

STOCHASTIC CONTROL AND DEEP LEARNING
APPROACHES TO HIGH-DIMENSIONAL
STATISTICAL ARBITRAGE

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF MATHEMATICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Jorge Guijarro Ordonez
July 2021

© 2021 by Jorge Guijarro Ordonez. All Rights Reserved.
Re-distributed by Stanford University under license with the author.



This work is licensed under a Creative Commons Attribution-
Noncommercial-No Derivative Works 3.0 United States License.
<http://creativecommons.org/licenses/by-nc-nd/3.0/us/>

This dissertation is online at: <http://purl.stanford.edu/wf857tj3102>

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

George Papanicolaou, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Markus Pelger, Co-Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Kay Giesecke

Approved for the Stanford University Committee on Graduate Studies.

Stacey F. Bent, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.

Abstract

The central problem of this dissertation is the mathematical study of statistical arbitrage in the case of a high-dimensional number of assets, which is analyzed from two complementary approaches.

In the first part of the dissertation, we consider the problem from a stochastic control perspective that extends and combines the Avellaneda and Lee model for statistical arbitrage [2] with the classical Merton framework for portfolio theory [61, 62]. In our framework, given a high-dimensional number of assets and a mean-reverting stochastic model for the dynamics of their residuals through a statistical factor model, an investor must decide how to trade the original assets to maximize the expected utility of her terminal wealth in a finite time horizon, while taking into account market frictions and common statistical arbitrage constraints like dollar neutrality. We study continuous-time and discrete-time versions of the trading problem with both exponential utility and a mean-variance objective, and we prove the existence of interpretable analytic or semi-analytic optimal trading strategies through the study of the corresponding Hamilton-Jacobi-Bellman partial differential equations. We supplement this theoretical study with extensive Monte Carlo simulations that provide further insight about the qualitative behavior of the found optimal strategies under different parameter regimes.

In the second part of the dissertation, we complement the previous study with a general deep-learning framework that mitigates two limitations of the stochastic control approach: strong modeling assumptions on the residual dynamics, and solving the high-dimensional Hamilton-Jacobi-Bellman equations for more realistic objective functions, models, and constraints. To this end, we frame the residual modeling and trading problems as a double optimal control problem, that we solve numerically by restricting the controls to a series of functional classes that range from classical parametric models to the most advanced neural network architectures adapted to our problem. We test these methods by conducting an extensive out-of-sample empirical study with high-capitalization U.S. equity data over the main families of factor models, which provides a comprehensive analysis of the importance of the different elements of a statistical arbitrage strategy and the gains from machine learning methods.

Acknowledgments

The last five years have been a deeply transformative experience in my life, and they would not have been the same without all the people who have helped me along the way.

First of all, I would like to express my most profound gratitude to my advisor, Professor George Papanicolaou. Meeting George was one of the reasons I decided to come to Stanford, and he has always supported and advised me during the last five years, academically and personally. He has always encouraged me to go on, and provided good and detailed criticism of my work. George first introduced me to mathematical finance, guided my first steps into research, and has been an extraordinary mentor during the last five years. I have been very fortunate to have been advised by him.

I would also like to thank my co-adviser, Professor Markus Pelger. Markus introduced me to financial machine learning, to empirical work, and to the research being done at the Advanced Financial Technologies Laboratory at Stanford, and he has been a constant source of big ideas, great feedback and exposure to the latest financial research. Markus also taught me to examine the problems from a more economic perspective and to value interpretability and financial intuition, and he has always been extremely supportive both academically and personally. I have also been very fortunate to have been advised by Markus, and I have learned a lot from him.

I am very thankful to Professor Kay Giesecke for accepting to be part of my reading committee and for his wonderful class on Credit Risk Modeling. I also want to express my gratitude to Professors Tze Leung Lai and Lexing Ying for being part of my oral defense committee and for all what I learned from their courses. I would also like to thank the Math Department and the Advanced Financial Technologies Laboratory at Stanford University for being my academic home during the last years, and Gretchen Lantz for her absolutely fantastic help with all the administrative work and questions.

