66% and 0.79.

Table 4.6.5 compares the A2C strategies with their competitors under normal market conditions. For instance, A2C_VSCGX generates an average monthly return of 0.15%, which is higher than VSCGX's 0.04% and significantly better than the conservative benchmark's return of -0.12%. This pattern holds over the annual horizon as well, with A2C_VSCGX achieving an annual return of 1.39%, compared to VSCGX's 0.00% and the conservative benchmark's -1.70%. On the other hand, A2C_BERIX underperforms, posting a negative an-

Table 4.6.4: Moderate funds performance under disaster states. (1) A2C_FBALX, (2) A2C_ACEIX, (3) FBALX, (4) ACEIX, (5) Moderate, (6) EW.

	(1)	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**
count	3	3	3	3	3	3
mean	2.60%	1.30%	3.57%	-0.37%	1.05%	0.72%
std	11.17%	10.35%	9.63%	11.77%	6.92%	4.65%
min	-9.40%	-10.25%	-6.51%	-13.61%	-6.48%	-4.34%
25%	-2.44%	-2.93%	-0.99%	-5.01%	-1.99%	-1.32%
50%	4.53%	4.38%	4.53%	3.59%	2.50%	1.69%
75%	8.61%	7.07%	8.61%	6.25%	4.81%	3.25%
max	12.68%	9.76%	12.68%	8.91%	7.12%	4.81%
annual return	29.66%	11.84%	47.04%	-9.75%	11.16%	8.04%
annual volatility	38.69%	35.87%	33.37%	40.77%	23.96%	16.10%
max drawdown	-0.09	-0.10	-0.07	-0.14	-0.06	-0.04
Sharpe ratio	0.79	0.41	1.26	-0.13	0.49	0.49
value at risk	-0.20	-0.19	-0.16	-0.24	-0.13	-0.09

nual return of -3.83%, which lags behind both its mutual fund counterpart BERIX (-0.95%) and the EW benchmark (0.90%). In terms of risk, A2C_VSCGX demonstrates marginally lower volatility, with an annual volatility of 9.73%, compared to VSCGX's 10.07%. This suggests that A2C_VSCGX delivers returns with slightly less risk exposure. The Sharpe ratio further reinforces the superior risk-adjusted performance of A2C_VSCGX, posting a value of 0.11 compared to VSCGX's -0.02, suggesting a better trade-off between risk and return. A2C_BERI, however, records a much lower Sharpe ratio of -0.43, indicating poor performance relative to the risk taken.

In table 4.6.6, the A2C_VSCGX strategy achieves a higher annual return of 44.59% compared to VSCGX's 31.51%, suggesting that the A2C strategy generated higher returns. However, the mutual fund VSCGX has a higher Sharpe ratio of 2.05 compared to A2C_VSCGX's Sharpe ratio of 1.29, indicating better risk-adjusted returns. For BERIX, the A2C strategy for BERIX (A2C_BERIX) improves upon the mutual fund BERIX, achieving an annual return of 1.16% compared to BERIX's -21.61%. It also has a positive Sharpe ratio of 0.10 versus BERIX's negative -0.67, indicating better risk-adjusted performance. The maximum drawdown for A2C_BERIX is smaller at -0.07 compared to BERIX's -0.12, showing greater resilience during market downturns. These results indicate that the A2C approach can enhance performance and manage risk effectively for conservative funds under disaster states, although its effectiveness may vary depending on the specific fund.

Table 4.6.5: Conservative funds performance under normal states. (1) A2C_VSCGX, (2) A2C_BERIX, (3) VSCGX, (4) BERIX, (5) Conservative, (6) EW.

	(1)	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**
count	33	33	33	33	33	33
mean	0.15%	-0.29%	0.04%	-0.06%	-0.12%	0.10%
std	2.81%	2.82%	2.91%	2.18%	1.98%	2.17%
min	-6.31%	-6.57%	-6.31%	-4.65%	-4.91%	-4.50%
25%	-1.48%	-1.79%	-1.97%	-1.20%	-1.17%	-1.48%
50%	0.20%	0.04%	0.17%	0.07%	0.05%	0.30%
75%	2.20%	1.25%	2.03%	1.29%	0.86%	1.36%
max	5.72%	5.44%	7.01%	4.56%	3.71%	3.94%
annual_return	1.39%	-3.83%	0.00%	-0.95%	-1.70%	0.90%
annual_volatility	9.73%	9.77%	10.07%	7.55%	6.86%	7.53%
max_drawdown	-0.16	-0.22	-0.23	-0.16	-0.15	-0.13
Sharpe_ratio	0.11	-0.43	-0.02	-0.19	-0.32	0.06
value_at_risk	-0.05	-0.06	-0.06	-0.04	-0.04	-0.04

Table 4.6.6: Conservative funds performance under disaster states. (1) A2C_VSCGX, (2) A2C_BERIX, (3) VSCGX, (4) BERIX, (5) Conservative, (6) EW.

	(1)	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**
count	3	3	3	3	3	3
mean	3.38%	0.24%	2.36%	-1.71%	0.72%	0.72%
std	8.92%	6.46%	3.89%	9.20%	3.34%	4.65%
min	-5.09%	-7.00%	-0.28%	-12.27%	-2.86%	-4.34%
25%	-1.28%	-2.34%	0.12%	-4.84%	-0.79%	-1.32%
50%	2.53%	2.32%	0.53%	2.59%	1.28%	1.69%
75%	7.61%	3.86%	3.68%	3.57%	2.51%	3.25%
max	12.68%	5.40%	6.82%	4.55%	3.74%	4.81%
annual return	44.59%	1.16%	31.51%	-21.61%	8.53%	8.04%
annual volatility	30.89%	22.37%	13.47%	31.86%	11.56%	16.10%
max drawdown	-0.05	-0.07	0.00	-0.12	-0.03	-0.04
Sharpe ratio	1.29	0.10	2.05	-0.67	0.69	0.49
value at risk	-0.14	-0.13	-0.05	-0.20	-0.06	-0.09

4.7 Conclusion

This study proposes a novel framework combining online inverse optimization and the A2C deep RL algorithm with learning risk aversion and optimal investment policies tailored to investors' risk profiles under normal and disaster states. Our findings demonstrate that the proposed framework effectively estimates risk aversion for each selected mutual fund. In disaster states, funds exhibit higher risk aversion, which has important implications for robo-advisors and fund managers in ensuring investors' risk tolerance, particularly during periods of market distress.

In the second stage of our study, we test the performance of the trained deep RL algorithm using the converged risk aversion values for each fund. The results illustrate that deep RL can generate exceptional investment strategies tailored to investors, consistently outperforming the EW portfolios, relevant investment type allocations and actual mutual funds. Deep RL demonstrates outstanding performance in normal states and serves as a buffer during disaster states, highlighting its ability to improve investment performance for investors.

While the small sample size limits our study during disaster states, it paves the way for future research in this area. A larger sample size would allow for more accurate estimation to the true risk aversion values. Moreover, future studies can explore solutions for addressing the imbalance between disaster and normal datasets and expand the range of states to estimate risk aversion comprehensively.

Appendices to chapter 4

4.A Proof of policy gradient theorem

The gradient $\nabla_\theta \pi(s,a)$ can be expressed as likelihood ratios. That is:

$$\begin{aligned}\nabla_\theta \pi(s,a) &= \pi(s,a)\frac{\nabla_\theta \pi(s,a)}{\pi(s,a)} \\ &= \pi(s,a)\nabla_\theta \log \pi(s,a).\end{aligned}$$

Thus, the gradient of policy values is:

$$\begin{aligned}\nabla_\theta J(\theta) &= \nabla_\theta[\sum_{s\in\mathcal{S}} d^\pi(s) \sum_{a\in\mathcal{A}} \pi(s,a)Q^\pi(s,a)] \\ &= \sum_{s\in\mathcal{S}} d^\pi(s) \sum_{a\in\mathcal{A}} \nabla_\theta \pi(s,a)Q^\pi(s,a) \\ &= \sum_{s\in\mathcal{S}} d^\pi(s) \sum_{a\in\mathcal{A}} \pi(s,a)\nabla_\theta \log \pi(s,a)Q^\pi(s,a) \\ &= \mathbb{E}_\pi[\nabla_\theta \log \pi(s,a)Q^\pi(s,a)].\end{aligned}$$

4.B Proof of function approximation

This section provides a proof of the function approximation in eq. (4.23).

If the parameter vector w satisfies to minimize MSE, the gradient of MSE must be zero.

$$\begin{aligned}\nabla_w \varepsilon &= 0 \\ \mathbb{E}_{\pi_\theta}[(f_w(s,a) - Q^\pi(s,a))\nabla_w f_w(s,a)] &= 0\end{aligned}$$

If w is compatible to the policy. That means $\nabla_w f_w(s,a) = \nabla_\theta \log \pi(s,a)$.

$$\begin{aligned}\mathbb{E}_{\pi_\theta}[(f_w(s,a) - Q^\pi(s,a))\nabla_\theta \log \pi(s,a)] = 0 \\ \mathbb{E}_{\pi_\theta}[Q^\pi(s,a)\nabla_\theta \log \pi(s,a)] = \mathbb{E}_{\pi_\theta}[f_w(s,a)\nabla_\theta \log \pi(s,a)]\end{aligned}$$

According to eq. (4.20), $\nabla_\theta J(\theta) = \mathbb{E}_\pi[\nabla_\theta \log \pi(s,a) Q^\pi(s,a)]$. Thus,

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[f_w(s,a)\nabla_\theta \log \pi(s,a)].$$

Chapter 5

Nonlinear pricing kernels via neural networks

Abstract

This study proposes a nonlinear pricing kernel approximated through neural networks, addressing limitations of traditional linear models, which capture linear relationships and are prone to overfitting when applied to the factor zoo. The proposed model specification test examines the validity of the nonlinearity assumption of the pricing kernel. Through optimal neural network selection, our findings reveal that a one-layer neural network significantly reduces quadratic pricing errors, indicating its superior pricing performance compared to deep neural networks. Moreover, the role of ESG variables in asset pricing, particularly within the extensive range of factors, remains underexplored. The significance test designed for neural networks shows that ESG variables are significant in asset pricing.

Keywords: asset pricing; pricing kernels; neural networks; model specifications; significance tests

5.1 Introduction

In asset pricing, the appropriate specification of a pricing kernel, or so-called stochastic discount factor (SDF), is a crucial question that drives much research interest. The simplicity and intuitive implications of linear pricing kernels have made them a common choice in asset pricing literature. Hansen and Jagannathan (1991) introduce a minimum-variance pricing kernel that simplifies to a linear projection of asset payoffs. However, Bansal and Viswanathan (1993) suggest the linear assumption might not hold, especially when the payoffs are nonlinear functions of risk factors. The nonlinear payoffs can arise from primitive payoffs or from derivative securities being priced using nonlinear factor pricing models. Moreover, the linear SDF within a *factor zoo* that is traditionally estimated by a cross-sectional regression is subject to the curse of dimensionality, resulting in large out-of-sample pricing errors and severe overfitting (Kozak et al., 2020). Though less explored, the nonlinear pricing kernel approximated by *neural networks* leads to a general representation of the pricing kernel and reduces the out-of-sample quadratic pricing errors.

In this study, we develop a nonlinear asset pricing model that employs neural networks to approximate the pricing kernel for characteristics-managed portfolios (factors) based on U.S. equities. The pricing kernel via neural networks is defined as a nonlinear span of risk factors, meaning it is a linear combination of their nonlinear transformations of factors. The nonlinear functions of these factors serve as basic functions spanning the pricing kernel. This structure allows the nonlinear pricing kernel to capture the nonlinearity in risk factors effectively.

Neural networks offer distinct advantages over conventional asset pricing models due to the approximation power, its flexibility functional form, and ability to handle high-dimensionality. The universal approximation theorem (Hornik et al., 1989; Cybenko, 1989) suggests that a sufficiently large neural network can approximate any function, regardless of what kinds of unknown function that we want to learn. They are considered a class of universal approximators when they have at least one hidden layer, enough hidden units and non-polynomial activation functions.

While neural networks are inherently parametric models, with parameters learned during training, their flexibility contrasts with traditional parametric models that assume fixed functional forms. The nonlinear pricing kernels discussed by Chapman (1997) and Almeida and Freire (2023) impose specific parametric structures, such as polynomial functions or predefined parameter estimations, to manage non-linearity and minimize variance among candidate pricing kernels. In contrast, the adaptability of neural networks allows them to capture more complex relationships and interactions between input features and the resulting pricing kernels (Gu et al., 2020; Chen et al., 2023). This enhanced flexibility enables neural networks to model intricate patterns in financial data without being constrained by a fixed functional form, thereby improving the approximation power of asset pricing models.

Furthermore, neural networks are more suitable for handling the high dimensionality in pricing kernels than conventional linear models. Extant research has identified hundreds of pricing factors which are classed in a factor zoo that effectively price cross-sectional assets (Cochrane, 2009; Feng et al., 2020; Hou et al., 2020). However, traditional methods struggle with the curse of dimensionality, where a small number of data points relative to the number of dimensions reduces the model performance. Gu et al. (2020) and Chen et al. (2023) employ neural networks with regularized techniques such as Ridge regression and dropout to address the curse of high-dimensionality issues.

This paper proposes *specification tests* to evaluate the validity of the linear pricing kernel approximated by the linear models or the nonlinear ones approximated via neural networks. The model specification test sets the null hypothesis as the hypothesized linear model yields the highest pricing errors compared to nonlinear pricing kernels. Then, we compare the out-of-sample test statistic with the quantile of the empirical distribution. As the distribution of test statistics is complicated, we employ Block Bootstrap to estimate the empirical distribution. Eventually, we find that the linear specification is the worst compared to its nonlinear competitors.

Although neural networks are powerful methods to approximate the nonlinear pricing kernel, there are lots of choices for model architectures, such as the number of hidden layers. They prevent us from understanding the optimal approximated pricing kernel via neural networks. Many economic and financial studies, such as those by Gu et al. (2020), Chen et al. (2023) and Fallahgoul et al. (2024), select the number of hidden units and layers arbitrarily or through hyperparameter tuning solely, highlighting the need for a more systematic model selection approach to network architectures.

The paper conducts the optimal neural network selection by employing a hypothesis test. The optional neural network selection sets the null hypothesis, which is the hypothesized optimal neural network that yields the lowest in-sample quadratic pricing errors compared to other neural networks. We compare neural networks with hidden layers from 1 to 5, which are common choices of neural network architectures in asset pricing (Gu et al., 2020; Chen et al., 2023; Fallahgoul et al., 2024). Then, the hypothesized neural network with the smallest out-of-sample pricing errors is selected in the hypothesis test.

Given the optimal selection of the neural network to approximate the pricing kernel, we further utilize *significance tests* to identify useful factors that significantly explain cross-sectional asset prices. Traditionally, neural networks have been viewed as "black boxes" due to their opaque operational nature. Gu et al. (2020), Gu et al. (2021) and Chen et al. (2023) measure the importance of variables and rank them according to the importance values. However, this measure cannot direct us to the absolute importance of factors from the relative rank of variable importance. By employing significance tests, we enhance the transparency of neural networks, addressing concerns about their interpretability in financial applications. This approach is supported by Horel and Giesecke (2020), who demonstrate the viability of significance tests in one-layer neural networks, providing a theoretical foundation for our use in SDF estimation. Different from Fallahgoul et al. (2024) that employ significance tests for various potential neural networks in factor pricing models, our framework focuses on conducting significance tests for the optimal neural network configuration. Our approach not only streamlines the identification of useful pricing factors but also avoids the confusion that can arise from presenting varying significant factors across different models.

Thanks to the feasibility of significance tests for neural networks, we assess the impact of ESG variables in asset pricing considering the factor zoo. Despite some empirical evidence showing that factors such as carbon emissions and climate change influence asset prices (Bolton and Kacperczyk, 2021; Sautner et al., 2023b,a), these ESG variables are underexplored among the factor zoo. A factor that is significant in the low dimensional factor models does not mean that it is significant among the factor zoo. The paper employs significant tests to examine the significance of ESG variables in the factor zoo. Moreover, due to ESG variables' low signal-to-noise ratio, which means the useful information is accompanied by lots of disturbances (Almeyda and Darmansya, 2019), asset pricing models such as CAPM or Fama-French three-factor model with the inclusion of ESG factors can lead to unstable estimates. In addition, ESG variables are often multi-correlated with other pricing factors. Neural networks are capable of managing multicollinearity among input variables, an issue that Ordinary Least Squares (OLS) typically fail due to their assumption of no multicollinearity. This capability is supported by Obite et al. (2020), demonstrating that neural networks provide a better fit and more reliable forecasts than OLS in the presence of multicollinearity.

Our paper addresses three crucial questions regarding pricing kernels in asset pricing. Firstly, we specify nonlinear pricing kernels by employing neural networks with different hidden layers. Secondly, we investigate whether nonlinear pricing kernels perform better than linear pricing kernels in the model specification test. Given the validity of the nonlinear assumption, we conduct model selection strategy to find the optimal neural network configuration. Third, we examine the impact of ESG factors on pricing kernels through the optimal selected neural network.

Our methodological framework introduces two innovations to the pricing kernel. First, we propose a model specification framework tailored for neural networks, distinct from traditional specification tests used in parametric asset pricing models. Also, our approach differs from that of Chen and Ludvigson (2009), who propose a model comparison framework for the intertemporal marginal rate of substitution (IMRS) in assumption based on a habit prior estimated by neural networks. Instead, our model specification test considers the nonlinear pricing kernel approximated directly via neural networks. Second, unlike Fallahgoul et al. (2024), which conduct significance tests across various neural network configurations, this study focuses on identifying the factors that significantly impact the optimal neural network configuration.

Our research about applying machine learning to asset pricing is significant in several aspects. First, unlike most studies that focus on predicting asset returns using machine learning approaches (Gu et al., 2020, 2021), we concentrate on estimating the pricing kernel. This approach not only identifies the fair prices of assets but also uncovers relevant factors for pricing kernel estimation in asset pricing. Conversely, the risk premiums do not necessarily indicate the usefulness of the factors in asset pricing, as these factors may merely correlate with truly influential factors without being directly useful for estimating pricing kernels (Cochrane, 2009; Feng et al., 2020; Kozak et al., 2020). Second, our work pioneers asymptotic tests and examines the null hypothesis of whether the linear pricing kernels have larger pricing errors than the nonlinear specifications via neural networks, which is an area not yet explored in the existing literature. Third, we address a gap in the literature by examining the significance of ESG factors within asset pricing. Given the growing relevance of ESG issues and their complex multi-correlations with traditional pricing factors, our use of neural networks, with their capability for nonlinear transformations, uniquely helps us analyze these relationships and assess the impact of ESG variables on asset pricing. This analysis is crucial as it overcomes the limitations of linear models in dealing with multicollinearity among factors.

This paper studies U.S. equities available in the CRSP database, covering the period from January 2002 to December 2017. This period marks the rising influence of ESG considerations in financial markets. Our dataset comprises 60 pricing factors, including 50 traditional factors from Kozak et al. (2020)'s database and 10 ESG factors. These ESG factors consist of 8 variables related to a range of ESG scores provided by Datastream and 2 variables focusing on climate change exposure and toxic emissions, as detailed by Sautner et al. (2023a) and Hsu et al. (2023) respectively. We integrate these factors with stock returns from CRSP to construct characteristic-based ESG factors, applying the same factor construction of Kozak et al. (2020).

Empirical evidence strongly supports the superiority of nonlinear pricing kernels over the linear candidates. The model specification test results show that the linear pricing kernel is not capable in pricing factors compared to the nonlinear pricing kernel. This finding challenges the traditional use of the linear pricing kernel in asset pricing, suggesting a shift towards nonlinear ones. Additionally, portfolios that mimic nonlinear pricing kernels achieve higher annualized SR than those based on linear pricing kernels, highlighting the outstanding investment performance of the nonlinear pricing kernel. Furthermore, Our analyses reveal that ESG factors are significant in explaining variations in cross-sectional asset prices. Surprisingly, many ESG factors surpass the importance of other commonly used factors in asset pricing. This highlights ESG's growing importance and substantial impact on financial markets.

5.1.1 Related literature

The paper is related to literature about the nonlinear pricing kernel. Bansal and Viswanathan (1993) is the first to implement neural networks in pricing kernel estimation. They use a semi-nonparametric method where the neural net is finite in length. Their study, however, is limited by smaller datasets and the network configuration, which uses only a small number of factors. With only a few features input into the neural networks and a large number of moment conditions, they utilize neural networks to address the over-identification problem. We extend nonlinear pricing kernel models to include a large amount of factors from the factor zoo. Additionally, they employ a single neural network, whereas we discuss the performance and model comparisons of various neural networks with multiple layers and a variety of parameters. Chapman (1997) uses polynomial functions to approximate nonlinear pricing kernels based on a factor called consumption growth. Dittmar (2002) investigates a nonlinear pricing kernel model derived from a Taylor series expansion of the marginal utility function, considering preference theory for covariance, skewness, and kurtosis. Chen and Ludvigson (2009) propose habit-based SDF models where the unknown habit functional form is estimated by the one-layer neural network. Their estimated pricing kernel with the estimated habit as a prior performs better than the Fama-French 3-factor model Fama and French (1993). Almeida and Freire (2023) estimate nonlinear pricing kernels by minimizing variances among candidate pricing kernels, presenting economic implications of nonlinear models. Different from previous literature, our paper extends the extant research of nonlinear pricing kernels to a large dimensional set of factor data via neural networks. This extension is more practical than the nonlinear pricing kernel subject to a smaller dataset (Bansal and Viswanathan, 1993), as there are a lot of pricing factors found in the up-to-date literature. Unlike Chapman (1997), Dittmar (2002) and Almeida and Freire (2023), our unknown pricing kernels are approximated by neural networks, allowing for approximation flexibility.