Pursuing a PhD minor in Philosophy, Literature and Art while working on my main PhD has been a wonderful experience I never imagined before coming to Stanford, and it has given me some of the happiest moments in my Stanford journey. I would like to thank George for allowing me to do this, Professors R. Lanier Anderson and Joshua Landy for coordinating the program and guiding my humanities journey, Professor Héctor Hoyos for all his support and kindness and his wonderful seminars on Latin American literature, and Professors Alexander Nemerov, Pamela Lee, and Nancy

Troy for opening to me the door of contemporary art history and criticism. I would also like to thank the faculty members and the graduate students of the departments that have taught me so much over the past five years (Mathematics, Computer Science, Statistics, Management Science and Engineering, the Graduate School of Business, the Design School, Iberian and Latin American Cultures, Philosophy, Art and Art History), and in general Stanford for providing such a fantastic environment. I couldn't have dreamed of a better place for my Ph.D studies.

I have also been very fortunate to supplement my academic journey with two internships that have taught me look at finance from a more practitioner perspective. In this regard, I would like to thank Francisco Blanch, Benjamin Bowler and the equity derivatives research team at Bank of America Merrill Lynch, and Fanesca Young and her team at GIC. I would also like to express my gratitude to Macarena Estévez, Luisa López, Rebeca Abella and the team at Conento for my first job experience in a non-academic setting and for my first exposure to data science before starting my PhD journey.

Coming to Stanford would not have been possible without the support and advice of many of the professors and teachers I had in Europe. I would therefore like to thank Professors Javier Cilleruelo, Antonio Córdoba, Javier Fernández de Bobadilla, José Manuel Gamboa, Julián López-Gómez, and many others at Universidad Complutense de Madrid, ICMAT, Universidad Autonómica de Madrid, Universitat de Barcelona, and Université de Paris VI, and the fantastic team of teachers I had in my final years of high school.

I would also like to express my gratitude to my friends on both sides of the Atlantic for all their support and good moments during these years. In Europe, I would like to thank Elena Castilla, Pedro Chocano, Carlos Corral, Pepe Dargallo, Kévin Le Divellec, Leon Friederichs, Ettore de Giorgio, Luis Gómez, Víctor González, Miren Ibarra, Javier Muñoz, Pedro Playán, Andrés Quintero, Jeyson Sherlock, Nelson Vides, Víctor Zamora, Marian Zapatero, and many others. In America, I would like to thank Daniel Álvarez-Gavela, Andrea Capra, Jack Ching, Miguel Cisneros, Javier Andréss Enevarre, Christian González, Panagiotis Lolas, Ricardo Mota, Andrea Ottolini, Atharva Parulekar, Mark Perlman, Juan Esteban Plaza, Alberto Quintero, Pedram Safaei, Deepak Sinha, Dipankar Maity, Nhi Truong, Katerina Velcheva, Skyler Wang, David Yang, Ye Ye, and many others. I would especially like to thank Greg Zanotti for all the work and discussions during the second part of this dissertation, and for the wonderful collaboration on the project. Lastly, I would like to thank Mario Sierra, Juan Diego Astudillo, José Gómez, and Naga Katta for the times we spent together.

Finally, I would like to thank my family and especially my parents, Vicente Guijarro and Francisca Ordóñez. My parents have always supported me unconditionally and helped me in any way they could during the past five years. They have always given me the best advice, been there to listen to me in the good and the bad times despite living on opposite sides of the world, and without them this thesis would not have been completed. This dissertation is especially dedicated to them.

Contents

Abstract	iv
Acknowledgments	v
1 Introduction	1
1.1 The stochastic control approach	3
1.2 The deep learning approach	4
2 Stochastic control for statistical arbitrage	6
2.1 Introduction	6
2.2 The model	9
2.2.1 Set-up and assumptions	9
2.2.2 Making the model tractable and imposing market-neutrality	11
2.2.3 Formulation of the control problems	13
2.3 The frictionless results	14
2.3.1 The exponential utility case	14
2.3.2 The mean-variance case	17
2.4 Two extensions	18
2.4.1 Incorporating soft constraints on the admissible portfolios	18
2.4.2 Incorporating quadratic transaction costs	19
2.5 A discrete-time version of the problem	23
2.5.1 The fixed-time setting	23
2.5.2 Optimizing over both times and allocations	25
2.6 Monte Carlo simulations	26
2.6.1 Simulations of the exponential-utility strategy	29
2.6.2 Simulations of the mean-variance strategy with dollar neutrality	30
2.6.3 Simulations of the mean-variance strategy with quadratic transaction costs .	31
2.6.4 Simulations for different factor model and market-neutrality regimes	32
2.6.5 Comparison of the simulated strategies	33

2.7	Conclusions and further research	35
2.8	Proofs	36
2.8.1	Proof of Theorem 2.3.1	36
2.8.2	Proof of Theorem 2.3.2	39
2.8.3	Proof of Theorem 2.4.1	40
2.8.4	Proof of Theorem 2.5.1	42
3	Deep learning for statistical arbitrage	44
3.1	Introduction	44
3.2	Model	50
3.2.1	Arbitrage portfolios	50
3.2.2	Arbitrage signal	52
3.2.3	Arbitrage trading	53
3.2.4	Models for Arbitrage Signal and Allocation Functions	54
3.3	Implementation and Empirical Analysis	65
3.3.1	Data	65
3.3.2	Factor model estimation	66
3.3.3	Implementation	67
3.3.4	Main Results	68
3.3.5	Mean-Variance Objective	72
3.3.6	Unconditional Residual Means	74
3.3.7	Importance of Time-Series Signal	76
3.3.8	Dependency between Arbitrage Strategies	77
3.3.9	Stability over Time	77
3.3.10	Estimated Structure	79
3.3.11	Market Frictions and Transaction Costs	87
3.4	Conclusion	90
A	Appendix for Chapter 3	91
A.1	Data	91
A.1.1	List of the Firm-Specific Characteristics	91
A.2	Implementation of Different Models	91
A.2.1	Feedforward Neural Network (FFN)	91
A.2.2	Ornstein-Uhlenbeck Model	91
A.2.3	Convolutional Neural Network with Transformer	93
A.2.4	Network Estimation Details	95
A.3	Additional Empirical Results	96
A.3.1	Robustness to Hyperparameter Selection	96

A.3.2	Interpretation	100
A.3.3	Dependency between Arbitrage Strategies	102
A.3.4	Time-Series Signal	102
A.3.5	Trading Friction Results for PCA Residuals	103
Bibliography		104

List of Tables

3.1	OOS Annualized Performance Based on Sharpe Ratio Objective	69
3.2	Significance of Arbitrage Alphas based on Sharpe Ratio Objective	70
3.3	OOS Annualized Performance Based on Mean-Variance Objective	73
3.4	Significance of Arbitrage Alphas based on Mean-Variance Objective	74
3.5	OOS Annualized Performance of Unconditional Average Residuals	75
3.6	Significance of Arbitrage Alphas Based on Unconditional Average Residuals	76
3.7	OOS Annualized Performance of CNN+Trans for 60 Days Lookback Window	78
3.8	Significance of Arbitrage Alphas for 60 Days Lookback Window	78
3.9	OOS Annualized Performance of CNN+Trans for Constant Model	79
3.10	Significance of Arbitrage Alphas for Constant Model	80
3.11	OOS Performance of CNN+Trans with Trading Frictions	87
A.1	Firm Characteristics by Category	92
A.2	Hyperparameter options for the network in the empirical analysis	97
A.3	Performance of candidate models on the last year of the validation data set	97
A.4	Alternative best performing models on the data from 2002–2016	98
A.5	Performance of the alternative models on our benchmark residual datasets, 2002–2016	98
A.6	Correlations between the Returns of the CNN+Transformer Arbitrage Strategies . .	102
A.7	OOS Annualized Performance Based on Sharpe Ratio Objective	103
A.8	OOS Performance of CNN+Trans with Trading Frictions	103

List of Figures

2.1	Sample paths of the first three coordinates of X in $[0, T]$	27
2.2	Results for the exponential utility	29
2.3	Results for the mean-variance utility when $r = 0.02$ with different dollar-neutrality restrictions	30
2.4	Results for the mean-variance utility when $r = 0.02$ with different quadratic transaction costs $C = \lambda\sigma\sigma'$	31
2.5	Results for the three strategies above with different p 's, when $r = 0.02, \alpha = 0, \gamma = 1, \lambda = 1$	32
3.1	Conceptual Arbitrage Model	55
3.2	Illustrative Example of Allocation Weights and Signals for Different Methods	57
3.3	Examples of Local Filters	61
3.4	Convolutional Network Architecture	62
3.5	Transformer Network Architecture	64
3.6	Cumulative OOS Returns of Different Arbitrage Strategies	71
3.7	Examples of Allocation and Returns of CNN+Transformer Strategy	81
3.8	Local Basic Patterns of Benchmark Model	82
3.9	Example Attention Weights for Sinusoidal Residual Inputs	83
3.10	CNN+Transformer Model Structure for Representative Residual	84
3.11	CNN+Transformer Model Structure for Representative Residual Over Time	85
3.12	Variable Importance for Allocation Weight	86
3.13	Turnover of CNN+Transformer Model with and without Trading Friction Objective	87
3.14	Proportion of Short Allocation Weights of CNN+Transformer Model with and without Trading Friction Objective	88
A.1	Feedforward Network Architecture	93
A.2	Additional Examples of Allocation Weights and Signals	100
A.3	Example Attention Weights for Sinusoidal Residual Inputs	101