Our study contributes to machine learning research in asset pricing by tackling issues related to optimal neural network selection and improving transparency. The first major concern is the rigorous justification for selecting a specific neural network method. Asset pricing commonly compares machine learning methods by averaging model performance results from multiple model runs (Gu et al., 2020, 2021; Chen et al., 2023). However, due to the stochastic nature of these models, comparing only a few average scores may not be reliable. The performance of neural networks can vary widely based on chosen hyperparameters and random elements like dropout masks due to their highly non-convex loss functions (Li et al., 2018). This variability makes it challenging to compare different network architectures directly. To address this issue, pioneering work by White (1989) introduces a statistical perspective by deriving a normal distribution for network parameters, and Anders and Korn (1999) apply White (1989)'s methods along with information criteria for model selection. However, the over-parameterization of networks leads to estimation challenges and potential overfitting, which make traditional information criteria less effective for model selection in neural networks. Our approach improves from these methods by running hypothesis testing to compare the out-of-sample performance of competitive models. Our hypothesis tests are developed from Chen and Ludvigson (2009). Different from their method, we compare the neural networks in hypothesis tests rather than the parametric pricing kernel functions. The second major concern of our paper tackles is the statistical significance of factors in explaining cross-sectional asset prices via neural networks. While Fallahgoul et al. (2024) and Horel and Giesecke (2020) provide frameworks for significance testing in neural networks, these primarily focus on return prediction. Our research, however, extends these tests to SDF pricing models, assessing the significance of factors in pricing assets non-arbitrarily.

The paper contributes to the emerging ESG literature by exploring the impact and significance of ESG factors in asset pricing models. On the one hand, lots of existing empirical studies illustrate that ESG variables are pricing factors. For instance, Bennani et al. (2018) treat ESG as a factor influencing abnormal returns and risk exposures in asset pricing. Engle et al. (2020) utilize environmental scores from Asset 4 as green characteristics in constructing factors. Maiti (2021) find ESG factors from Bloomberg statistically significant in an extended Fama-French five-factor model. Bolton and Kacperczyk (2021) identify a carbon risk factor that explains abnormal returns independently. Additionally, textual analyses by Engle et al. (2020), Ardia et al. (2022) and Sautner et al. (2023a,b) construct measures of climate change exposure that price assets effectively. Chen and Liu (2020) use data to forecast returns with deep learning and form profitable ESG trading strategies. Lanza et al. (2023) and D'Amato et al. (2022) use machine learning to link specific ESG indicators like CO2 emissions and waste management to abnormal returns and profitability, respectively. On the other hand, some research suggests ESG variables are not common factors. Halbritter and Dorfleitner (2015) and Naffa and Fain (2022) argue that ESG, despite varying data sources, including ASSET4 and Bloomberg, does not consistently produce abnormal returns within the Fama-French model. The opposite results are due to the different ESG measure matrices from ESG rating companies and the noise of ESG scores. Unlike existing ESG research, our study uses neural networks to extract the useful information from low signal-to-noise ESG data. Also, we implement significance testing to determine the significance of ESG factors among the factor zoo. These ESG variables have not yet been fully explored in the pricing kernel estimation, particularly considering such a comprehensive range of factors.

We organize the rest of the paper as follows. Section 5.2 introduces both linear and nonlinear pricing kernel models. Section 5.3 details the implementation of neural networks. The model specification test and optimal neural network selection are elaborated in section 5.4.1 and section 5.4.2, respectively. The significance test is presented in section 5.4.3. Empirical results are detailed in section 5.5, and the paper concludes with section 5.6.

5.2 Pricing kernel representation

This section explores the representation of the pricing kernel, beginning with the linear pricing kernel and developing to the nonlinear case. Two assumptions are posed to nonlinear pricing kernel models in asset pricing.

A pricing kernel, m_t, ensures that any asset or investment returns, denoted by R_t, satisfy the Euler equation $\mathbb{E}_{t-1}[m_t R_t \mid I_{t-1}] = 1$ (Hansen and Richard, 1987; Hansen and Jagannathan, 1997; Cochrane, 1996), where R_t represents an $N \times 1$ vector of asset returns. This equation illustrates that portfolios yielding the same payoffs must be priced equivalently.

When R_t changes to be the returns R_t^e in excess of the risk-free rates, the pricing kernel model can be expressed as a conditional moment condition.

$$\mathbb{E}_{t-1}[m_t R_t^e | I_{t-1}] = 0, \tag{5.1}$$

where I_{t-1} represents the information set observed by econometrics and investors at time $t-1$.

Hansen and Jagannathan (1991) define the admissible set of pricing kernels $\mathcal{U}$,

$$\mathcal{U} = \{m_t | m_t \geq 0, \mathbb{E}_{t-1}[m_t R^e_t | I_{t-1}] = 0\}. \tag{5.2}$$

Non-negativity of the pricing kernel $m_t \geq 0$ indicates the absence of arbitrage opportunities (Ross, 1976; Harrison and Kreps, 1979; Kreps, 1981). When the market is complete and with common information sets, the admissible pricing kernel is unique across investors; while when the market is incomplete, admissible pricing kernels are not equated. Every pricing kernel in $\mathcal{U}$ can price cross-sectional assets given the moment conditions and non-negativity in eq. (5.2).

The pricing kernel satisfies the unconditional orthogonality conditions as stated below:

$$\mathbb{E}[m_t Z'_{t-1} R^e_t] = 0, \tag{5.3}$$

where $Z_{t-1} \in \mathbb{R}^{N \times q}$ is a vector of instruments comprising firm characteristics included in the information set I_{t-1}. These instruments must be both relevant and exogenous. Relevance ensures that the instruments are related to the excess returns of assets R^e_t. Additionally, the exogeneity implies the instruments are not correlated with the error terms. The characteristics-managed portfolios, also called factors $F_t \in \mathbb{R}^q$, are constructed from the instruments applied to the excess returns (Cochrane, 2009).

$$F_t = Z'_{t-1} R^e_t. \tag{5.4}$$

Assigning eq. (5.4) to conditions eq. (5.3), the unconditional moment conditions becomes:

$$\mathbb{E}[m_t F_t] = 0. \tag{5.5}$$

5.2.1 Linear pricing kernel

The linear arbitrage-pricing theory (APT) implies the existence of a linear pricing kernel. Ross (1976)'s linear APT assumes payoffs are linear in factors and the idiosyncratic noise. The linear payoffs can be priced by the pricing kernel function that is linear and low-dimensional with only a few factors.

The linear pricing kernel is in the linear span of factor returns F_t following Hansen and Jagannathan (1991). F_t should be mean-variance efficient to span the pricing kernel with minimum variance. The linear pricing kernel model $m(F_t)$ is

$$m(F_t) = 1 - b'F_t, \tag{5.6}$$

where $b \in \mathbb{R}^q$ is the pricing kernel loadings that can be estimated by satisfying the pricing eq. (5.5). The functional form of the optimal efficient weights b to construct the mean-variance portfolios on the efficient frontier is

$$b = \Sigma_F^{-1}\mu_F, \tag{5.7}$$

where μ_F is a sample mean of factors $\mu_F = \frac{1}{T}\sum_{t=1}^{T} F_t$, and Σ_F is a second moment of factors $\Sigma_F = \frac{1}{T}\sum_{t=1}^{T} F_t F_t'$.

The invertibility of Σ_F is critical in estimating the SDF coefficients. Challenges arise particularly when the dimension of factors q is high relative to the number of time series observations, or when factors exhibit strong multicollinearity or redundancy. Σ_F tends to be unstable or nearly singular, leading to the difficult inversion of the covariance matrix and enlarging estimation errors. Furthermore, a large q can lead to overfitting of the cross-sectional regression. The overfitting results in a perfect in-sample performance, but adversely affects the model's out-of-sample performance.

To avoid the overfitting of cross-sectional SDF regression, SDF can be regularized using regularization methods such as Lasso, Ridge regression, Elastic Net, and so on. These methods manage the model complexity and reduce overfitting by introducing penalties. When there are redundant factors, research uses Lasso to remove the redundant factors. When features present multicollinearity, Ridge regression is better than Lasso, as Lasso might remove the multi-correlated factors that are relevant to the pricing kernel estimation. Kozak et al. (2020) utilize Elastic Net that combines the penalties of Lasso and Ridge regression to regularize pricing kernel loadings.

$$b = \underset{b}{\operatorname{argmin}}(\mu_F - \Sigma_F b)'\Sigma_F^{-1}(\mu_F - \Sigma_F b) + \eta\lambda||b||_1 + \eta(1-\lambda)||b||_2^2, \tag{5.8}$$

where η is the learning rate of Elestic net, and $\lambda \in (0,1)$ is the parameter of l_1 norm of pricing kernel loadings. The higher the λ, the more parameters b are shrinkaged to exactly zero via the l_1 norm. When $\lambda = 1$, eq. (5.8) takes effect of Lasso. Otherwise, the smaller the λ, the more parameters b shift to near zero by the l_2 norm. When $\lambda = 0$, eq. (5.8) is equivalent to Ridge regression.

5.2.2 Nonlinear pricing kernel

The proposed pricing kernel is a nonlinear span of factors. This means that the factors F_t are processed with nonlinear transformations and are then projected to be the pricing kernel $m(F_t)$. In neural networks, this is achieved by using nonlinear activation functions in the hidden layers. Such transformations enable the neural networks to effectively capture the complex behaviours and dependencies between factors and the pricing kernel.

The nonlinear pricing kernel obeys the essential conditions of admissible pricing kernels. First, the q-dimensional factors F_t exist such that nonlinear SDF functions $m(F_t)$ satisfy the orthogonality conditions. Developed from the linear pricing kernel, the orthogonality unconditional moment conditions are:

$$\mathbb{E}[m(F_t)F_t] = 0. \tag{5.9}$$

Moreover, the nonlinear pricing kernel is non-negative. The non-negativity of pricing kernels is equivalent to no-arbitrage opportunity. These conditions imply that the identified nonlinear pricing kernel exits in the admissible pricing kernel set $\mathcal{U}$, and can be correctly specified.

We do not restrict the pricing kernel to low-dimensional. The APT implies that there are only a few risk factors to price assets (Ross, 1976). Bansal and Viswanathan (1993) follow Ross (1976)'s implication and apply a low-dimensional pricing kernel through neural networks. We argue that their model is limited, as current research has examined hundreds of pricing factors existing in the factor zoo. Ross (1976)'s model specification is subject to only a small number of factors, and we develop it to deal with the high-dimensional factor zoo, discussed in section 5.3.1.

5.3 Approximate the nonlinear pricing kernel

We utilize neural networks to approximate the nonlinear pricing kernel functions. These networks adapt to the amount of data without assuming a predefined functional form, thereby minimizing approximation errors. Neural networks support a broad dimension of parameters, enabling them to handle a large set of input factors effectively in the SDF estimation.

As the input features increases, the parameters of the network expand. Particularly in a single-layer neural network, the network can become very wide to effectively approximate the unknown function, thereby facilitating a flexible pricing kernel function to adapt the data complexity and volume (Goodfellow et al., 2016). However, careful construction of the network is crucial to manage its approximation potential while avoiding overfitting, ensuring that the model represents the underlying data patterns and does not merely adapt to noise.

5.3.1 Construction of neural networks

Our framework employs a *multilayer perceptron* (MLP) which is a specialized type of *feed-forward network* (FFN) renowned for its ability to handle the relationship between input and output variables. In an FFN, information flows strictly forward from input to output layers without any backward connections. The MLP develops this configuration by ensuring that every unit within a layer is fully connected, receiving inputs from all units of the preceding layer. MLPs utilize nonlinear activation functions to capture complex patterns and relationships. The ability of MLPs to approximate the complex relationships between input variables and outputs makes them particularly well-suited for approximating the nonlinear pricing kernel.

Regarding the network structure of the MLP, we use the 2-layer MLP shown in Figure 5.3.1 as an example. The MLP begins with q input units corresponding to their covariates and ends with one output unit for the outcome. The configuration of the MLP features hidden units organized into a sequence of L hidden layers. L also measures the depth of the network. Each layer with $l = 1, ..., L$ is defined by its position in the sequence. The width of the network at each layer is the same, denoted as K, which is the number of hidden units as well. A unit

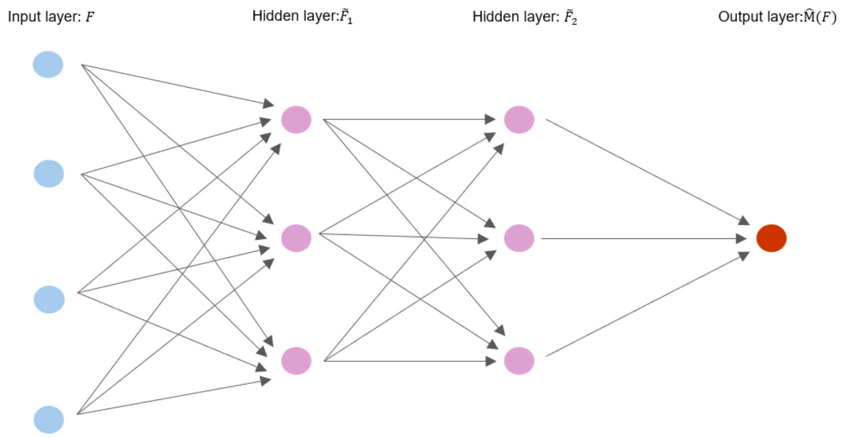

Input layer: F
Hidden layer:$\tilde{F}_1$
Hidden layer: $\tilde{F}_2$
Output layer:$\widehat{\mathrm{M}}(F)$

Fig. 5.3.1: Diagram of F_{MLP}. Input dimension $q = 4$, number of hidden units $K = 3$ and number of hidden layers $L = 2$.

belongs to layer l if it receives input from layer $l-1$ and has no predecessors in subsequent layers.

The nonlinear pricing kernel is estimated via these neural networks, specifically within an MLP class $\hat{m}(F_t) \in \mathcal{C}_{MLP}$. Units on the input layer are factors $F \in \mathbb{R}^q$. Then, we use $\tilde{F}_{k,l} \in \mathbb{R}$ to denote an output of unit k on layer l for $k = 1,...,K$ and $l = 1,...,L$. A set of units for layer $l \leq L$ denotes as $\tilde{F}_l = (\tilde{F}_{1,l},...,\tilde{F}_{K,l})'$. Each unit computes the output with the activation function σ_k as $\tilde{F}_{k,l} = \sigma_k(b_{k,l-1}\tilde{F}_{l-1} + a_{k,l-1})$ where $b_{k,l-1} \in \mathbb{R}^{1\times K}$ is a row vector of weights and $a_{k,l-1} \in \mathbb{R}$ is an intercept for the unit $\tilde{F}_{k,l}$. The pricing kernel function via MLP can be written as an affine function of a sequence of nonlinear functions.

$$\hat{m}(F) = b_L\sigma\Big(\cdots\sigma\big(b_1\sigma(b_0F + a_0) + a_1\big) + \cdots\Big) + a_L, \tag{5.10}$$

where the network weights $b_l \in \mathbb{R}^{K\times K}$ for $l = 1,...,L-1$. In addition, $b_0 \in \mathbb{R}^{K\times q}$ for the input layer and $b_L \in \mathbb{R}^{1\times K}$ for the output layer. The intercepts $a_l \in \mathbb{R}^K$ is a vector where K is the same as the number of the first dimension of b_l for all layers ($l = 1,...,L$). The activation function $\sigma : \mathbb{R}^K \to \mathbb{R}^K$ is componentwise, such that it is applied individually to the combination of outputs from units of layer $l-1$ before being passed to each corresponding unit in layer l.

We use a sigmoid function as the activation function[1] which is smooth. The sigmoid function

[1] There are some other choices for activation functions, including the rectified linear unit (ReLu), tanh, softmax.

is the logistic function, and its functional form is

$$\sigma(b_l\tilde{F}_l + a_l) = \frac{1}{1 + e^{-(b_l\tilde{F}_l + a_l)}}, \tag{5.11}$$

for $l \leq L$. The smoothness of the activation function is essential for the significance tests discussed in section 5.4.3.

To ensure the non-negativity of the pricing kernel, we transform the output with a Softplus function which is a smooth version of ReLu. ReLu obtain zero value for negative input and keeps the positive input unchanged. Nonetheless, the gradient is zero for the negative input when applying ReLu. Thus, we apply Softplus, as a smooth ReLu, to enable all SDF values to be positive and everywhere differentiable in the function region. The function of softplus is $log(1 + e^{\hat{m}(F)})$. Bansal and Viswanathan (1993) utilize ReLU and add a small positive number θ with a specific transformation: $0.5 \times \hat{m}(F) + 0.5 \times \sqrt{\hat{m}(F)^2 + \theta}$. However, their method needs to decide the value of the small θ, while Softplus avoids this necessity.

Our candidate neural network architectures with hidden layers varying from 1 to 5 exhibit varying depths and complexities. The one-layer network (MLP1) is suitable for relatively small and low-dimension datasets because it is simple, fast converged, and has a good approximation with fewer units (Hornik et al., 1989). In contrast, deeper networks with more than one hidden layer are better suited for high-dimensional datasets or to learn complex functions. These multi-layer networks not only require fewer units to represent complex functions but also help minimize generalization errors (Goodfellow et al., 2016). They are suitable for high-dimensional datasets and when the function we want to learn is complex, especially when it is composed of several more straightforward functions.

The total number of parameters for MLPs is calculated as $(q+1)K + (L-1)(K^2+K) + K + 1$ (Farrell et al., 2021). To ensure consistency in neural networks, the number of parameters must increase with the sample size T, and it is naturally limited by T (Chen and Ludvigson, 2009).

Neural network training adjusts weights using forward and back-propagation. Forward propagation processes information from input to output, and back-propagation computes the gradient of the loss function to optimize the network. We use mini-batch gradient descent, a variant of Stochastic Gradient Descent (SGD), to increase computation efficiency and avoid easily getting stuck in local optimization. Moreover, we employ "Adam" a gradient-based optimization algorithm to update parameters. Adam adjusts the adaptive learning rate for each parameter.

5.3.2 Estimation

The criterion function for estimating the pricing kernel via neural networks is:

$$J(b) = \hat{G}_T(b)' W \hat{G}_T(b), \tag{5.12}$$

where $\hat{G}_T(b)$ are $q \times 1$ sample moments. That is:

$$\hat{G}_T(b) = \frac{1}{T} \sum_{t=1}^{T} \hat{m}(F_t) F_t, \tag{5.13}$$

where W is a $q \times q$ weighting matrix. Eq. (5.12) is also the criterion function for the generalized method of moments (GMM).

There are a few choices of the weighting matrix W (Cochrane, 2009). Hansen and Jagannathan (1991) propose to use the inverse of the second moment of $\hat{m}(F_t)F_t$ as a weighting matrix to obtain the efficient GMM estimator. However, this weighting matrix introduces volatile pricing errors. Hansen and Jagannathan (1997) suggest using the inverse of the second moments of returns as the weighting matrix in model comparison. The empirical studies, such as Chapman (1997), show that the criterion function with this weighting matrix attributes to a smaller variance of the estimated pricing kernel than the inverse of the second moments of pricing errors as the weighting matrix. We therefore set the inverse of second moments of factors $W = \mathbb{E}[F_t F_t']^{-1}$ as the weighting matrix.

Eq. (5.12) is equivalent to the first HJ distance (Hansen and Jagannathan, 1997) which shows below:

$$\begin{aligned} \min_{m \in L^2} \quad & ||m - \hat{m}(F)||^2, \\ \text{s.t.} \quad & \mathbb{E}[mF] = 0, \end{aligned} \tag{5.14}$$

where m is the admissible SDF exiting in the admissible set $\mathcal{U}$ and L is the space of potential payoffs.

Neural network literature commonly sets the least square function between the estimator and the observed values as the objective function. Then, the neural networks estimate the network parameters by minimizing the loss, which is the value of the least squares function, to achieve the best fit to the observed values. As the first HJ distance in eq. (5.14) is equivalent to the criterion function in eq. (5.12) with the inverse second moments of factors as the weighting matrix, choosing eq. (5.12) as the objective function in our framework is consistent with the objective function settings used in the neural network literature.

The paper estimates the parameters in neural networks by minimizing loss in eq. (5.12). The nonlinear pricing kernel is therefore estimated.

5.3.3 Cross-validation and sample splitting

The hyperparameters are selected using cross-validation. Cross-validation in neural networks is computationally intensive but essential for ensuring stable and high out-of-sample performance. We adopt three-fold cross-validation, which reduces the computational cost associated with evaluating a vast number of networks compared to higher-fold cross-validation. Hyperparameters are chosen based on minimizing the average loss (quadratic pricing errors) across the three folds.

For the data splitting scheme, we allocate 80% of the data to the training and validation sets, and 20% to the test set. The training set is used to train the neural networks and estimate the weights of neural networks. The validation set is used to tune the hyperparameters. The test set evaluates the trained network. The 80% training and validation data are further split into three-fold subsets for cross-validation. The test set remains entirely separate during the cross-validation process and is solely used for out-of-sample performance evaluation. Alternative sample-splitting methods are discussed in Appendix 5.A.

We utilize “Optuna” to select hyperparameters, which automates the hyperparameter optimization process efficiently. Optuna is an open-source hyperparameter optimization framework designed to automate the search for optimal hyperparameters. It employs state-of-the-art algorithms to efficiently explore the hyperparameter space and identify the best set of parameters (Akiba et al., 2019). We define a search space for hyperparameters, and the "Optuna" iteratively samples different values, evaluating their performance based on the loss using the data set split by three-fold cross-validation. This process significantly reduces the manual effort involved in hyperparameter tuning and ensures a thorough exploration of potential hyperparameter values.

A table of potential hyperparameters and their optimal values is provided in Appendix 5.B.

5.3.4 Overfitting and robustness

The paper employs batch normalization to accelerate training efficiency, while also utilizing dropout and ensemble learning techniques to mitigate the risk of overfitting and increase the overall model robustness.

Batch normalization is a technique designed to improve the training of deep neural networks by standardizing the inputs of each layer. By normalizing the layer inputs, batch normalization allows for a higher learning rate and reduces the dependence on initialization. It stabilizes the learning process and dramatically reduces the number of training epochs required to train deep networks.

Dropout is a regularization method that randomly omits a subset of units during the training phase while retaining all units in the testing. This technique outperforms traditional regularized models such as Lasso and Ridge regression to reduce the overfitting (Chen et al., 2023).

In addition to dropout, we utilize an ensemble learning approach by training 10 independent neural networks with the same model configuration. The approximated pricing kernel of each neural network is obtained by averaging the outputs of these ten ensemble networks. The quadratic pricing error is also the average among ten ensemble networks. The ensemble learning strategy reduces the initialization randomness among the individual networks and increases the robustness of neural network approximation.

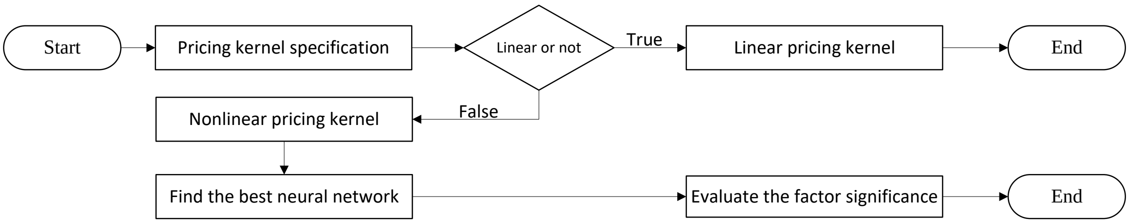

Start
Pricing kernel specification
Linear or not
True
Linear pricing kernel
End
False
Nonlinear pricing kernel
Find the best neural network
Evaluate the factor significance
End

Fig. 5.4.1: Flow chart

5.4 Model selection and evaluation

This paper conducts three kinds of hypothesis tests. The first test is a model specification test, which examines whether the nonlinear pricing kernel through neural networks specifies better than the linear pricing kernel approximated via linear models. The second one involves selecting the optimal neural network architecture among neural networks with hidden layers from 1 to 5. The third one is the significance test of risk factors designed for neural networks.

The flow chart in figure 5.4.1 presents the procedures for the methodology framework. First, the pricing kernel can be assumed to be linear or nonlinear. The linear specification can be approximated through linear models including the linear model and the regularized linear model, while the nonlinear one can be approximated through neural networks. Second, the specification test is conducted to determine whether the pricing kernel is linear or not. If the linear pricing kernel is correctly specified, the linear pricing kernel is estimated, and then the process ends.

Otherwise, the pricing kernel should be correctly specified as nonlinear. Then, we conduct the optimal neural network selection to find the optimal neural network architecture with the smallest out-of-sample quadratic pricing errors compared to other neural networks. Then, we proceed to the significance test for the nonlinear pricing kernel through the optimal neural network. This test is designed to identify the significant factors that explain the cross-sectional variations.

5.4.1 Specification test

According to Hansen and Jagannathan (1991, 1997), the traditional specification tests for parametric SDF models compare the J values or HJ distance to the quantile of χ^2 distribution with a degree of freedom, which is the number of moments minus the number of parameters. However, this approach is not applicable to neural networks because the concept of degrees of freedom does not apply in the same way. Instead, neural networks rely on gradient descent for parameter updates, making the sample size, rather than the number of moments, critical for ensuring consistent estimation. In contrast, GMM representations of SDF models estimate parameters when the model is exactly or over-identified, ensuring that degrees of freedom are positive.

To determine which model specification is better, we conduct a hypothesis test to examine whether all the nonlinear pricing kernels have a smaller pricing error than the linear pricing kernels. If so, the nonlinear pricing kernels approximated by neural networks have better pricing performance than the corresponding linear models. The specification test compares two linear models, including a standard linear model in eq. (5.7) and a regularized linear model in eq. (5.8). We conduct the specification test for each, separately hypothesizing each model as the worst compared to neural networks.

Let $j = 1,...,J$ index the J candidate SDF models. $J = 7$ in total, as we have 7 competing models. $j = 1,2,3,4,5$ represent the neural networks with 1 to 5 hidden layers, respectively. $j = 6$ index the linear pricing kernel model, and $j = 7$ is allocated for the regularized linear pricing kernel. We choose each model's parameters b_j to minimize the quadratic form in eq. (5.12). The HJ distance is the square root of its quadratic value $d_j = \sqrt{J(b_j)}$.

First, we set the hypothesised worst model as the linear model $j = 6$ and the competing models as neural networks $j = 1,2,3,4,5$. As this specification test compares the linear model sorely with neural networks, the regularized linear model is excluded and will be tested separately in the next test. The null hypothesis is:

$$H_0: \quad \max_{j=1,2,3,4,5} \{d_j^2 - d_6^2\} \leq 0. \tag{5.15}$$

The null hypothesis means that even the worst neural network is better than the linear model.

The alternative hypothesis is:

$$H_1 : \max_{j=1,2,3,4,5} \{d_j^2 - d_6^2\} > 0. \tag{5.16}$$

The alternative hypothesis illustrates that the linear model has a smaller pricing error than at least one neural network.

We employ the test statistic formulated in White's reality check test (White, 2000). The test statistic is given by:

$$\phi = \max_{j=1,2,3,4,5} \{d_j^2 - d_6^2\}. \tag{5.17}$$

The test statistic ϕ measures the maximum difference between the square HJ distance for model j and the linear model.

White (2000)'s reality check test addresses the issue of data snooping, which occurs when the dataset is repeatedly used for statistical inference. The problem with data snooping is that it increases the likelihood to find spurious results that appear significant purely by chance. The test statistic is specifically designed to mitigate this issue by testing the null hypothesis that all the neural networks that do not have larger pricing errors over a benchmarked linear model and exploring Bootstrap in the reality check test.

The test statistic has a complicated limiting distribution, making it challenging to directly infer the distribution equation. To address this, we employ the *Block Bootstrap* method to estimate the finite sample distributions of the test statistics following (Chen and Ludvigson, 2009). Block Bootstrap is a nonparametric approach that does not assume the data distribution. It allows for resampling with replacement while preserving the temporal dependence structure within the blocks of data. It is suitable for the time-series data. By dividing the data into blocks, this method maintains the correlation between observations within each block, which is crucial for accurate inference in time-series data. The use of Block Bootstrap helps mitigate the potential bias and variability that could arise from the complex dependence structure in the time-series financial data, especially when a given dataset is used more than once for the purpose of statistical inference. It leads to a robust estimation of the test statistic's distribution.

The Block Bootstrap splits data samples with fixed length, and there are n blocks, which are an integer that derives the number of samples by the length of blocks. We set the length as 21 to cover a one-month length of trading days. If the block length is smaller like a one-weak length than a one-month length, the Block Bootstrap captures the shorter-length time-dependent relationship.

The pseudo-code to conduct the Block Bootstrap for estimation is listed below.

Algorithm 5 Block bootstrap

```
Sample $n$ non-overlapping blocks with replacement.
Split 80% of the data into a training set and 20% into a test set.
Optimize parameters $b_j$ to minimize the squared HJ distance using the training set.
Evaluate the squared HJ distance using the test set.
Calculate the bootstrap test statistic $\phi^b$ with the test set.
```

We repeat the above procedures B times to compute the Bootstrap estimates of p values.

$$\hat{p} = \frac{1}{B} \sum_{b=1}^{b} 1\{\phi^b > \phi\}, \tag{5.18}$$

where ϕ^b is the Block Bootstrap test statistic.

The p values are formulated assuming the null hypothesis H_0 exits. We conduct an upper-tailed test to decide whether to reject or not to reject H_0. At a $\alpha = 5\%$ significance level, if $\hat{p} > \alpha$, the test cannot reject H_0, and concludes that even the worst neural network has a smaller quadratic pricing error than the linear model. Otherwise, we reject H_0 and indicate the linear specification is better than the nonlinear pricing kernels.

Moreover, the paper also conducts another specification test to examine whether all nonlinear pricing kernels perform better than the regularized linear pricing kernel via the Elastic Net. We set the hypothesised worst model $j = 7$ as the Elastic Net, and the competing models as neural networks with $j = 1,2,3,4,5$ for the nonlinear pricing kernels. The following hypothesis settings, test statistics, and p values are the same as those of the above specification test for the linear model.

5.4.2 Neural network selection

We argue that performance comparisons based on the averaged pricing errors over ten ensemble learnings are not enough to decide the optimal neural network configuration. This is because the performance of neural networks depends seriously on the selection of hyperparameters, the initialization of network parameters, or stochastic factors like dropout. Thus, we compare the out-of-sample performance of candidate pricing kernels for neural networks.

According to the results of average quadratic pricing errors in table 5.5.3, we know that MLP1 is the model with the smallest in-sample quadratic pricing errors. Thus, the test evaluates the MLP1 with $j = 1$ as the hypothezed best case among other models $j = 2,3,4,5$. We change to hypothesis to be:

$$H_0: \quad \max_{j=2,3,4,5} \{d_1^2 - d_j^2\} \leq 0. \tag{5.19}$$

The null hypothesis indicates MLP1 has the smallest quadratic pricing error compared to the four other competing neural networks. The alternative hypothesis is:

$$H_1: \quad \max_{j=2,3,4,5} \{d_1^2 - d_j^2\} > 0. \tag{5.20}$$

The alternative hypothesis means that at least one deeper neural network has a smaller quadratic pricing error than MLP1.

The test statistic is shown below:

$$\delta = \max_{j=2,3,4,5} \{d_1^2 - d_j^2\}. \tag{5.21}$$

The neural network selection follows the Block Bootstrap procedures (algorithm 5) to estimate the distribution of the test statistic. Then, we repeat the above procedures B times to compute the Bootstrap estimates of p values. Then:

$$\hat{p} = \frac{1}{B} \sum_{b=1}^{B} 1\{\delta^b > \delta\}. \tag{5.22}$$

If $\hat{p} > \alpha$ at the 5% significance level, the test cannot reject H_0, and concludes that MLP1 is the best candidate pricing kernel with the smallest pricing error. Otherwise, we reject H_0 and conclude that there is at least one candidate pricing kernel model except for MLP1 has a smaller pricing error. In this case, we turn back to the step of selecting another hypothesized model that has a second smaller pricing error than MLP1, and conduct the neural network selection again.

The neural network selection utilizes out-of-sample data to calculate the test statistic to evaluate its generalization ability. By setting the neural network with the smallest in-sample quadratic pricing errors, and then calculating the test statistic and distribution based on out-of-sample data, the framework balances the in-sample approximation ability, as well as the model generalization. This also helps to avoid the overfitting.

5.4.3 Significance test

Unlike traditional regression models where we can employ a t-test to determine the significance of variables, machine learning methods often act as a *black box*, making it challenging to investigate variable significance. Machine learning research in asset pricing derives *variable importance* to measure how importance of pricing factors or characteristics are (Gu et al., 2020, 2021; Chen et al., 2023). The variable importance of a variable of interest is measured by the average of partial derivatives of the neural network estimator with respect to this variable. Although we can see whether a factor is relatively important or not by its ranking of variable importance compared to other factors, the method does not infer whether such a factor is significant to the pricing kernel.

Recent research has made progress in this area by developing statistical methods to test the significance of variables in one-layer neural networks (Horel and Giesecke, 2020). Fallahgoul et al. (2024) apply this method to asset pricing, providing variable significance tests for multi-layer neural networks.

Regarding the significance test for neural networks, the null hypothesis assumes that the test statistics ξ of factor i equals zero. The null hypothesis is:

$$H_0: \quad \xi_i = 0. \tag{5.23}$$

The alternative hypothesis is:

$$H_1: \quad \xi_i \neq 0. \tag{5.24}$$

The test statistic ξ_i is the average of squared partial derivatives, with weights P defined by the distribution of input factors F.

$$\xi_i = \int_{\mathcal{F}} (\frac{\partial m(F)}{\partial F_i})^2 dP(F), \tag{5.25}$$

where $m(F)$ is the pricing kernel in the admissible set $\mathcal{U}$ in eq. (5.2). P describes the distribution of F over the factor space $\mathcal{F} := \mathbb{R}^q$. If the pricing kernel is assumed to be linear, the null hypothesis can take the form of $H_0 : b_i = 0$ where b_i is the coefficient of a factor in linear regression. Thus, the hypothesis can be tested using a standard t-test. However, in the case of a nonlinear pricing kernel function, the derivative of $\frac{\partial \hat{m}(F)}{\partial F_i}$ is not explicit but depends on F.

Since the paper works with data samples, we use sample averages to measure the integral in the empirical function of ξ_i. Thus, $\hat{\xi}_j$ is the average sample squared partial derivative of the neural network estimator $\hat{m}(F_t)$,

$$\hat{\xi}_i = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{\partial \hat{m}(F_t)}{\partial F_i}\right)^2. \tag{5.26}$$

Horel and Giesecke (2020) asymptotically derive a function of test statistic distribution. The scaled test statistic ξ_i[2] asymptotically converges to the distribution of $\tau[h^*]$ where h^* is the argmax of the Gaussian process with the zero mean, and the covariance matrix of the sampled neural networks. The argmax is the index of the unique maximum value of the Gaussian process. The zero mean and the covariance matrix of the sampled neural networks are used to construct the Gaussian process, so as to find the argmax h^*. The empirical function of

[2] The scaler is inverse to the square of the upper bound, which is the difference between the estimator and the true function (Fallahgoul et al., 2024). The smaller the bound, the larger the scaler.

$\tau[h^*]$ is

$$\tau_i[h^*] = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{\partial h^*(F_t)}{\partial F_i}\right)^2. \tag{5.27}$$

The derivation of asymptotic distribution helps avoid the need to employ bootstrap to approximate the distribution of statistics by repeat sampling and reduce the computation cost.

The framework utilizes a discretization approach to estimate the asymptotic distribution of test statistics. The discretization approach can be used to approximate a continuous Gaussian process by a discrete set of sampled values. As the limiting distribution of test statistics is a function of the h^* of a Gaussian process, the distribution is estimated by repetitively sampling h^* many times.

To estimate the Gaussian process indexed by the function space that includes all possible neural networks $\hat{M}(F_t)$, we generate n_m random neural networks in eq. (5.10) by sampling network parameters. The parameters are sampled from a Glorot normal distribution, which is a truncated normal distribution centred at 0 with a standard deviation of $\sqrt{2/(q+1)}$ where q refers to the number of input factors, and 1 refers to the number of output. The sampled n_m neural networks approximate the $\xi-$cover function space.

Then, we obtain h^* in the following ways shown in algorithm 6. By sampling n_m random neural networks, we have the n_m dimensional multivariate normal distribution. Then we obtain a sample from the multivariate normal distribution. The argmax is the index of the maximum value from the sample. By repetitively sampling n_h times of h^* and calculating $\tau[h^*]$, the framework estimates the limiting distribution of the test statistics.

The discretization is challenging due to the difficulty of simulating the Gaussian process and estimating the limiting distribution of test statistics. The correct estimation depends on the number of sampled neural networks, the number of sampled h^*, and the model complexity of neural networks. If the model complexity is complex, we need to have many n_m and n_h. Otherwise, the distribution is likely to increase a few redundant values if the model complexity is simple but with large n_m and n_h (Fallahgoul et al., 2024).

We set $n_m = 500$ neural networks for each iteration and $n_h = 10,000$ iterations to cover enough values to estimate the test statistic distribution[3].

[3] Fallahgoul et al. (2024) generate data using a data-generating process in the Monte Carlo study. They evaluate

Algorithm 6 Compute the asymptotic distribution

```
1: Sample $n_m$ one-layer random neural networks by initializing weights from the Glorot normal distribution.
2: Simulate a sample from the multivariate normal distribution of $n_m$ networks.
3: Find $h^*$ from the sample.
4: Calculate $\tau_i[h^*]$.
5: Repeat the above steps $n_h$ times to generate the empirical distribution of test statistics.
```

The framework calculated the test statistics by averaging $\hat{\xi}_i$ over ten times of ensembling learning for the same network structure. Then, we conduct the significance test using the following rules. The scaled test statistic $\hat{\xi}_i$ of F_i is compared to the $100(1-\alpha)\%$ quantile of the estimated distribution of test statistics. If the test statistic is larger than the quantile, we reject H_0 in favour of H_1 at the $100\alpha\%$ significance level. Otherwise, we do not reject H_0, which means factor i is insignificant in pricing cross-sectional assets among the high-dimensional factor zoo.

5.5 Empirical analysis of U.S. equities

5.5.1 Data and factor construction

There are 60 factors in total, including 50 pricing factors that have been proven to price cross-sectional assets, and 10 ESG factors that we are interested in whether they are significant to asset pricing. The returns of 50 pricing factors are collected from Kozak et al. (2020)[4]. The description of these factors is available in Appendix 5.C. Data to construct ESG factors are collected from Datastream, Sautner et al. (2023a) and Hsu et al. (2023). Also, we collect stock returns from CRSP and risk-free rates from Fama-French website. We deduce risk-free returns from asset returns to obtain excess returns of assets.

the performance of the significance test using lots of distributions simulated in the Monte Carlo. However, it is not necessary to do this in the empirical study, as long as the number n_h of sampled $\hat{h}^*$ is enough to estimate the limiting distribution. Otherwise, the computation cost will be massive.

[4] Their data is available at: https://sites.google.com/site/serhiykozak/data.

Data were collected from January 2002 to December 2017. ESG data started gaining attention in the 21^{st} century, and subsequently, various data vendors began providing ESG data. To ensure that the ESG data of interest corresponds with the pricing factor data, we discard the pricing factor data prior to 2002. Given the numerous pricing factors we introduced, and to avoid overfitting due to insufficient observations, we constructed daily factor returns using the daily asset returns. To avoid forward-looking bias, we applied time lags to the factors to match returns (Kozak et al., 2020; Gu et al., 2020). To match returns at month t, we use the most recent monthly characteristics at the end of month $t-1$ and the most recent quarterly data by the end of $t-4$. Annual characteristics available in year $t-1$ are matched to returns from July at year t to June at year $t+1$.

There is a set of ESG collection from Datastream. They are ESG combined scores (esgscore), environmental scores (envscore), social scores (socialscore), and governance scores (govscore). Also, we collect their sub-categories from environment scores, including emission (emission), innovation scores (envinnova), and resource use scores (resuse). In addition, we construct a greenness to measure the stocks' greenness level following Pástor et al. (2022)'s measure. The unadjusted greenness score of firm s at the beginning of month t is

$$greenness_{s,t-1} = -(100 - envscore_{s,t-1}) \times envweight_{s,t-1}, \tag{5.28}$$

where $envscore_{s,t-1}$ is environmental scores, and $envweight_{s,t-1}$ is the weight of environmental scores across the same industry[5]. The greenness measures how close the company t reach to the perfect environmental score of 100. The closer the greenness to zero, the better the company's greenness level. We consider the environmental weights to make the greenness level comparable across industries. Moreover, we collect the climate change exposure (ccexpo) from Sautner et al. (2023a) and the toxic emission (toxicemission) from Hsu et al. (2023). Both of them are evaluated as effective pricing factors in their papers.

After gathering all relevant characteristics and asset excess returns, we apply specific normalizations to these characteristics to define our characteristics-based factors, as detailed by Kozak et al. (2020). This normalization process aims to concentrate exclusively on the cross-sectional aspect of return predictability, eliminate the impact of outliers, and ensure consistency across all portfolios. The characteristics are normalized as follows.

[5] The sum of environmental scores, social scores and governance scores is 1 across the same industry

First, following Asness et al. (2019) and Freyberger et al. (2020), we implement a rank transformation for each characteristic. For each characteristic i of a stock s at a given time t, denoted as $c_{i,s,t}$, we sort all stocks based on the values of their respective characteristics $c_{i,s,t}$ and rank them cross-sectionally (across all s) from 1 to n_t, where n_t is the number of stocks at time t for which this characteristic is available. For the unavailable characteristics, we replace them with zero, which is the mean value of weights. We then normalize all ranks by dividing by $n_t + 1$ to obtain the value of the rank transform:

$$rc_{i,s,t} = \frac{rank(c_{i,s,t})}{n_t + 1}. \tag{5.29}$$

Next, we normalize each rank-transformed characteristic $rc_{i,s,t}$ by first centering it cross-sectionally and then dividing by the sum of absolute deviations from the mean of all stocks:

$$z_{i,s,t} = \frac{rc_{i,s,t} - \bar{rc}_{i,s,t}}{\sum_{s=1}^{n_t} |rc_{i,s,t} - \bar{rc}_{i,s,t}|}, \tag{5.30}$$

where $\bar{rc}_{i,t} = \frac{1}{n_t}\sum_{s=1}^{n_t} rc_{i,s,t}$. The resulting zero-investment long-short portfolios of transformed characteristics $z_{i,s,t}$ are insensitive to outliers and allow us to maintain a fixed absolute amount of long and short positions invested in the characteristic-based strategy. Finally, we combine all transformed characteristics $z_{i,s,t}$ for all stocks into a matrix of instruments, $Z_t \in \mathbb{R}^{n\times q}$. Interaction with the excess returns of assets, $F_t = Z_{t-1}' R_t^e$, then yields one factor for each characteristic.

In addition, we orthogonalize all characteristic-based factor returns with respect to the CRSP value-weighted index returns. According to the factor pricing models, we know that the market factor is the dependent variable of all factor returns (Fama and French, 1993). Since we focus on the factors that explain the cross-sectional variations, we exclude the market factor from the factor zoo to estimate the pricing kernel.

5.5.2 Descriptive statistics

Table 5.5.1 tabulates statistical measures, including mean, standard deviation, minimum, median, and maximum for the interested variables including traditional asset pricing factors, alongside ten ESG variables. The top 5 factors are studied from the Fama-French 5 factor model (Fama and French, 1996) and are useful to include within asset pricing literature because of their abilities to price cross-sectional assets[6]. The rest of the factors are 10 ESG variables that are potentially useful in asset pricing.

Table 5.5.1: Descriptive statistics for factor returns

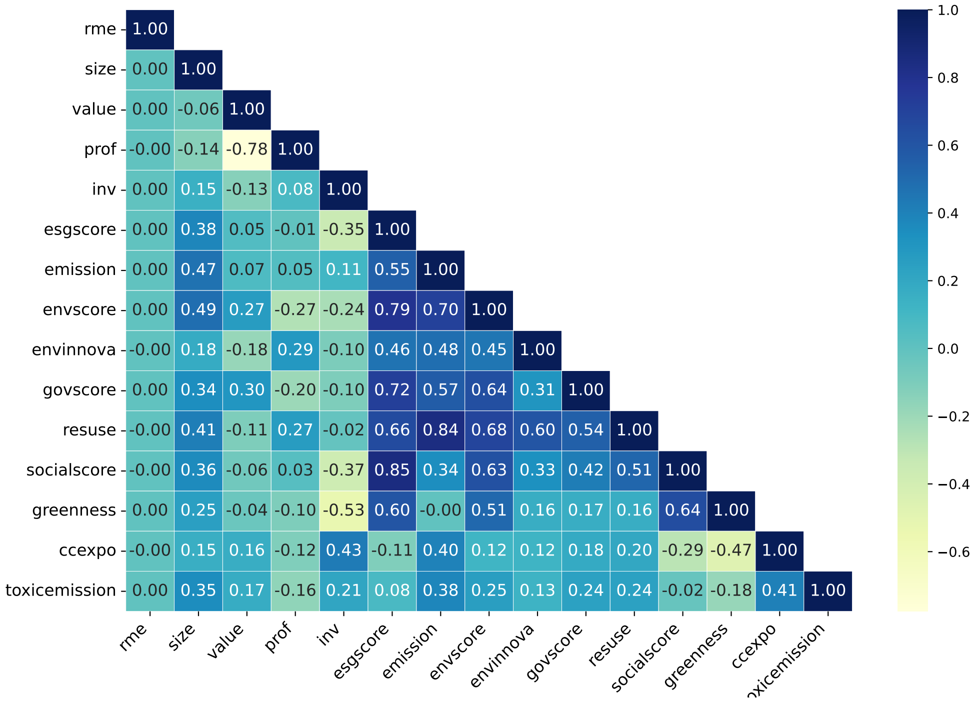
	Mean	Std Dev	Min	Median	Max
rme	0.03	1.19	-8.95	0.07	11.35
size	-0.01	0.93	-8.93	-0.02	7.86
value	0.00	0.81	-5.88	-0.00	6.16
prof	0.03	1.11	-9.94	0.03	7.14
inv	-0.02	1.05	-7.05	-0.02	6.94
esgscore	-0.01	0.16	-1.76	-0.01	0.93
emission	-0.01	0.18	-1.53	-0.00	1.60
envscore	-0.01	0.18	-1.71	-0.01	1.25
envinnova	-0.00	0.17	-1.25	-0.00	1.76
govscore	-0.01	0.14	-1.18	-0.01	1.25
resuse	-0.01	0.16	-1.60	-0.00	1.05
socialscore	-0.00	0.17	-1.87	-0.00	1.19
greenness	-0.01	0.24	-1.55	-0.01	2.02
ccexpo	-0.00	0.13	-0.78	-0.00	0.92
toxicemission	-0.00	0.19	-2.06	-0.00	1.60

It is apparent that most factors's mean values stay around zero, indicating a normalization that we demean the mean of each characteristic cross-sectionally in the data processing stage described in section 5.5.1. The standard deviation values reveal the volatility or risk associated with each factor. The traditional Fama-French factors such as market excess returns (rme) and profitability (prof) show relatively higher volatility than most ESG variables, where greenness and toxic emissions are comparatively less volatile with values 0.24 and 0.19, respectively. The contrast in volatility highlights the differing stability and risk profiles between traditional financial metrics and ESG considerations. The ESG factors demonstrate a smaller spread between the minimum and maximum values than the Fama-French five factors, indicating that ESG factors have less variability and are less likely to be exposed to outliers. Furthermore, the median values closely align with the means for most factors, suggesting a symmetric distribution of data points around the centre. However, slight deviations in some ESG scores, like emission and resource use scores where the medians are closer to zero than the means, indicate a skewed distribution of these factors.

[6] Please note that the paper does not use market excess returns (rme) to pricing kernel estimation as it is correlated with lots of pricing factors. We orthogonalize the studied factors to market returns.

The summary statistics of all factors use in the pricing kernel estimation are shown in Appendix 5.D.

The correlation matrix shown in figure 5.5.1 provides a visualized relationship between the 5 Fama-French factors and ESG variables. The size factor shows moderately positive correlations with most ESG variables such as ESG combined scores, emission, and environmental scores, suggesting that larger firms tend to have higher ESG ratings. The ESG variables present high correlations among themselves. ESG factors including ESG combined scores, emission, environmental scores, and social scores exhibit strong correlations with each other, especially notable between ESG combined scores and social scores with 0.85 covariance. It is because the ESG combined scores are a weighted combination of environmental scores, social scores and governance scores. The significant correlations observed among ESG variables and between ESG and traditional factors suggest a potential for multicollinearity if used together in traditional linear pricing kernel models.



	rme	size	value	prof	inv	esgscore	emission	envscore	envinnova	govscore	resuse	socialscore	greenness	ccexpo	oxicemission
rme	1.00														
size	0.00	1.00													
value	0.00	-0.06	1.00												
prof	-0.00	-0.14	-0.78	1.00											
inv	0.00	0.15	-0.13	0.08	1.00										
esgscore	0.00	0.38	0.05	-0.01	-0.35	1.00									
emission	0.00	0.47	0.07	0.05	0.11	0.55	1.00								
envscore	0.00	0.49	0.27	-0.27	-0.24	0.79	0.70	1.00							
envinnova	-0.00	0.18	-0.18	0.29	-0.10	0.46	0.48	0.45	1.00						
govscore	0.00	0.34	0.30	-0.20	-0.10	0.72	0.57	0.64	0.31	1.00					
resuse	-0.00	0.41	-0.11	0.27	-0.02	0.66	0.84	0.68	0.60	0.54	1.00				
socialscore	-0.00	0.36	-0.06	0.03	-0.37	0.85	0.34	0.63	0.33	0.42	0.51	1.00			
greenness	0.00	0.25	-0.04	-0.10	-0.53	0.60	-0.00	0.51	0.16	0.17	0.16	0.64	1.00		
ccexpo	-0.00	0.15	0.16	-0.12	0.43	-0.11	0.40	0.12	0.12	0.18	0.20	-0.29	-0.47	1.00	
toxicemission	0.00	0.35	0.17	-0.16	0.21	0.08	0.38	0.25	0.13	0.24	0.24	-0.02	-0.18	0.41	1.00

Fig. 5.5.1: Correlation matrix for factors

Figure 5.5.2 displays the variance inflation factor (VIF) values for all risk factors used in the SDF estimation. The VIF quantifies how much the variance of an estimated regression coefficient increases due to collinearity. A common threshold suggests that a VIF greater than 5 indicates distinct multicollinearity, and values above 10 suggest high multicollinearity (Davidson, 2004). Factors such as leverage (lev), value and profitability (prof) exhibit very high VIF values, indicating significant multicollinearity within the model. The ESG variables also show relatively high VIF values. For instance, ESG combined scores have a VIF of 17.04 and environmental scores reach 10.97, illustrating that ESG-related variables are also subject to multicollinearity.

Given the collinearity issues among these factors, traditional linear regression models may be inadequate for capturing the complex interrelations inherent in this data. Linear models struggle with the correlated explanatory variables. Nonetheless, neural networks are good at capturing and modelling the complex relationships and intricate interactions present in data through their hidden layers. Thus, neural networks becomes a more popular choice in asset pricing when analyzing ESG factors characterized by high multicollinearity.

Table 5.5.2: VIF of factors

Factor	VIF	Factor	VIF	Factor	VIF
size	10.00	lev	144.55	ivol	16.63
value	62.00	roaa	32.66	betaarb	6.94
prof	72.50	roea	34.08	season	1.42
valprof	13.48	sp	60.42	indrrev	13.22
fscore	3.89	gltnoa	1.48	indrrevlv	4.45
debtiss	5.43	mom	29.91	indmomrev	5.92
repurch	4.89	indmom	7.97	ciss	4.11
nissa	11.76	valmom	53.19	price	10.89
accruals	1.54	valmomprof	52.92	age	6.88
growth	8.14	shortint	6.49	shvol	15.36
aturnover	44.21	mom12	10.91	esgscore	17.04
gmargins	8.52	momrev	2.21	emission	9.00
divp	5.30	lrrev	4.21	envscore	10.97
ep	30.03	valuem	36.21	envinnova	2.04
cfp	18.82	nissm	10.99	govscore	4.98
noa	19.66	sue	4.40	resuse	8.37
inv	5.74	roe	46.25	socialscore	8.11
invcap	7.54	rome	12.33	greenness	6.73
igrowth	3.94	roa	38.21	ccexpo	2.23
sgrowth	5.85	strev	10.84	toxicemission	1.59

5.5.3 Model performance

Table 5.5.3 presents the quadratic values of pricing errors for candidate pricing kernels, including the linear model, Elastic Net, and neural networks with varying hidden layers from 1 to 5 hidden layers. The quadratic pricing errors, defined by the weighted squares of errors according to the criterion function in eq. (5.12), utilize the inverse matrix of second moments of factors as the weighting matrix. This method also refers to calculating the quadratic HJ distance between the specified and admissible pricing kernels. The in-sample column shows the quadratic pricing errors used as training loss, whereas the out-of-sample column describes these errors computed using unforeseen data, assessing the models' performance beyond the training dataset.

The in-sample loss for the linear model (Linear) is the smallest, as the analytical approach fits the training data to estimate linear pricing kernel loadings. However, this model exhibits a significant difference in performance between in-sample and out-of-sample data. The poor out-of-sample performance indicates a severe overfitting for the linear model with the factor zoo. As the dimension of input factors is high, the SDF loadings estimated by a cross-sectional regression are prone to high estimation errors, which are reflected in their out-of-sample performance.

The linear model regularised by the Elastic Net (Enet) is used to deal with overfitting. However, the Elastic Net yields the highest quadratic pricing errors either in-sample or out-of-sample. The unsatisfied performance of the Elastic Net might be due to its difficulty in identifying the difference between important factors and those that are redundant but still closely related to useful factors, especially in situations where there are many multi-correlated factors.

Machine learning methods demonstrate good out-of-sample performance compared to both the linear model and Elastic Net. This improvement in performance is because of the feature extraction and non-linear transformation of neural networks, which are able to capture complex patterns and avoid the multicollinearity in the data. With increasing hidden layers, the quadratic pricing errors are increasing for both in-sample and out-of-sample datasets. A shallow network architecture with one hidden layer looks capable of modelling the pricing kernel as shown in table 5.5.3. This finding will be further explored in the model specification tests.

Figure 5.5.2 illustrates the in-sample training and out-of-sample testing losses for the Elastic Net and neural networks (MLP1 to MLP5). We plot the training loss during training while recording the evaluated testing loss to see whether these models are prone to overfitting. We

Table 5.5.3: Squared pricing errors of pricing kernels

	In-sample	Out-of-sample
Linear	2.27e-12	1.29e-01
Enet	2.91e-02	1.38e-01
MLP1	3.46e-06	3.09e-05
MLP2	9.58e-06	7.93e-05
MLP3	9.92e-06	8.06e-05
MLP4	1.48e-05	1.13e-04
MLP5	2.93e-04	1.63e-03

tune the best hyperparameters by three-fold cross-validation using the in-sample dataset for all these models. Cross-validation evaluates the losses for three folds of data respectively to find the best set of hyper-parameters for each model. For Elastic Net, although the trends of both in-sample and out-of-sample loss are the same, its generalization ability is poor with the evidence of a high out-of-sample loss.

Despite concerns about over-parameterization, which typically increases with more hidden layers and hidden units potentially leading to overfitting, the minimal difference between in-sample and out-of-sample losses across the neural network models suggests that the overfitting problem has been effectively resolved due to the techniques used, including Cross-validation for hyperparameter tuning and dropout of some parameters randomly in the training process. Cross-validation reduces the overfitting of unseen data and obtains a realizable and stable estimation performance.

5.5.4 Linear or nonlinear?

The model specification tests compare linear pricing kernels to nonlinear ones. Specifically, we analyze the specification of the linear model and the Elastic Net against all nonlinear pricing kernels approximated by neural networks in $\mathcal{C}_{MLP}$, as illustrated in figure 5.5.3. In plot (a), we assess the performance of the linear model, while in plot (b), we evaluate the Elastic Net. The figure shows the test statistics, the distributions of test statistics and the 95% quantiles of distributions.

In figure 5.5.3 (a), the Bootstrap empirical distribution of the test statistic is consistently below zero, indicating that all neural networks yield smaller quadratic pricing errors than the

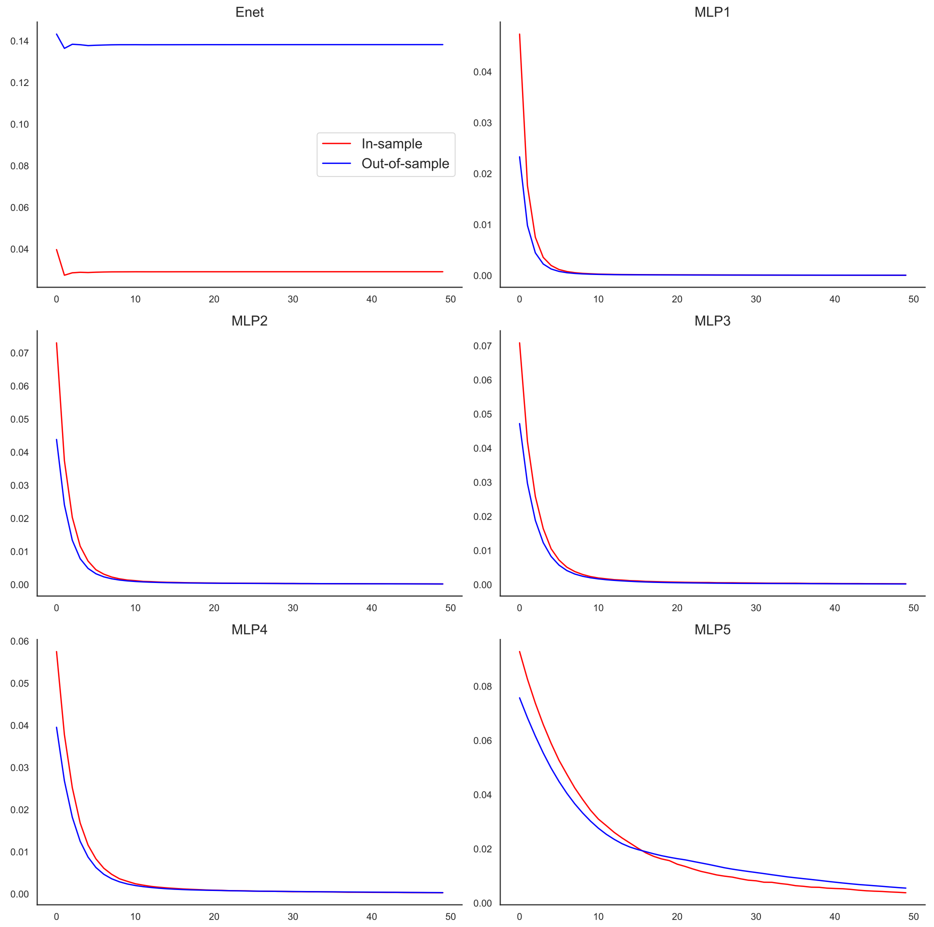
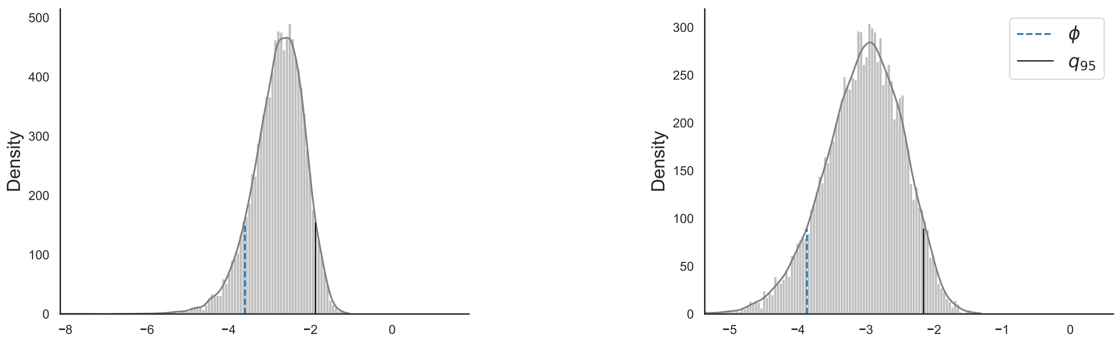


Fig. 5.5.2: Squared pricing errors over episodes

linear model. Also, the test statistic -3.60 is smaller than the critical value -1.88, which is the 95% quantile of its distribution. This comparison confirms that the null hypothesis cannot be rejected based on the observed data. Thus, figure 5.5.3 (a) proves that the nonlinear pricing kernel through neural networks provide a better specification than the linear model.

Additionally, figure 5.5.3 (b) presents the comparative analysis of the Elastic Net to neural networks. The Bootstrap distribution of the test statistic, similar to those in figure 5.5.3 (a), is positioned to the left of zero. Moreover, the values of $\phi = -3.87$ fall below the 95% quantile -2.15, illustrating that we cannot reject the null hypothesis. These results suggest that even when the linear model is regularized with the Elastic Net to mitigate overfitting, the linear pricing kernel incurs higher quadratic pricing errors than the nonlinear pricing kernel.

We tabulate the results of model specification tests in table 5.5.4. Each specification test applies a one-tailed distribution to critically assess the performance against hypothesized models. The specification tests compare test statistics against the upper bounds of the 95% critical interval (CI). These upper bounds serve as crucial benchmarks. Test statistics falling



(a) **(b)**

Fig. 5.5.3: Distribution of test statistics for model specification tests. The specification tests examine whether all nonlinear pricing kernels specify better than the linear model (a) and the Elastic Net (b). The test statistics ϕ are represented in the blue dotted line for the specification test. The 95% quantile q_{95} is outlined in black solid line to compare with the test statistics in the tests.

below these thresholds indicate that there is not enough evidence to reject the null hypothesis. The p-values indicate the likelihood of observing the sampling data, or more extreme, under the null hypothesis. According to table 5.5.4, since all derived p-values exceed the significance threshold of 5%, there is insufficient statistical evidence to reject the null hypothesis for any of the tested models. The results indicate that the nonlinear pricing kernel outstands the linear pricing kernel approximated in either the linear model or the Elastic Net.

Table 5.5.4: Model specification test results

	Test statistics	Upper bounds of 95% CI	P values
Specification test (a)	-3.60	-1.88	0.90
Specification test (b)	-3.87	-2.15	0.92

5.5.5 Which neural network?

The neural network selection is designed to find the optimal neural network configuration among nueral networks with hidden layers from 1 to 5. As depicted in table 5.5.3, MLP1 is hypothesized as the optimal model under the null hypothesis, because of its lowest in-sample squared pricing errors compared to other neural networks. The test statistic and the distribution under the null hypothesis are then calculated. Specifically, the test statistic δ represents the maximum difference in out-of-sample squared pricing errors between MLP1 and its competing models. This distribution is estimated through Block Bootstrap of the test statistic, denoted as δ^b.

Figure 5.5.4 depicts the test statistic and the empirical distribution of the bootstrap test statistic. The positioning of the test statistic lies left to the 95% quantile of the distribution. It suggests that the null hypothesis, which is MLP1 is the best model, cannot be rejected. Consequently, MLP1 has the smallest squared pricing errors among the competitive neural networks, confirming its outstanding performance in both model approximation and generalization.

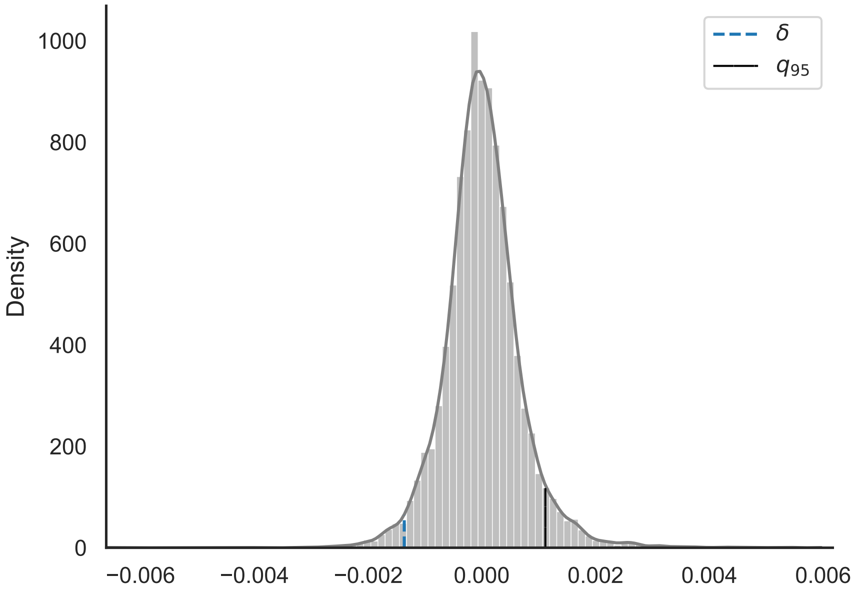


Fig. 5.5.4: Distribution of the test statistic for neural network selection. The blue dotted line, representing the test statistics, is positioned significantly to the left of the 95% quantile of the empirical bootstrap distribution, marked by the black solid line.

5.5.6 Significance of factors

Figure 5.5.5 shows the empirical distribution of statistics $\tau_i[h^*]$ and test statistics ξ_i of ESG portfolios evaluated via MLP1. The $\tau_i[h^*]$ samples are gathered together from 10000 times of h^* by sampling the Gaussian process. Each h^* sample is obtained from the argmax of a sample from the Gaussian process by sampling 500 neural networks with random weights. For the test statistics, we conduct ensemble learning to estimate the average test statistics ξ_i, and then scale them by the convergence rate r_T. The distributions of the aggregated $\tau_i[h^*]$ samples exhibited heavy right tails shown in the distribution for factors emission, innovation scores and toxic emission. It indicates that these distributions are similar to the χ^2 distributions.

In figure 5.5.5, the dotted, dashed and solid lines are the critical value at the 90%,95% and 99% quantile ($(1-100\alpha)\%$) of distribution. If the scaled test statistic of the ESG portfolio exceeds the critical value, the ESG portfolio is significant to the nonlinear pricing kernel. Otherwise, it is insignificant at a $100\alpha\%$ significance level. The figure shows that scaled test statistics of all ESG portfolios stay far right away from the 99% quantiles. It illustrates that these ESG portfolios are significant at 1% significant level, except for the environmental innovation scores which is significant at the 10% significance level. It shows that ESG variables are helpful to price the cross-sectional variations among the factor zoo.

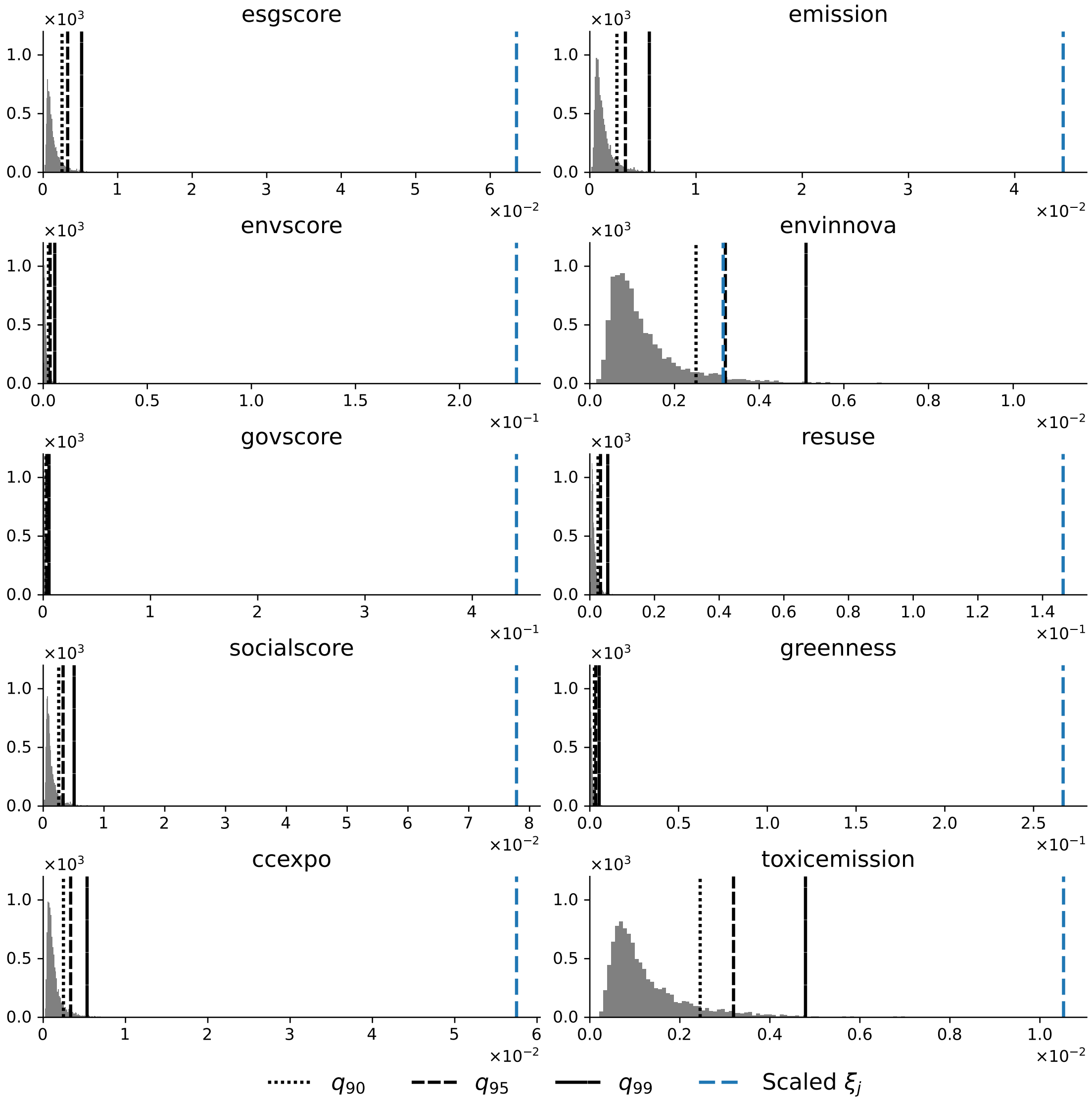


Fig. 5.5.5: Distribution of test statistics for ESG portfolios

Table 5.5.5 tabulates the t-statistics and corresponding p-values for all the factors considered in the pricing kernel estimation. For each factor across the neural networks, the t-statistics represent the averaged values of the scaled test statistics ξ_i obtained from 10 ensemble learning. The p-values are calculated based on the distribution of the $\tau[h^*]$ samples directly linked to the observed t-statistics. The factors are sorted according to their t-statistic values from high to low.

Remarkably, the top three factors which are governance scores, greenness and environment scores, are all ESG variables. The evidence reveals that the growing importance of ESG considerations and a substantial influence on asset pricing. Furthermore, nearly all ESG-related factors demonstrate statistical significance at the 1% significance level. The only exception is a factor called environmental innovation scores, which is significant at the 10% significance level. This slight deviation still suggests it is a significant but somewhat less important effect compared to the other ESG factors in pricing kernel estimation.

Table 5.5.5: Significance test of neural networks

	t-stat	p-value
govscore	0.4412***	0
greenness	0.2666***	0
envscore	0.2273***	0
indrrevlv	0.1745***	0
resuse	0.1464***	0
cfp	0.1112***	0
sue	0.0835***	0
socialscore	0.0779***	0
esgscore	0.0635***	0
ccexpo	0.0575***	0
indrrev	0.0532***	0
betaarb	0.0516***	0
indmomrev	0.0458***	0
emission	0.0446***	0
ivol	0.0422***	0
mom	0.0363***	0
lrrev	0.0357***	0
nissm	0.0325***	0
prof	0.0307***	0
repurch	0.0255***	0
debtiss	0.0232***	0
roea	0.0219***	0
igrowth	0.0209***	0
size	0.0190***	0
nissa	0.0174***	0
valuem	0.0155***	0
valmom	0.0141***	0
aturnover	0.0134***	0
sgrowth	0.0132***	0
divp	0.0126***	0
toxicemission	0.0105***	0
gltnoa	0.0090***	0.0001
value	0.0087***	0.0015
valprof	0.0075***	0.0014
age	0.0072***	0.0029
indmom	0.0062***	0.0041
valmomprof	0.0060***	0.0059
price	0.0055***	0.0077
roaa	0.0052**	0.0123
growth	0.0043**	0.0216
sp	0.0041**	0.0235
inv	0.0041**	0.0216
ep	0.0040**	0.0259
gmargins	0.0040**	0.0276
shvol	0.0035**	0.0473
envinnova	0.0031*	0.0522
momrev	0.0019	0.1910
lev	0.0016	0.2376
strev	0.0016	0.2549
shortint	0.0012	0.3930
ciss	0.0011	0.4358
season	0.0009	0.5873
roe	0.0006	0.8477
fscore	0.0004	0.9832
roa	0.0003	0.9946
invcap	0.0002	0.9991
rome	0.0002	0.9990
accruals	0.0002	1.0000
noa	0.0002	1.0000
mom12	0.0002	0.9999

In addition, 5.5.6 shows the ranking for the importance of factors based on their test statistics. The green bars in the figure represent the importance of the ESG portfolios. As clearly observed and discussed before, ESG portfolios consistently rank among the top factors. Conversely, the factors ranked at the bottom of the chart are barely visible due to their very low test statistics. This suggests that these factors have very little importance on the model, highlighting the contrast in importance between the top-ranking ESG factors and the least impactful variables.

The results of the significance test emphasize the critical importance of ESG factors among the factor zoo in asset pricing. It is the first study that includes ESG variables in the factor zoo in asset pricing. Surprisingly, these ESG variables are all significant in explaining the cross-sectional variations of asset prices. The prominence of these factors in the results suggests that investors and analysts should consider ESG variables when estimating pricing kernels and evaluating cross-sectional asset prices in stock trading and investment.

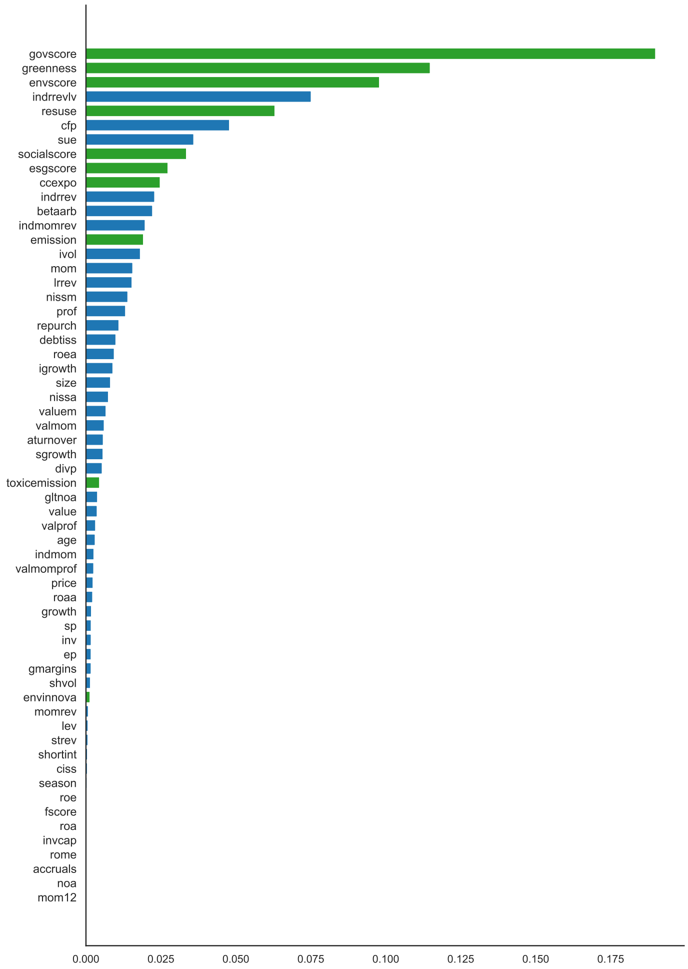


Fig. 5.5.6: Variable importance

5.5.7 Pricing kernel visualization

This section visualizes the time-series estimated pricing kernels using the MLP1 and examines the structure of the pricing kernels as functions of various factors.

Figure 5.5.7 illustrates that the pricing kernels during different economic periods, as recorded by NBER. The grey-shaded area represents the economic downturns, while the white area illustrates the economic expansions. The pricing kernels are represented in light blue and light red for those estimated by daily factor returns in-sample and out-of-sample, respectively. The deep-coloured lines represent the monthly averages of these daily pricing kernels.

The figure shows that the pricing kernel exhibits the highest volatility during economic downturns, especially at the end of 2008 when the 2008 financial crisis was ongoing. The increased volatility of pricing kernels during economic downturns is consistent with economic theory. During recessions, returns tend to be more volatile, and investments become increasingly unstable. This reflects the uncertainty in financial markets and fluctuations in investors' sentiments.

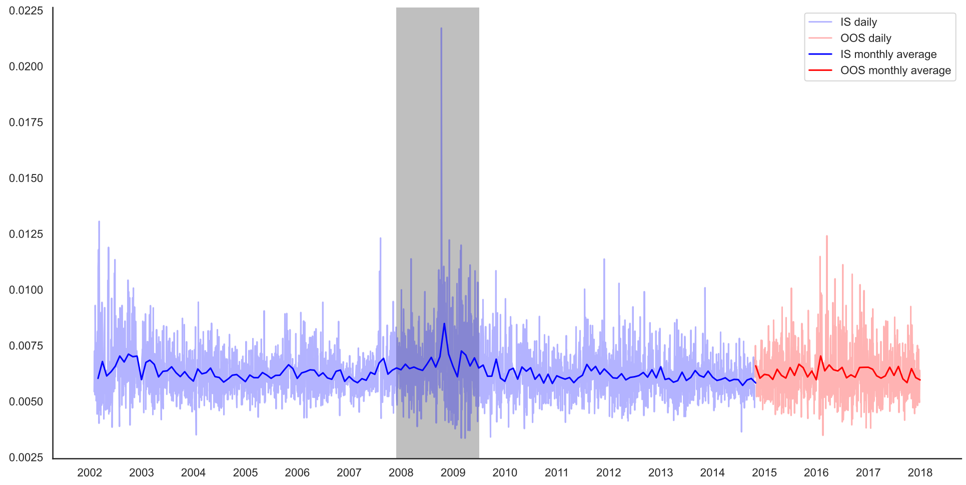


Fig. 5.5.7: Pricing kernel visualization

Additionally, figure 5.5.8 presents the time series of the top five factor returns, ranked by their importance. The results reveal that these factors also exhibit greater volatility during periods of economic downturn, with their maximum and minimum values significantly exceeding those observed during expansion periods. The time-series patterns in figure 5.5.7 along with figure 5.5.8 demonstrate that the estimated pricing kernels are effectively implied by the factor returns. The alignment of pricing kernel volatility with economic cycles and the corresponding behaviours of key factor returns reflect the effectiveness of MLP1 in capturing the dynamics of financial markets through data-driven pricing kernel estimation.

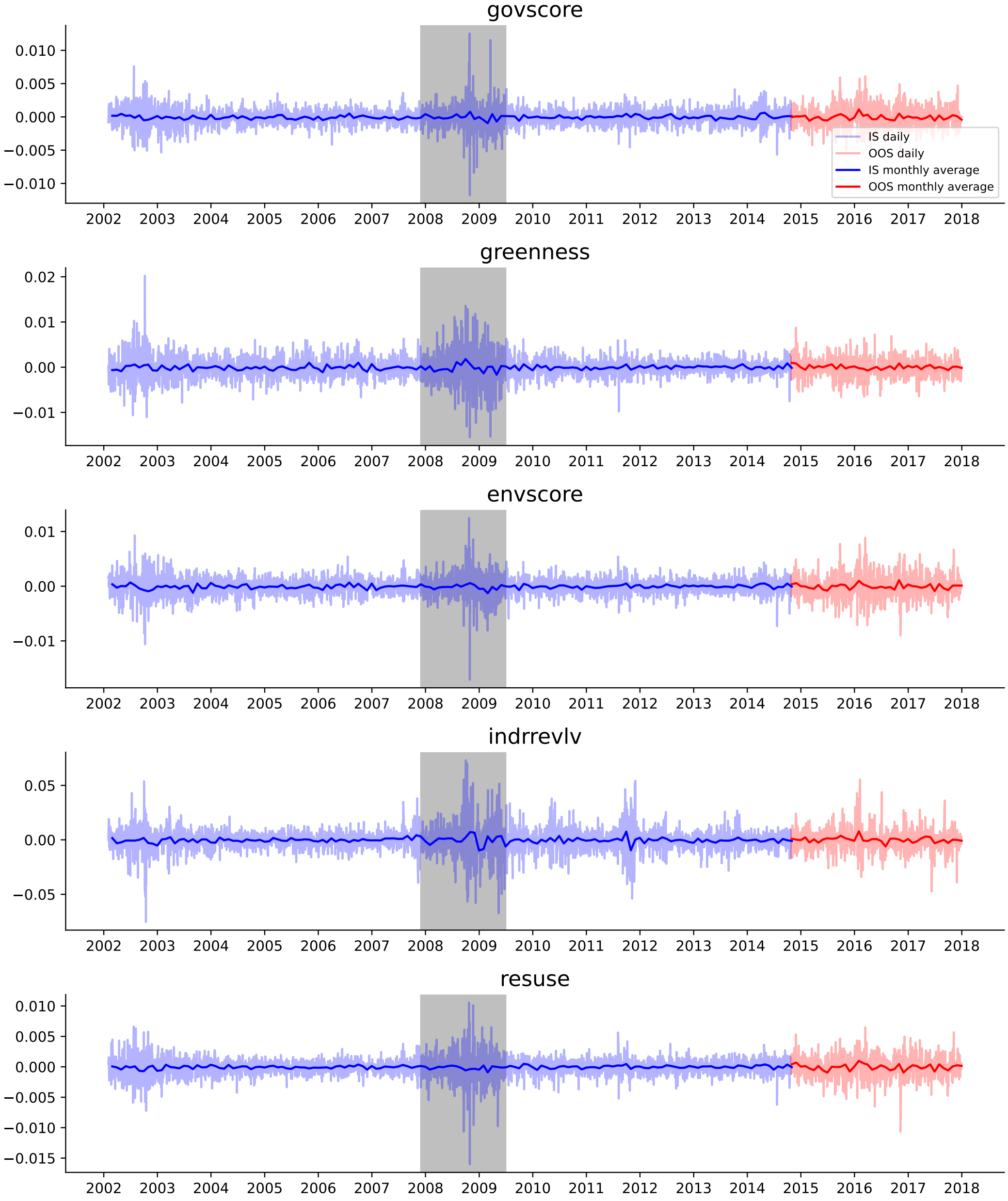


Fig. 5.5.8: Factor returns

Figure 5.5.9 illustrates the shape of the pricing kernel as a function of the two top-ranked ESG portfolios. They are greenness and governance scores. Also, we plot the pricing kernel function concerning the profitability and value within the MLP1 model. The relationships between the pricing kernel and these two ESG variables are positive, indicating that as ESG values increase, the pricing kernel also increases. Given that the pricing kernel is aligned with asset returns in the factor pricing model implied by the APT, the positive relationships suggest that higher ESG values are associated with higher expected asset returns. Additionally, the data indicates that companies with higher ESG values tend to perform better than those with lower ESG values.

Regarding the linearity of the pricing kernel, it exhibits a certain degree of nonlinearity of ESG variables and firm value within the MLP1 model. The nonlinearity between profitability and the pricing kernel is more obvious than other presented variables. It indicates that the relationship between the pricing kernel and the profitability is more complex than other presented variables to some extent. If more hidden layers are added to neural networks, the nonlinear patterns are more distinct due to the nonlinear transformation of the activation function.

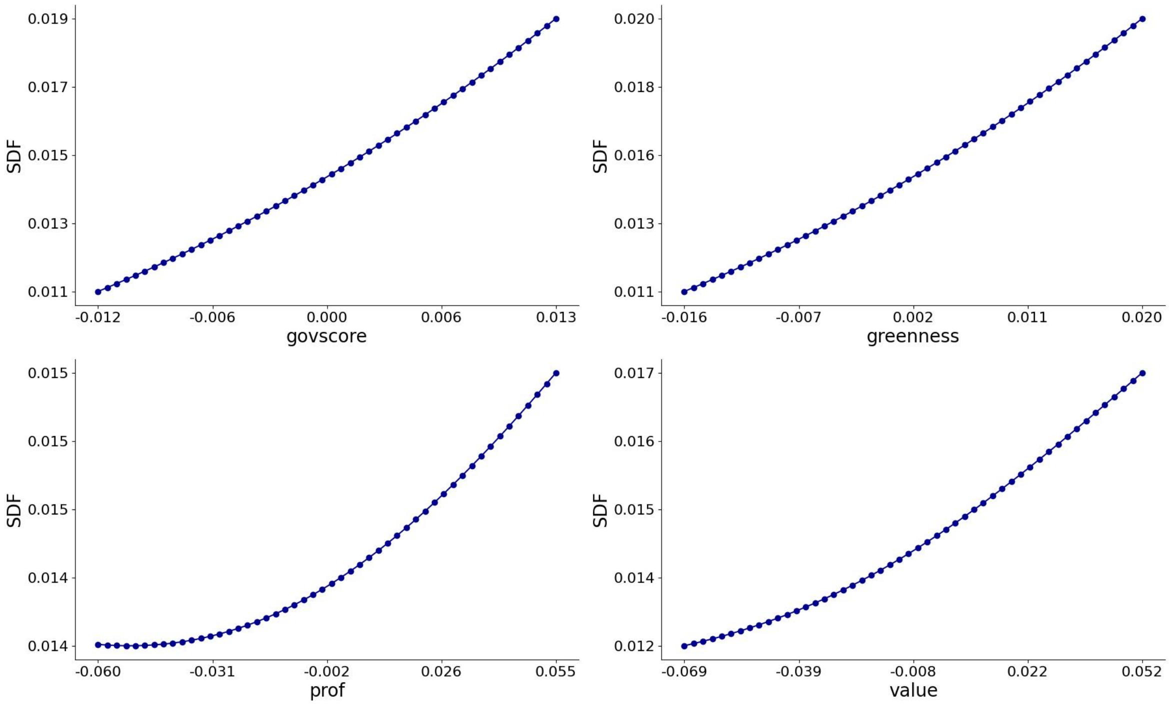


Fig. 5.5.9: Pricing kernel as a function of each ESG variable

In addition to examining the pricing kernel with respect to the single factor, we also explore the two-dimensional pricing kernel for MLP1, as shown in figure 5.5.10. This figure captures the interactions between firm size and the top two ESG variables, which are the governance scores and greenness. Also, we capture the effect of firm size with profitability or firm value on the pricing kernel. The highest pricing kernel values are observed in the scenarios where the firm size is the smallest, and the ESG variables are at their maximum levels. This finding is consistent with the Fama-French factor pricing models, which suggest that smaller firms tend to outperform larger ones. Also, firms with higher values have higher pricing kernels, especially when their size values are small. The results show that incorporating ESG variables into investment strategies improves returns, particularly for smaller companies. Our findings indicate that MLP1 captures the interactions among factors and the nonlinear relationships between factors and the pricing kernel.

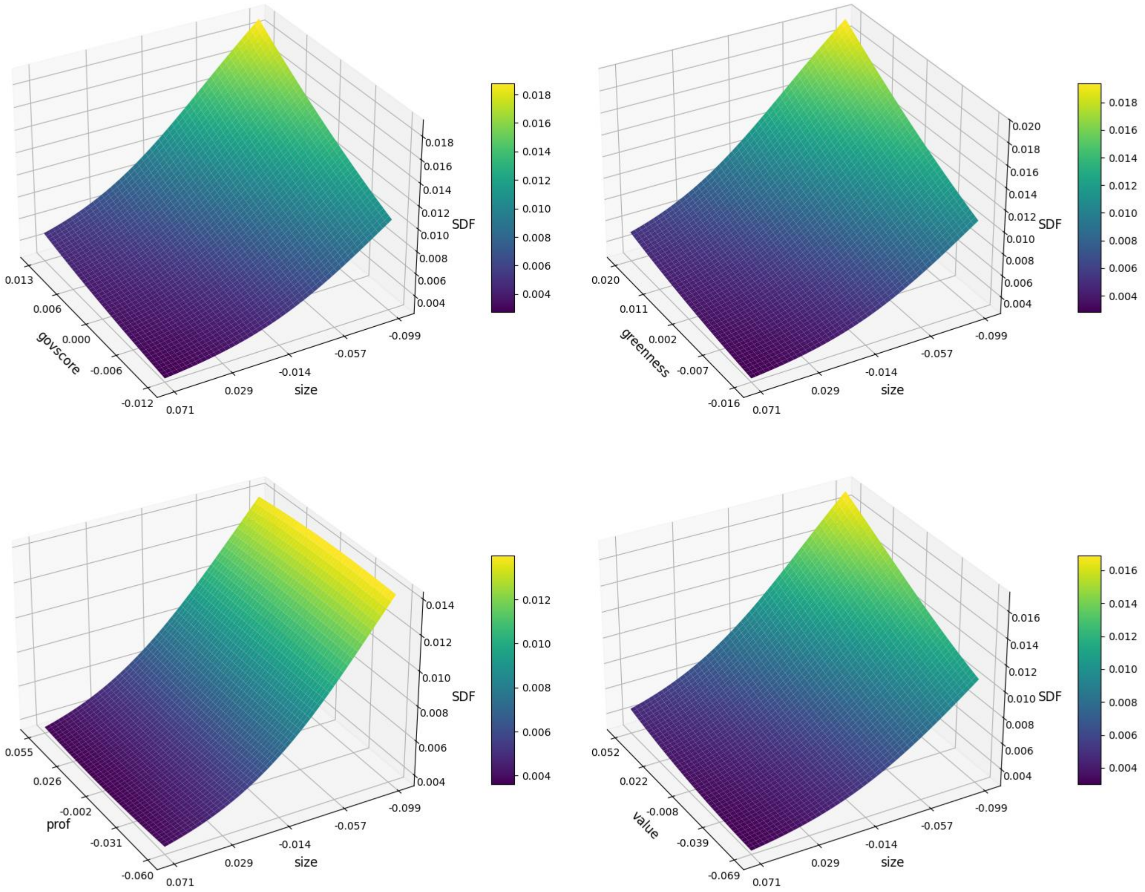


Fig. 5.5.10: Pricing kernel as a function of two factors

5.5.8 State-dependent performance

We evaluate the out-of-sample pricing performance of the linear and nonlinear SDFs across different economic states. These states are defined based on eight macroeconomic variables, as detailed in Welch and Goyal (2008). The variables include the dividend-price ratio (dp), earnings-price ratio (ep), book-to-market ratio (bm), net equity expansion (ntis), Treasury-bill rate (tbl), term spread (tms), default spread (dfy), and stock variance (svar) [7]. Each of these macroeconomic variables is chosen to represent the economic state because it has been documented as a predictor of the equity risk premium. By segmenting the sample according

[7] Data are collected from Amit Goyal's website: https://sites.google.com/view/agoyal145/home?authuser=0.

to whether these variables are above or below their medians of the samples, we determine whether the linear or nonlinear SDF is superior in pricing equities under varying economic conditions. For example, the “high bm” state comprises the state in which the bm value exceeds its median.

Table 5.5.6 presents the quadratic pricing errors for the linear SDF, the linear SDF regularized by Elastic Net, and the nonlinear SDF approximated via MLP1 under different states. Across all defined states, the nonlinear SDF (MLP1) consistently achieves the lowest quadratic pricing errors compared to the linear and Elastic Net models. For instance, under the high bm state, the quadratic pricing error of the linear SDF is approximately 0.22, while the nonlinear MLP1 error is only 5.49×10^{-5}. Similarly, in the high ep state, the linear SDF error reaches 1.24 and the Elastic Net model yields 0.93, whereas the nonlinear MLP1 reduces the error dramatically to 1.28×10^{-4}.

These results strongly imply that incorporating nonlinearities into the SDF enhances its ability to price assets in a state-dependent manner. In particular, the nonlinear SDF adapts to different financial market conditions more effectively than its linear candidates. This adaptability is evident in states characterized by the eight macro-economic predictors. The superior performance of the nonlinear SDF suggests that traditional linear models cannot capture the relationship between macroeconomic conditions and asset pricing, while a nonlinear SDF can exploit these complex relationships.

5.5.9 Robustness

This paper provides the robustness check for the estimated pricing kernel via MLP1. Different test assets are priced by the estimated pricing kernel to illustrate that our estimated pricing kernels are robust to explain variations for universal test assets. Table 5.5.7 shows there are three sets of test assets. The first one is 36 portfolios collected from Fama and French websites. They are 6 Portfolios Formed on Size and Book-to-Market, 6 Portfolios Formed on Size and Operating Profitability, 6 Portfolios Formed on Size and Investment, 6 Portfolios Formed on Size and Momentum based on prior 2 to 12 months’ returns, 6 Portfolios Formed on Size and Short-Term Reversal based on prior 0 to 1 month’s returns, and 6 Portfolios Formed on Size and Long-Term Reversal based on prior 13 to 60 months’ returns.

Table 5.5.6: State-dependent performance of SDFs

	Linear	Enet	MLP1
High bm	2.22e-01	2.28e-01	5.49e-05
Low bm	2.00e-01	2.13e-01	4.53e-05
High dfy	2.49e-01	2.63e-01	6.08e-05
Low dfy	2.03e-01	2.10e-01	4.67e-05
High dp	1.29e-01	1.44e-01	3.25e-05
Low dp	4.72e-01	4.62e-01	1.01e-04
High ep	1.24e+00	9.32e-01	1.28e-04
Low ep	1.31e-01	1.42e-01	3.17e-05
High ntis	9.63e-01	9.46e-01	2.06e-04
Low ntis	1.37e-01	1.49e-01	3.32e-05
High svar	2.55e-01	2.75e-01	6.32e-05
Low svar	2.04e-01	2.10e-01	4.64e-05
High tbl	3.14e-01	3.10e-01	7.05e-05
Low tbl	1.64e-01	1.82e-01	4.04e-05
High tms	-6.45e+00	2.23e+01	7.15e-03
Low tms	1.31e-01	1.40e-01	3.06e-05

Table 5.5.7: Robust check for the estimated pricing kernels to price test assets

	In-sample	Out-of-sample
36ff	5.76e-06	5.76e-06
49industry	3.62e-06	3.62e-06
118hxz	1.90e-05	1.90e-05

The second set of test assets is 49 industry portfolios from Fama-French's webiste[8]. The last one is the 118 factors collected from Hou et al. (2020)[9].

The quadratic pricing errors of test assets are very small and similar to the results of the 60 risk factors used in table 5.5.3. The minimal quadratic pricing errors indicate that the arbitrage opportunity is absent. The estimated pricing kernel is robust when pricing with a broad range of test assets in the financial market.

5.5.10 Investment performance

This section compares the performance of the outstanding neural network model MLP1 with traditional linear models in factor pricing to evaluate whether the nonlinear pricing kernel

[8] These data are available at: https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html.

[9] Hou et al. (2020)'s data are collected from: https://global-q.org/testingportfolios.html.

enhances investment performance over the linear kernel. By studying the *Sharpe ratio* (SR) of the factor pricing models, we can evaluate the investment performance of the candidate SDFs. The higher the SR, the smaller the abnormal returns generated in the factor pricing model. Thus, it is more likely that the pricing kernel can price the payoffs.

Unlike the managed portfolio constructed in the linear pricing kernel from eq. (5.6) is deemed as a pricing factor, the difficulty of nonlinear pricing kernel is that they cannot be treated as pricing factors directly. Alternatively, we treat the nonlinear pricing kernel as a non-traded factor so that we can construct the mimicking portfolios of the non-traded factor (Almeida and Freire, 2023). The regression between the non-traded factor $m(F_t)$ and test asset returns R_t^e is:

$$m(F_t) = \alpha + \beta' R_t^e + \varepsilon_t. \tag{5.31}$$

The nonlinear candidate pricing kernel $m(F_t)$ projects to test asset returns R_t^e and a constant α to obtain its mimicking portfolio. Therefore, the mimicking portfolio of the nonlinear pricing kernel is:

$$m_t^{np} = \hat{\beta}' R_t^e. \tag{5.32}$$

The SR of the mimicking portfolios of the nonlinear pricing kernels is therefore as follows:

$$SR(m_t^{np}) = \frac{\mathbb{E}[m_t^{np}]}{\sqrt{\Sigma(m_t^{np})}}. \tag{5.33}$$

In addition, we calculate the SR of the combination of traded factors, which is the same as the SR of linear SDFs, as they are treated as traded factors. That is:

$$SR(m_t^{lp}) = \frac{\mathbb{E}[F_t]}{\sqrt{\Sigma(F_t)}}. \tag{5.34}$$

Table 5.5.8 presents the annualized SR[10] for both linear pricing kernels and mimicking portfolios generated by various test assets for the nonlinear pricing kernel. Like what we use for the robust check in section 5.3.4, these test assets are 36 common factors, 49 industry factors and 118 single-sorted hxz factors. They are used to construct mimicking portfolios MLP1_36ff, MLP1_49industry and MLP1_118hxz, respectively. In addition to test assets, the 60 factors are also used to form the portfolios.

[10] The annualized SR is calculated by multiplying the daily SR by $\sqrt{252}$.

The SR of the linear pricing kernel model is recorded at 2.05, matching the SR of its traded factor. In contrast, the Elastic Net model displays a notably low SR of 0.96, highlighting its limitations in investment. On the other hand, MLP1 outperforms the linear models significantly, with all mimicking portfolios priced from MLP1 achieving higher SRs. Overall, the empirical evidence supports the superior performance of neural networks.

Table 5.5.8: Out-of-sample annualized SR.

Linear	Enet	MLP1_36ff	MLP1_49industry	MLP1_118hxz
2.05	0.96	3.36	3.75	8.72

Figure 5.5.11 displays the out-of-sample cumulative returns of mimicking portfolios derived from both the linear and nonlinear pricing kernels. To ensure comparability, all portfolios are adjusted to have the same volatility as the market returns. Among the candidates, the mimicking portfolios generated from the nonlinear pricing kernel via MLP1 exhibit the higher cumulative returns than the portfolios from the linear pricing kernel over time. Notably, the cumulative returns of the mimicking portfolios from 118 HXZ test assets are the most outstanding, aligned with the highest SR of the mimicking portfolios from HXZ test assets. Conversely, portfolios based on the Elastic Net and the linear model show lower cumulative returns, with the linear portfolios marginally outperforming those of the Elastic Net. However, both lag behind the nonlinear pricing kernel.

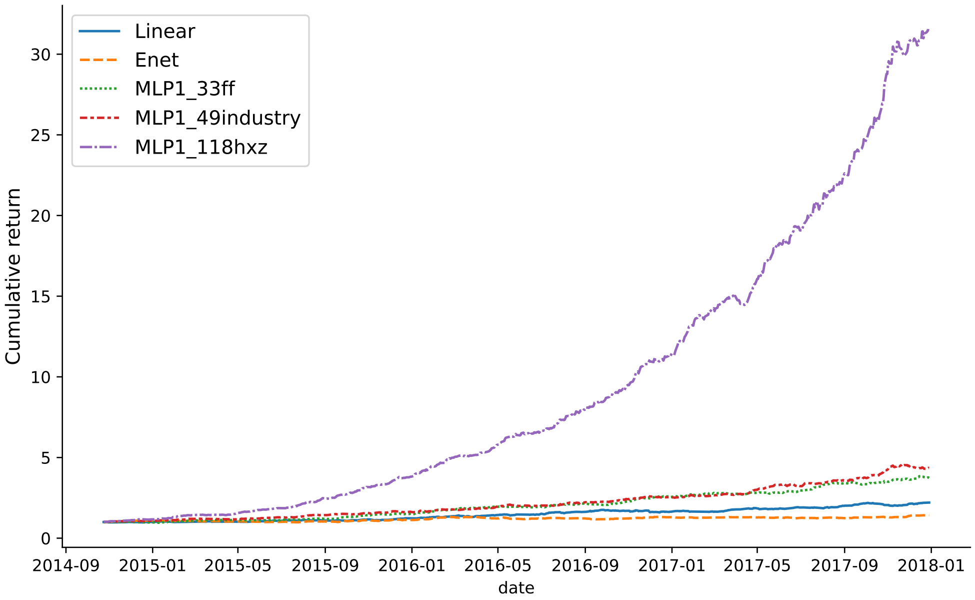


Fig. 5.5.11: Out-of-sample cumulative portfolio returns

5.6 Conclusion

We utilize neural networks to estimate the nonlinear pricing kernel. Unlike the linear pricing kernel, which supposes that the pricing kernel is linearly spanned by mean-variance efficient factors, our nonlinear specification allows for the inclusion of nonlinear factor components without the necessity of constructing efficient portfolios. By enforcing a non-negativity constraint on the candidate pricing kernel, our methodology ensures the estimation of asset prices without arbitrage opportunities. Significantly, this estimation method captures the cross-sectional variations attributable to the factors in SDF estimation rather than to the factor pricing models.

This paper assesses model specifications and performs the specification test between the linear pricing kernel approximated by the linear models and the nonlinear pricing kernel approximated by neural networks. The superiority of nonlinear specification leads to a general representation of the pricing kernel.

The neural network selection is conducted to find the optimal neural network configuration. Current machine learning papers in asset pricing typically compare the average model performance among various neural networks with different numbers of hidden layers. However, such approaches do not infer which model is robustly superior. Our research addresses this gap by demonstrating the critical necessity for neural network selection that accounts for the non-convex nature of neural networks in asset pricing.

Building upon the appropriately specified pricing kernel model using MLP1, we explore the importance of various factors in explaining cross-sectional variations. We employ the significance test designed for neural networks to identify statistically significant factors. Current research assesses the relative importance of variables. This method, while informative, does not clarify which factors are absolutely significant.

Our findings hold practical implications for empirical research. Firstly, we demonstrate that a nonlinear pricing kernel outperforms its linear counterpart in terms of quadratic pricing errors and Sharpe ratio. By comparing the quadratic pricing errors of the nonlinear pricing kernel, we observe the smallest in-sample pricing errors and robust out-of-sample performance. Moreover, by constructing mimicking portfolios of the nonlinear pricing kernel, we note that the annualized Sharpe ratio is higher than that of the linear pricing kernels. This suggests that the nonlinear pricing kernel has superior pricing and portfolio investment capabilities.

Secondly, the optimal neural network selection results show that MLP1 is the optimal neural network. This result is statistically robust when applied to various test assets. For future empirical research in asset pricing using neural networks, researchers can utilize MLP1 to estimate unknown functions, which are optimal both in terms of the universal approximation theory and the proposed statistical tests.

Thirdly, we have incorporated ESG variables into the factor zoo to estimate the pricing kernel. Through the significance test, we find that ESG variables significantly impact asset prices and rank highly. Due to the low signal-to-noise ratio of ESG and multicollinearity between ESG and traditional pricing factors, it is challenging to disentangle these effects in linear models. Our study leverages neural networks to simulate these interrelationships. Given the high level of attention that the public is currently paying increasing attention to ESG, this paper helps investors, stakeholders, and policymakers better understand the role of ESG in asset pricing through neural networks.

Appendices to chapter 5

5.A Sample splitting methods

In addition to cross-validation, there are some other sample-splitting methods, including the fixed scheme, rolling scheme, and recursive window scheme. The fixed scheme splits data into training, validation and testing samples. Then, it estimates the model once the training and validation samples are used, and then attempts to fix the trained model in the test set. Second, the rolling scheme shifts the training and validation sets forward in time to include more recent data but holds the window length of training and validation fixed. It refits the model at the time of each rolling, and evaluates the model performance on the unforeseen test samples. As the model is trained again using the data in the newly training set independently at each rolling, there are a few sets of weight parameters. Rolling approach is suitable to the model that is sensitive to data in the short-time, like stoch trading data. Third, the recursive window approach includes the data in the training set as new data arrive but retains the historical data in the training set. When it refits the model, it does not discard the previous trained model, but continues to update the pre-trained model using the recursive training set. The recursive approach is suitable for the model that utilize the long-term history of data.

5.B Hyperparameters

We select the optimal hyperparamers for each neural network via “Optuna”. The range of the number of hidden units is decided according to the growth rate of hidden units compared to the data sample. Also, the number of network parameters is smaller than the data sample to decide the optional number of hidden units in each layer. The options and optimal hyperparameters are listed in table 5.B.1 for each neural network. We select the optimal number of units from a range of potential units. The pre-defined set of units per layer cannot exceed the observation sizes T, in order to obtain model consistency. As shown in table 5.B.1, the optimal units per layer are decreasing as the number of hidden layers increases, indicating that the optimal model complexity for each network cannot be too large for estimating the pricing kernels. The dropout rate of the 5-hidden layer network is smaller, as its model complexity is smaller compared to other network configurations. The initialize learning rate is optimized to be the same.

Table 5.B.1: Hyperparameter tuning and optimization

Hidden layers	**Units per layer**		**Dropout rate**		**Learning rate**	
	Options	Optimal	Options	Optimal	Options	Optimal
1	$(4,45)$	42	$\{0.05,0.1,0.2,0.3\}$	0.1	$\{0.001,0.01\}$	0.01
2	$(4,30)$	30	$\{0.05,0.1,0.2,0.3\}$	0.1	$\{0.001,0.01\}$	0.01
3	$(4,24)$	24	$\{0.05,0.1,0.2,0.3\}$	0.1	$\{0.001,0.01\}$	0.01
4	$(4,21)$	19	$\{0.05,0.1,0.2,0.3\}$	0.1	$\{0.001,0.01\}$	0.01
5	$(4,19)$	4	$\{0.05,0.1,0.2,0.3\}$	0.05	$\{0.001,0.01\}$	0.01

5.C Description of factors

Table 5.C.1: Summary of financial factors

Abbreviation	Factor Name	Description	Reference
size	Size	Market capitalization as the end of June price times shares outstanding.	Fama and French (1993)
value	Value (annual)	Book equity to market equity ratio at the end of June each year.	Fama and French (1993)
prof	Gross Profitability	Gross profits over total assets, indicating operational efficiency.	Novy-Marx (2013)
valprof	Value-Profitability	Ranking combination of book-to-market and profitability.	Novy-Marx (2013)
fscore	Piotroski's F-score	Financial scoring system assessing profitability, funding, and efficiency.	Piotroski (2000)
debtiss	Debt Issuance	Indicator of whether long-term debt was issued during the year.	Spiess and Affleck-Graves (1999)
repurch	Share Repurchases	Indicates activity of share repurchases within the fiscal year.	Ikenberry et al. (1995)
nissa	Share Issuance (annual)	Yearly change in number of shares outstanding, excluding dividends and splits.	Pontiff and Woodgate (2008)
accruals	Accruals	Difference in earnings and cash from operations, adjusted for non-cash items.	Sloan (1996)

Continued on next page

Table 5.C.1 continued from previous page

Abbreviation	Factor name	Description	Reference
growth	Asset Growth	Growth rate of total assets from year t-1 to year t.	Cooper et al. (2008)
aturnover	Asset Turnover	Sales revenue relative to total assets.	Soliman (2008)
gmargins	Gross Margins	Gross profits relative to total sales.	Novy-Marx (2013)
divp	Dividend Yield	Dividend payments scaled by market equity, assessed in December.	Naranjo et al. (1998)
ep	Earnings/Price	Earnings relative to market value of equity, evaluated annually.	Basu (1977)
cfp	Cash Flow/Equity Market Value	Sum of net income and depreciation, scaled by market equity.	Lakonishok et al. (1994)
noa	Net Operating Assets	Operating assets over total assets minus financial liabilities.	Hirshleifer et al. (2004)
inv	Investment	Annual change in property, plant, and equipment plus inventory changes.	Lyandres et al. (2008)
invcap	Investment-to-Capital	Capital expenditures relative to total physical capital.	Xing (2008)
growth	Investment Growth	Invest- ment growth is the percentage change in capital expenditure.	Xing (2008)
sgrowth	Sales Growth	Annual sales growth, calculated as current year's sales over previous year's.	Lakonishok et al. (1994)
lev	Leverage	Total assets divided by market value of equity, measured annually.	Barbee Jr et al. (1996)

Continued on next page

Table 5.C.1 continued from previous page

Abbreviation	Factor name	Description	Reference
roaa	Return on Assets	Net income scaled by total assets, updated annually.	Chen et al. (2011)
roea	Return on Equity (annual)	Net income scaled by book value of equity, updated annually.	Haugen and Baker (1996)
sp	Sales-to-Price	Total revenues divided by stock price, updated annually.	Barbee Jr et al. (1996)
gltnoa	Growth in LT-NOA	Growth in Long-Term Net Operating Assets minus accruals.	Fairfield et al. (2003)
mom	Momentum (6m)	Cumulative return excluding the most recent month, over the previous six months.	Jagadeesh and Titman (1993)
indmom	Industry Momentum	Industry ranking based on past 6-month performance.	Moskowitz and Grinblatt (1999)
valmom	Value-Momentum	Combination of book-to-market and past 6-month returns.	Novy-Marx (2013)
valmomprof	Value-Momentum-Profitability	Aggregate ranking of book-to-market, profitability, and momentum.	Novy-Marx (2013)
shortint	Short Interest	Ratio of shares shorted to shares outstanding.	Dechow et al. (1998)
mom12	Momentum (1 year)	Cumulative return over the past year, skipping the most recent month.	Jagadeesh and Titman (1993)

Continued on next page

Table 5.C.1 continued from previous page

Abbreviation	Factor name	Description	Reference
momrev	Momentum-Reversal	Returns from a defined past period, used to predict reversals.	Jagadeesh and Titman (1993)
lrrev	Long-term Reversals	Long-term cumulative returns used to forecast market corrections.	DeBondt and Thaler (1985)
valuem	Value (monthly)	Monthly book-to-market ratio using the latest financial data.	Asness and Frazzini (2013)
nissm	Share Issuance (monthly)	Monthly change in share count, accounting for stock actions.	Pontiff and Woodgate (2008)
sue	PEAD (SUE)	Standardized unexpected earnings, reflecting surprises in quarterly reports.	Foster et al. (1984)
roe	Return on Book Equity	Quarterly net income divided by book equity from three months prior.	Chen et al. (2011)
rome	Return on Market Equity	Quarterly earnings scaled by market equity from four months prior.	Chen et al. (2011)
roa	Return on Assets	Quarterly net income relative to total assets, from three months prior.	Chen et al. (2011)
strev	Short-term Reversal	Return of the previous month, used to predict immediate reversals.	Jegadeesh (1990)
ivol	Idiosyncratic Volatility	Standard deviation of residuals from a firm-level return model.	Ang et al. (2006)

Continued on next page

Table 5.C.1 continued from previous page

Abbreviation	Factor name	Description	Reference
beta	Beta Arbitrage	Beta value calculated over the past 60 months, used in arbitrage strategies.	Cooper et al. (2008)
season	Seasonality	Average return from the same month over the previous five years.	Heston and Sadka (2008)
indrrev	Industry Relative Reversals	Difference between a stock's return and its industry's return, from the previous month.	Da et al. (2013)
indrrevlv	Industry Relative Reversals (Low Volatility)	As above, for stocks with below-median volatility.	Da et al. (2013)
indmomrev	Industry Momentum-Reversal	Combined ranking of industry momentum and relative reversals for low volatility stocks.	Moskowitz and Grinblatt (1999)
ciss	Composite Issuance	Log difference of market equity, adjusted for past returns.	Daniel and Titman (2006)
price	Price	Logarithm of market equity divided by shares outstanding.	Blume and Husic (1973)
age	Firm Age	Logarithm of months since a firm's listing in the CRSP database.	Barry and Brown (1984)
shvol	Share Volume	Average trading volume over the past three months, relative to outstanding shares.	Datar et al. (1998)

5.D Summary statistics

Table 5.D.1: Descriptive statistics of all factor returns (%)

Factor	Mean	Std Dev	Min	Median	Max
size	-0.01	0.93	-8.93	-0.02	7.86
value	0.00	0.81	-5.88	-0.00	6.16
prof	0.03	1.11	-9.94	0.03	7.14
valprof	0.02	0.91	-6.89	0.01	5.20
fscore	0.02	0.86	-6.77	0.00	5.50
debtiss	0.01	0.81	-6.19	0.04	5.45
repurch	0.03	0.79	-4.69	-0.00	5.24
nissa	-0.04	0.75	-5.19	-0.01	5.72
accruals	-0.01	0.92	-4.59	-0.02	5.86
growth	-0.03	0.81	-4.16	-0.01	8.96
aturnover	0.04	1.14	-10.90	0.05	8.74
gmargins	-0.02	0.97	-4.93	-0.00	7.94
divp	0.01	1.00	-8.51	0.01	11.44
ep	0.02	0.81	-5.36	0.01	4.96
cfp	0.02	0.81	-5.66	-0.00	4.92
noa	-0.00	0.78	-5.06	-0.03	4.66
inv	-0.02	1.05	-7.05	-0.02	6.94
invcap	-0.02	0.65	-6.84	-0.02	6.99
igrowth	-0.04	0.99	-5.72	-0.01	10.36
sgrowth	-0.03	0.86	-5.40	0.00	7.30
lev	-0.00	0.92	-7.15	-0.01	7.89
roaa	0.04	1.11	-9.33	0.03	7.71
roea	0.03	1.01	-5.08	0.00	7.70
sp	0.02	0.81	-7.86	0.02	4.59
gltnoa	-0.02	0.69	-3.54	-0.02	3.83
mom	-0.00	1.06	-7.52	0.05	6.93
indmom	0.01	0.96	-6.34	0.06	4.87
valmom	0.01	0.99	-6.07	0.06	4.69
valmomprof	0.01	1.00	-6.50	0.04	5.60

Continued on next page

Table 5.D.1 continued from previous page

Factor	Mean	Std Dev	Min	Median	Max
shortint	-0.01	0.90	-10.30	0.00	7.56
mom12	0.01	1.08	-7.34	0.07	7.21
momrev	0.01	0.95	-9.77	0.03	6.74
lrrev	-0.01	1.03	-8.19	0.02	8.30
valuem	0.00	0.86	-6.48	-0.03	6.13
nissm	-0.04	0.71	-4.55	-0.01	3.93
sue	0.03	1.04	-9.52	0.04	7.98
roe	0.04	0.96	-6.58	0.04	6.53
rome	0.04	0.75	-6.92	0.03	6.12
roa	0.04	1.01	-7.38	0.03	7.09
strev	0.00	1.02	-7.53	0.02	5.92
ivol	-0.03	0.68	-5.64	-0.01	3.98
betaarb	-0.03	0.68	-5.94	-0.03	5.30
season	-0.01	1.04	-6.23	0.00	5.30
indrrev	-0.00	1.00	-7.20	0.02	6.30
indrrevlv	-0.02	1.08	-7.54	-0.00	7.30
indmomrev	0.01	1.11	-8.97	0.03	7.22
ciss	-0.03	0.82	-6.83	-0.03	4.23
price	0.01	0.94	-6.30	0.03	6.17
age	0.01	0.63	-3.49	-0.02	5.49
shvol	-0.02	0.64	-5.04	-0.01	3.77
esgscore	-0.01	0.16	-1.76	-0.01	0.93
emission	-0.01	0.18	-1.53	-0.00	1.60
envscore	-0.01	0.18	-1.71	-0.01	1.25
envinnova	-0.00	0.17	-1.25	-0.00	1.76
govscore	-0.01	0.14	-1.18	-0.01	1.25
resuse	-0.01	0.16	-1.60	-0.00	1.05
socialscore	-0.00	0.17	-1.87	-0.00	1.19
greenness	-0.01	0.24	-1.55	-0.01	2.02
ccexpo	-0.00	0.13	-0.78	-0.00	0.92
toxicemission	-0.00	0.19	-2.06	-0.00	1.60

Chapter 6

Conclusion

This thesis examines portfolio optimization and asset pricing, focusing on the application of machine learning methods. Through three interconnected essays, the research collectively enhances the understanding and applications of advanced machine learning techniques in finance, addressing critical challenges and providing significant implications for both academia and industry.

Chapter 1 illustrates the link between machine learning and finance, and outlines the motivations for exploring robo-advisory portfolio optimization and asset pricing. Also, this chapter introduces the research questions, methodologies, main findings and contributions of each essay presented in this thesis. Regarding the advantages of machine leanring methods, they enable financial institutions to improve efficiency, reduce costs and offer sophisticated products to their clients. Moreover, applying machine learning to financial theories helps tackle issues that traditional methods cannot address, such as modelling complex and nonlinear relationships, as well as adapting to rapidly changing market conditions.

Chapter 2 provides a comprehensive literature review that forms the theoretical backbone of this thesis. The chapter covers critical areas of portfolio optimization, asset pricing, and machine learning methods. The chapter begins by discussing portfolio choices, especially the mean-variance optimization. It addresses the challenges of rare disasters and introduces the importance sampling as a technique to oversample disaster events. Risk aversion and inverse optimization concepts are examined to understand how investors' preferences influence portfolio choices. The literature review then shifts focus to asset pricing, discussing pricing kernels and the significance of ESG factors. Furthermore, the chapter summarizes machine learning methods in finance applications, presenting the definitions and applications of supervised, unsupervised, and reinforcement learning. Moreover, model selection and interpretability of neural networks are discussed.

Chapter 3 develops a novel computational framework that integrates RL with importance sampling in the presence of disaster events. The COVID-19 pandemic highlighted the vulnerabilities of traditional robo-advisors, which struggled to maintain satisfactory performance during unprecedented market downturns. Empirical evidence indicates that robo-advisors underperformed during such periods, emphasizing the need for algorithms that ensure reliability and robustness under extreme market conditions. By incorporating importance sampling into the RL framework, we ensure that robo-advisors acquire sufficient learning experiences to optimize investment strategies during rare disasters. This framework effectively reduces potential investment losses, leads to higher investor utilities and increases portfolio returns in the face of rare disasters.

However, this framework simplifies financial markets into three states, which may not fully capture the complexities of real-world markets where numerous features contribute to defining a state. Additionally, tabular RL methods face challenges in handling high-dimensional state and action spaces, limiting their generalization ability in complex environments. Addressing these limitations is crucial for enhancing the practical applicability of the framework. Future research can focus on the high-dimensional state spaces to model the dynamics of financial markets. Incorporating techniques such as deep RL algorithms can help manage the model complexity and improve performance.

Building upon the limitations identified in Chapter 3, Chapter 4 shifts focus on investors' risk aversion estimation under normal state space and disaster state space. The proposed framework combines online inverse optimization with the A2C deep RL algorithm to estimate state-dependent risk aversion and formulate adaptive investment strategies that align with individual risk profiles. Inverse optimization can inversely estimated the unknown risk aversion given the observable portfolio choices which are subject to noise such as behavioural biases, cognitive limitations and measurement errors. Moreover, we employ deep RL since it effectively handles the complexities and high dimensionality of financial environments.

Our findings demonstrate that the framework effectively estimates risk aversion for each selected mutual fund. This has important implications for robo-advisors and fund managers in aligning investment strategies with individual risk profiles, particularly during periods of market distress. By enhancing the alignment of investment strategies with investors' risk preferences, the framework improves portfolio performance across varying market conditions. The deep RL algorithm consistently outperforms equal-weighted portfolios, relevant investment-type allocations, and actual mutual funds, demonstrating its capability to generate exceptional investment strategies tailored to investors.

However, data scarcity during disaster states poses challenges in accurately estimating risk aversion. The limited sample size may affect estimation accuracy and the generalizability of the results. Future research can methods such as incorporate importance sampling, as utilized in Chapter 3, to address this data limitation and further refine investment strategies. Additionally, exploring solutions for addressing the imbalance between disaster and normal datasets, such as data augmentation techniques or synthetic data generation, can enhance the robustness of the estimation process.

Chapter 5 proposes a nonlinear pricing kernel approximated by neural networks, moving beyond traditional linear specifications in asset pricing models. Unlike the linear pricing kernel, the nonlinear specification allows for a span of nonlinear components of factors. By conducting model specification tests, this work pioneers a comprehensive approach to validate the nonlinear specification of the pricing kernel. Also, the optimal neural network selection finds the optimal neural network architecture with minimized out-of-sample quadratic pricing errors. Based on the selection of optimal neural network architecture, we evaluate the factor significance through a significance test designed for neural networks.

This essay contributes to the development of asset pricing studies. The findings suggest that a nonlinear pricing kernel outperforms its linear counterpart in terms of quadratic pricing errors and Sharpe ratios, indicating superior pricing and portfolio investment capabilities. By comparing the quadratic pricing errors of neural networks, we observe that the one-layer neural network has both the smallest in-sample and out-of-sample pricing errors. Constructing mimicking portfolios of the nonlinear pricing kernel, we note that the annualized Sharpe ratio is higher than that of linear pricing kernels, suggesting enhanced investment performance of the nonlinear stochastic discount factor. Moreover, the state-dependent performance of the nonlinear pricing kernel outstands the linear pricing kernel across all economic states.

The significance tests for neural networks highlight the importance of ESG variables. We find that ESG variables significantly impact asset prices and rank highly among factors based on the results of significance tests. This insight is valuable given the increasing attention on

ESG considerations in investment decisions. Incorporating ESG factors into asset pricing models not only aligns with the shift toward sustainable finance but also provides a more comprehensive understanding of the pricing factors. We find that ESG factors explain the cross-sectional asset prices under a range of pricing factors in the factor zoo.

Exploring the broader impact of ESG factors in asset pricing is interesting for further investigation. Future research can extend to how ESG factors interact with traditional financial variables. In the linear model, ESG factors are known to be linear-correlated with some traditional pricing factors. However, the correlation may be isolated when the features are mapped to high-dimensional space. Also, exploring the temporal dynamics, industry-specific effects, and geographic variations of ESG factors is worthwhile. Moreover, enhancing the interpretability of neural networks in finance is another important direction. We can dive into the explainable machine learning to make the black-box nature of neural networks more transparent, allowing practitioners and regulators to better understand financial decision-making.

The research on robo-advisory has several important policy implications. First, optimized investment strategies during rare economic states strengthen financial stability and protect investors in turbulent times. Policymakers and regulators may consider encouraging the use of these technologies to make financial markets more resilient. Second, adapting investment strategies to tailor investors' personalized risk profiles can make financial services more responsive to individual needs, increasing investor trust and investment satisfaction. Therefore, policymakers and practitioners should delicately assess investors' risk profiles before making investments on behalf of their clients.

The implications in asset pricing are also practically useful. By introducing a nonlinear pricing kernel, this research provides a more general pricing kernel for pricing various financial instruments such as derivatives and structured securities. Unlike traditional linear models, neural networks capture complex patterns in the data. This improvement motivates investors and portfolio managers to employ neural networks in estimating asset prices. Moreover, understanding how ESG factors affect asset prices can guide regulators to shape sustainable investment policies. At the same time, making AI-driven financial models more transparent is essential. Interpretable models help investors understand the reasons behind certain investment decisions, reduce the risks associated within "black-box" algorithms, and support more effective risk management.

In conclusion, this thesis advances the integration of machine learning techniques in finance, offering novel frameworks and methodologies for asset pricing and portfolio optimization. By addressing key challenges such as learning from rare disaster events, tailoring strategies to investor risk profiles, modelling the nonlinear pricing kernel, selecting the best neural network architecture and evaluating the significance of pricing factors, the research contributes valuable insights that bridge the gap between traditional financial theories and advanced machine learning methods.

Bibliography

Abraham, F., Schmukler, S. L., and Tessada, J. (2019). Robo-advisors: Investing through machines. *World Bank Research and Policy Briefs*, 21:134881.

Acharya, V. V., Pedersen, L. H., Philippon, T., and Richardson, M. (2017). Measuring systemic risk. *The review of financial studies*, 30(1):2–47.

Akbarzadeh, N., Tekin, C., and Van Der Schaar, M. (2018). Online learning in limit order book trade execution. *IEEE Transactions on Signal Processing*, 66(17):4626–4641.

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631.

Almahdi, S. and Yang, S. Y. (2017). An adaptive portfolio trading system: A risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown. *Expert Systems with Applications*, 87:267–279.

Almeida, C. and Freire, G. (2023). Which (nonlinear) factor models? *Available at SSRN 4421179*.

Almeyda, R. and Darmansya, A. (2019). The influence of environmental, social, and governance (ESG) disclosure on firm financial performance. *IPTEK Journal of Proceedings Series*, (5):278–290.

Alsabah, H., Capponi, A., Ruiz Lacedelli, O., and Stern, M. (2021). Robo-advising: Learning investors' risk preferences via portfolio choices. *Journal of Financial Econometrics*, 19(2):369–392.

Anders, U. and Korn, O. (1999). Model selection in neural networks. *Neural networks*, 12(2):309–323.

Anderson, E. W., Hansen, L. P., and Sargent, T. J. (2003). A quartet of semigroups for model specification, robustness, prices of risk, and model detection. *Journal of the European Economic Association*, 1(1):68–123.

Ang, A., Hodrick, R. J., Xing, Y., and Zhang, X. (2006). The cross-section of volatility and expected returns. *The journal of finance*, 61(1):259–299.

Ardia, D., Bluteau, K., Boudt, K., and Inghelbrecht, K. (2022). Climate change concerns and the performance of green vs. brown stocks. *Management Science*.

Asness, C. S. and Frazzini, A. (2013). The devil in hml's details. *Journal of Portfolio Management*, 39(4):49–68.

Asness, C. S., Frazzini, A., and Pedersen, L. H. (2019). Quality minus junk. *Review of Accounting studies*, 24(1):34–112.

Aswani, A., Shen, Z.-J., and Siddiq, A. (2018). Inverse optimization with noisy data. *Operations Research*, 66(3):870–892.

Bansal, R. and Viswanathan, S. (1993). No arbitrage and arbitrage pricing: A new approach. *The Journal of Finance*, 48(4):1231–1262.

Barbee Jr, W. C., Mukherji, S., and Raines, G. A. (1996). Do sales–price and debt–equity explain stock returns better than book–market and firm size? *Financial Analysts Journal*, 52(2):56–60.

Barro, R. J. (2006). Rare disasters and asset markets in the twentieth century. *The Quarterly Journal of Economics*, 121(3):823–866.

Barry, C. B. and Brown, S. J. (1984). Differential information and the small firm effect. *Journal of financial economics*, 13(2):283–294.

Basu, S. (1977). Investment performance of common stocks in relation to their price-earnings ratios: A test of the efficient market hypothesis. *The Journal of Finance*, 32(3):663–682.

Baulkaran, V. and Jain, P. (2023). Who uses robo-advising and how? *Financial Review*, 58(1):65–89.

Beketov, M., Lehmann, K., and Wittke, M. (2018). Robo advisors: quantitative methods inside the robots. *Journal of Asset Management*, 19(6):363–370.

Bennani, L., Le Guenedal, T., Lepetit, F., Ly, L., Mortier, V., Roncalli, T., and Sekine, T. (2018). How ESG investing has impacted the asset pricing in the equity market. *Available at SSRN 3316862*.

Bertsekas, D. P. (1997). Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334.

Best, M. J. and Grauer, R. R. (1991). On the sensitivity of mean-variance-efficient portfolios to changes in asset means: some analytical and computational results. *The review of financial studies*, 4(2):315–342.

Blume, M. E. and Husic, F. (1973). Price, beta, and exchange listing. *The Journal of Finance*, 28(2):283–299.

Bolton, P. and Kacperczyk, M. (2021). Do investors care about carbon risk? *Journal of financial economics*, 142(2):517–549.

Bonaccolto, G. and Paterlini, S. (2020). Developing new portfolio strategies by aggregation. *Annals of Operations Research*, 292(2):933–971.

Brenner, L. and Meyll, T. (2020). Robo-advisors: A substitute for human financial advice? *Journal of Behavioral and Experimental Finance*, 25:100275.

Bryzgalova, S., Pelger, M., and Zhu, J. (2019). Forest through the trees: Building cross-sections of stock returns. *Available at SSRN 3493458*.

Bucciol, A. and Miniaci, R. (2011). Household portfolios and implicit risk preference. *Review of Economics and Statistics*, 93(4):1235–1250.

Campbell, J. Y. (2006). Household finance. *The journal of finance*, 61(4):1553–1604.

Capponi, A., Olafsson, S., and Zariphopoulou, T. (2022). Personalized robo-advising: Enhancing investment through client interaction. *Management Science*, 68(4):2485–2512.

Chan, T. C., Mahmood, R., and Zhu, I. Y. (2023). Inverse optimization: Theory and applications. *Operations Research*.

Chapman, D. A. (1997). Approximating the asset pricing kernel. *The Journal of Finance*, 52(4):1383–1410.

Chen, L., Novy-Marx, R., and Zhang, L. (2011). An alternative three-factor model. *Available at SSRN 1418117*.

Chen, L., Pelger, M., and Zhu, J. (2023). Deep learning in asset pricing. *Management Science*, 0(0):1–37.

Chen, Q. and Liu, X.-Y. (2020). Quantifying ESG alpha using scholar big data: an automated machine learning approach. In *Proceedings of the First ACM International conference on AI in finance*, pages 1–8.

Chen, X. and Ludvigson, S. C. (2009). Land of addicts? an empirical investigation of habit-based asset pricing models. *Journal of Applied Econometrics*, 24(7):1057–1093.

Chetty, R. (2006). A new method of estimating risk aversion. *American Economic Review*, 96(5):1821–1834.

Cochrane, J. H. (1996). A cross-sectional test of an investment-based asset pricing model. *Journal of Political Economy*, 104(3):572–621.

Cochrane, J. H. (2009). *Asset pricing: Revised edition*. Princeton university press.

Cooper, M. J., Gulen, H., and Schill, M. J. (2008). Asset growth and the cross-section of stock returns. *The Journal of Finance*, 63(4):1609–1651.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

Da, Z., Liu, Q., and Schaumburg, E. (2013). A closer look at the short-term return reversal. *Management Science*, 60(3):658–674.

Dai, M., Jin, H., Kou, S., and Xu, Y. (2021). Robo-advising: A dynamic mean-variance approach. *Digital Finance*, 3(2):81–97.

D'Amato, V., D'Ecclesia, R., and Levantesi, S. (2022). Firms' profitability and ESG score: A machine learning approach. *Applied Stochastic Models in Business and Industry*.

Damodaran, A. (1999). Estimating risk free rates. *WP, Stern School of Business, New York*.

Damodaran, A. (2019). Equity risk premiums (erp): Determinants, estimation and implications–the 2019 edition. *NYU Stern School of Business*.

Daniel, K. and Titman, S. (2006). Market reactions to tangible and intangible information. *Journal of Finance*, 61:1605–1643.

Dann, C., Neumann, G., and Peters, J. (2014). Policy evaluation with temporal differences: A survey and comparison. *Journal of Machine Learning Research*, 15:809–883.

Datar, V. T., Naik, N. Y., and Radcliffe, R. (1998). Liquidity and stock returns: An alternative test. *Journal of Financial Markets*, 1(2):203–219.

Davidson, R. (2004). Econometric theory and methods.

DeBondt, W. F. and Thaler, R. (1985). Does the stock market overreact? *Journal of Finance*, 40:793–805.

Dechow, P. M., Kothari, S. P., and Watts, R. L. (1998). The relation between earnings and cash flows. *Journal of Accounting and Economics*, 25(2):133–168.

Deng, Y., Bao, F., Kong, Y., Ren, Z., and Dai, Q. (2016). Deep direct reinforcement learning for financial signal representation and trading. *IEEE transactions on neural networks and learning systems*, 28(3):653–664.

Dittmar, R. F. (2002). Nonlinear pricing kernels, kurtosis preference, and evidence from the cross section of equity returns. *The Journal of Finance*, 57(1):369–403.

Dohmen, T., Falk, A., Huffman, D., Sunde, U., Schupp, J., and Wagner, G. G. (2011). Individual risk attitudes: Measurement, determinants, and behavioral consequences. *Journal of the european economic association*, 9(3):522–550.

Dong, C., Chen, Y., and Zeng, B. (2018). Generalized inverse optimization through online learning. *Advances in Neural Information Processing Systems*, 31.

Dong, Z.-L., Zhu, M.-X., and Xu, F.-M. (2022). Robo-advisor using closed-form solutions for investors' risk preferences. *Applied Economics Letters*, 29(16):1470–1477.

Duchin, R. and Harford, J. (2021). The COVID-19 crisis and the allocation of capital. *Journal of Financial and Quantitative Analysis*, 56(7):2309–2319.

D'Acunto, F., Prabhala, N., and Rossi, A. G. (2019). The promises and pitfalls of robo-advising. *The Review of Financial Studies*, 32(5):1983–2020.

Engle, R. F., Giglio, S., Kelly, B., Lee, H., and Stroebel, J. (2020). Hedging climate change news. *The Review of Financial Studies*, 33(3):1184–1216.

Fabozzi, F. J. and Fabozzi, F. A. (2021). *Bond markets, analysis, and strategies*. MIT Press.

Fairfield, P. M., Whisenant, J. S., and Yohn, T. L. (2003). Accrued earnings and growth: Implications for future profitability and market mispricing. *The accounting review*, 78(1):353–371.

Fallahgoul, H., Franstianto, V., and Lin, X. (2024). Asset pricing with neural networks: Significance tests. *Journal of Econometrics*, 238(1):105574.

Fama, E. F. and French, K. R. (1992). The cross-section of expected stock returns. *the Journal of Finance*, 47(2):427–465.

Fama, E. F. and French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of financial economics*, 33(1):3–56.

Fama, E. F. and French, K. R. (1996). Multifactor explanations of asset pricing anomalies. *The journal of finance*, 51(1):55–84.

Farrell, M. H., Liang, T., and Misra, S. (2021). Deep neural networks for estimation and inference. *Econometrica*, 89(1):181–213.

Feng, G., Giglio, S., and Xiu, D. (2020). Taming the factor zoo: A test of new factors. *The Journal of Finance*, 75(3):1327–1370.

Foerster, S., Linnainmaa, J. T., Melzer, B. T., and Previtero, A. (2017). Retail financial advice: does one size fit all? *The Journal of Finance*, 72(4):1441–1482.

Foster, G., Olsen, C., and Shevlin, T. (1984). Earnings releases, anomalies, and the behavior of security returns. *Accounting Review*, pages 574–603.

Frank, J., Mannor, S., and Precup, D. (2008). Reinforcement learning in the presence of rare events. In *Proceedings of the 25th International Conference on Machine Learning*, pages 336–343.

Frank, J. W. (2009). *Reinforcement learning in the presence of rare events*. PhD thesis, McGill University.

Freyberger, J., Neuhierl, A., and Weber, M. (2020). Dissecting characteristics nonparametrically. *The Review of Financial Studies*, 33(5):2326–2377.

Gan, L. Y., Khan, M. T. I., and Liew, T. W. (2021). Understanding consumer's adoption of financial robo-advisors at the outbreak of the COVID-19 crisis in Malaysia. *Financial Planning Review*, 4(3):1127.

Gao, X. and Xu, T. (2022). Order scoring, bandit learning and order cancellations. *Journal of Economic Dynamics and Control*, 134:104287.

Giudici, P., Polinesi, G., and Spelta, A. (2022). Network models to improve robot advisory portfolios. *Annals of Operations Research*, 313(2):1–25.

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

Grable, J. E. (2000). Financial risk tolerance and additional factors that affect risk taking in everyday money matters. *Journal of business and psychology*, 14:625–630.

Gu, S., Kelly, B., and Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273.

Gu, S., Kelly, B., and Xiu, D. (2021). Autoencoder asset pricing models. *Journal of Econometrics*, 222(1):429–450.

Halbritter, G. and Dorfleitner, G. (2015). The wages of social responsibility—where are they? a critical review of ESG investing. *Review of Financial Economics*, 26:25–35.

Hambly, B., Xu, R., and Yang, H. (2023). Recent advances in reinforcement learning in finance. *Mathematical Finance*, 33(3):437–503.

Hansen, L. P. and Jagannathan, R. (1991). Implications of security market data for models of dynamic economies. *Journal of political economy*, 99(2):225–262.

Hansen, L. P. and Jagannathan, R. (1997). Assessing specification errors in stochastic discount factor models. *The Journal of Finance*, 52(2):557–590.

Hansen, L. P. and Richard, S. F. (1987). The role of conditioning information in deducing testable restrictions implied by dynamic asset pricing models. *Econometrica: Journal of the Econometric Society*, pages 587–613.

Hansen, L. P. and Singleton, K. J. (1982). Generalized instrumental variables estimation of nonlinear rational expectations models. *Econometrica: Journal of the Econometric Society*, pages 1269–1286.

Harrison, J. M. and Kreps, D. M. (1979). Martingales and arbitrage in multiperiod securities markets. *Journal of Economic theory*, 20(3):381–408.

Haugen, R. A. and Baker, N. L. (1996). Commonality in the determinants of expected stock returns. *Journal of Financial Economics*, 41:401–439.

Heston, S. L. and Sadka, R. (2008). Seasonality in the cross-section of stock returns. *Journal of Financial Economics*, 87(2):418–445.

Hinich, M. J. (2003). Risk when some states are low-probability events. *Macroeconomic Dynamics*, 7(4):636–643.

Hirshleifer, D., Hou, K., Teoh, S. H., and Zhang, Y. (2004). Do investors overvalue firms with bloated balance sheets. *Journal of Accounting and Economics*, 38:297–331.

Horel, E. and Giesecke, K. (2020). Significance tests for neural networks. *Journal of Machine Learning Research*, 21(227):1–29.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

Hou, K., Xue, C., and Zhang, L. (2020). Replicating anomalies. *The Review of financial studies*, 33(5):2019–2133.

Hsu, P.-H., Li, K., and Tsou, C.-Y. (2023). The pollution premium. *The Journal of Finance*, 78(3):1343–1392.

Ikenberry, D., Lakonishok, J., and Vermaelen, T. (1995). Market underreaction to open market share repurchases. *Journal of financial economics*, 39(2-3):181–208.

Jagadeesh, N. and Titman, S. (1993). Returns to buying winners and selling losers: Implications for stock market efficiency. *Journal of Finance*, 48:65–91.

Jegadeesh, N. (1990). Evidence of predictable behavior of security returns. *Journal of Finance*, 45:881–898.

Jiang, Z., Xu, D., and Liang, J. (2017). A deep reinforcement learning framework for the financial portfolio management problem. *arXiv*, 1706.10059.

Juneja, S. and Shahabuddin, P. (2006). Rare-event simulation techniques: An introduction and recent advances. *Handbooks in operations research and management science*, 13:291–350.

Kahneman, D. and Tversky, A. (2013). Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part I*, pages 99–127. World Scientific.

Karoui, A. (2013). The asset allocation of managers and investors: Evidence from hybrid funds. *The Journal of Wealth Management*, 16(3):69–81.

Keffert, H. (2024). Robo-advising: Optimal investment with mismeasured and unstable risk preferences. *European Journal of Operational Research*, 315(1):378–392.

Kelly, B. T., Pruitt, S., and Su, Y. (2019). Characteristics are covariances: A unified model of risk and return. *Journal of Financial Economics*, 134(3):501–524.

Khalilpourazari, S. and Hashemi Doulabi, H. (2021). Designing a hybrid reinforcement learning based algorithm with application in prediction of the COVID-19 pandemic in quebec. *Annals of Operations Research*, 312(2):1–45.

Kim, K.-j. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1-2):307–319.

Koijen, R. S. and Yogo, M. (2015). *An equilibrium model of institutional demand and asset prices*. National Bureau of Economic Research.

Konda, V. and Tsitsiklis, J. (1999). Actor-critic algorithms. *Advances in neural information processing systems*, 12.

Konno, H. and Kobayashi, K. (1997). An integrated stock-bond portfolio optimization model. *Journal of Economic Dynamics and Control*, 21(8-9):1427–1444.

Konno, H. and Yamazaki, H. (1991). Mean-absolute deviation portfolio optimization model and its applications to Tokyo stock market. *Management science*, 37(5):519–531.

Kozak, S., Nagel, S., and Santosh, S. (2018). Interpreting factor models. *The Journal of Finance*, 73(3):1183–1223.

Kozak, S., Nagel, S., and Santosh, S. (2020). Shrinking the cross-section. *Journal of Financial Economics*, 135(2):271–292.

Kreps, D. M. (1981). Arbitrage and equilibrium in economies with infinitely many commodities. *Journal of Mathematical Economics*, 8(1):15–35.

Lakonishok, J., Shleifer, A., and Vishny, R. W. (1994). Contrarian investment, extrapolation and risk. *Journal of Finance*, 49:1541–1578.

Lanza, A. A., Bernardini, E., and Faiella, I. (2023). Machine learning, ESG indicators, and sustainable investment. In *Financial Risk Management and Climate Change Risk: The Experience in a Central Bank*, pages 223–250. Springer.

Ledoit, O. and Wolf, M. (2004a). Honey, I shrunk the sample covariance matrix. *Journal of Portfolio Management*, 30(4):110–119.

Ledoit, O. and Wolf, M. (2004b). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of multivariate analysis*, 88(2):365–411.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. (2018). Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31.

Lyandres, E., Sun, L., and Zhang, L. (2008). The new issues puzzle: Testing the investment-based explanation. *The review of financial studies*, 21(6):2825–2855.

Maeda, I., deGraw, D., Kitano, M., Matsushima, H., Sakaji, H., Izumi, K., and Kato, A. (2020). Deep reinforcement learning in agent based financial market simulation. *Journal of Risk and Financial Management*, 13(4):71.

Maiti, M. (2021). Is ESG the succeeding risk factor? *Journal of Sustainable Finance & Investment*, 11(3):199–213.

Markowitz, H. M. (1952). Portfolio selection. *The Journal of Finance*, 7(1):77–91.

Markowitz, H. M. and Todd, G. P. (2000). *Mean-variance analysis in portfolio choice and capital markets*, volume 66. John Wiley & Sons.

Michaud, R. O. (1989). The Markowitz optimization enigma: Is ‘optimized’ optimal? *Financial analysts journal*, 45(1):31–42.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR.

Moskowitz, T. J. and Grinblatt, M. (1999). Do industries explain momentum? *The Journal of Finance*, 54(4):1249–1290.

Naffa, H. and Fain, M. (2022). A factor approach to the performance of ESG leaders and laggards. *Finance Research Letters*, 44:102073.

Naranjo, A., Nimalendran, M., and Ryngaert, M. (1998). Stock returns, dividend yields, and taxes. *The Journal of Finance*, 53(6):2029–2057.

Nicholas, N. (2008). The black swan: the impact of the highly improbable. *Journal of the Management Training Institut*, 36(3):56.

Novy-Marx, R. (2013). The other side of value: The gross profitability premium. *Journal of financial economics*, 108(1):1–28.

Obite, C., Olewuezi, N., Ugwuanyim, G., and Bartholomew, D. (2020). Multicollinearity effect in regression analysis: A feed forward artificial neural network approach. *Asian J. Probab. Stat*, 6(1):22–33.

Pástor, L., Stambaugh, R. F., and Taylor, L. A. (2022). Dissecting green returns. *Journal of Financial Economics*, 146(2):403–424.

Phoon, K. and Koh, F. (2017). Robo-advisors and wealth management. *The Journal of Alternative Investments*, 20(3):79–94.

Piotroski, J. D. (2000). Value investing: The use of historical financial statement information to separate winners from losers. *Journal of Accounting Research*, pages 1–41.

Pontiff, J. and Woodgate, A. (2008). Share issuance and cross-sectional returns. *Journal of Finance*, 63:921–945.

Reinhart, C. M. (2009). *This time is different: Eight centuries of financial folly*. Princeton University Press.

Rietz, T. A. (1988). The equity risk premium: a solution. *Journal of monetary Economics*, 22(1):117–131.

Ross, S. A. (1976). The arbitrage theory of capital asset pricing. *Journal of Economic Theory*, 13(3):341–360.

Sautner, Z., Van Lent, L., Vilkov, G., and Zhang, R. (2023a). Firm-level climate change exposure. *The Journal of Finance*, 78(3):1449–1498.

Sautner, Z., Van Lent, L., Vilkov, G., and Zhang, R. (2023b). Pricing climate change exposure. *Management Science*.

Schnaubelt, M. (2022). Deep reinforcement learning for the optimal placement of cryptocurrency limit orders. *European Journal of Operational Research*, 296(3):993–1006.

Singh, S. P. and Sutton, R. S. (1996). Reinforcement learning with replacing eligibility traces. *Machine learning*, 22(1):123–158.

Sloan, R. (1996). Do stock prices fully reflect information in accruals and cash flows about future earnings? *Accounting Review*, 71:289–315.

Soliman, M. T. (2008). The use of dupont analysis by market participants. *The Accounting Review*, 83(3):823–853.

Spiess, D. K. and Affleck-Graves, J. (1999). The long-run performance of stock returns following debt offerings. *Journal of Financial Economics*, 54(1):45–73.

Sutton, R. S. and Barto, A. G. (1998). Introduction to reinforcement learning.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.

Tao, R., Su, C.-W., Xiao, Y., Dai, K., and Khalid, F. (2021). Robo advisors, algorithmic trading and investment management: Wonders of fourth industrial revolution in financial markets. *Technological Forecasting and Social Change*, 163:120421.

Tertilt, M. and Scholz, P. (2018). To advise, or not to advise—how robo-advisors evaluate the risk preferences of private investors. *The Journal of Wealth Management*, 21(2):70–84.

Tsai, C.-F., Hu, Y.-H., and Lu, Y.-H. (2015). Customer segmentation issues and strategies for an automobile dealership with two clustering techniques. *Expert Systems*, 32(1):65–76.

Van Staden, P. M., Dang, D.-M., and Forsyth, P. A. (2021). The surprising robustness of dynamic mean-variance portfolio optimization to model misspecification errors. *European Journal of Operational Research*, 289(2):774–792.

Wang, Y.-H., Li, T.-H. S., and Lin, C.-J. (2013). Backward Q-learning: The combination of sarsa algorithm and Q-learning. *Engineering Applications of Artificial Intelligence*, 26(9):2184–2193.

Watkins, C. J. C. H. (1989). Learning from delayed rewards. *Machine Learning*, 8(3):225–229.

Welch, I. and Goyal, A. (2008). A comprehensive look at the empirical performance of equity premium prediction. *The Review of Financial Studies*, 21(4):1455–1508.

White, H. (1989). Learning in artificial neural networks: A statistical perspective. *Neural computation*, 1(4):425–464.

White, H. (2000). A reality check for data snooping. *Econometrica*, 68(5):1097–1126.

Xing, Y. (2008). Interpreting the value effect through the q-theory: An empirical investigation. *The Review of Financial Studies*, 21(4):1767–1795.

Yu, S., Wang, H., and Dong, C. (2023). Learning risk preferences from investment portfolios using inverse optimization. *Research in International Business and Finance*, page 101879.