Chapter 1

Introduction

Financial markets have changed considerably over the last few decades, due in great measure to the advent of electronic trading systems and to the parallel increase in the amount of available financial data and computing power. This new environment has led in particular to a surge in algorithmic trading, in which computers execute trading strategies in an automated way through the application of sophisticated mathematical models to large amounts of data.

The central problem of this dissertation is the mathematical study of one of the fundamental classes of these algorithmic trading strategies, called *statistical arbitrage*. There is no universally precise definition of what statistical arbitrage is given the wide variety of investment programs that it encompasses, but, roughly speaking, it may be characterized as a family of trading strategies which (1) identify temporary mispricings among similar assets in a trading universe often involving hundreds to thousands of securities, and (2) attempt to profit from these short-term mispricings through statistical trading rules, usually betting on mean reversion. For these reasons, statistical arbitrage is normally classified within the class of systematic, relative value, contrarian trading strategies.

The simplest case and the historical origin of this kind of strategies is known as *pairs trading*. This corresponds to the special case in which only two assets are traded, and its development is often traced to the quantitative research group of Morgan Stanley in the 1980s [67]. The intuitive idea is to identify two “similar” stocks either by economic fundamentals or by statistical analysis, and, when one outperforms the other, to buy the underperforming asset and to sell the outperforming one with the expectation that the situation will eventually mean-revert.

From a mathematical perspective, this corresponds to three key steps. First, given the time series (r_t) and (s_t) of the returns of the two securities, a long-short portfolio that identifies their relative mispricing is constructed, which corresponds to a model of the form

$$r_t - \beta_t s_t = \epsilon_t \tag{1.1}$$

where the coefficient β_t is estimated from historical data so that ϵ_t is a stationary stochastic process that mean-reverts near some statistical equilibrium. Second, a trading signal is extracted given the observations of ϵ_t and some mathematical model for their dynamics. Third, a trading policy is designed given the investment signals found in the previous step in order to maximize some expected future payoff.

Since its initial development in the late 1980s, pairs trading has been the subject of a considerable amount of academic research in mathematical and quantitative finance, of which [47], [53] and [13] give broad surveys. In contrast, the high-dimensional case, in which not just two but a large number of securities are traded, has remained relatively unexplored and only a few studies have been published (see, for example, [14, 82, 56, 2, 83, 47], which will be reviewed in more detail in upcoming sections).

The overall focus of this dissertation may then be described as the study of the three previous steps in this general case in which we observe the returns of a high-dimensional number of assets $(r_{n,t})_{1 \leq n \leq N}$, and in which, inspired by [2], we replace the simple pairs model

$$r_t = \beta_t s_t + \epsilon_t \quad (1.2)$$

of equation (1.1) with a more general asset pricing model of the form

$$r_{n,t} = \beta_{n,t}^\top F_t + \epsilon_{n,t}, \quad (1.3)$$

where now the factors $F_t \in \mathbb{R}^K$ capture systematic risk, the loading $\beta_{n,t} \in \mathbb{R}^K$ measures the exposure of the n th asset to this common risk, and the residual $\epsilon_{n,t} \in \mathbb{R}$ identifies the relative mispricing of the n th asset. As we will show in detail in the next two chapters, for many empirically important classes of such factor models the residuals ϵ_t correspond to constructible portfolios of the original assets, and this provides a general way of obtaining statistical arbitrage portfolios in our high-dimensional setting that generalize the two-dimensional pairs.

Once these residual portfolios are constructed, our work bifurcates depending on the approach that we follow to study the modeling and trading steps of the residuals. In the second chapter of this dissertation, which is based on the publication [36], this is done through stochastic control techniques in an initial exploration of the problem. In the third chapter, which is based on the preprint [37] and which was presented at the 10th Western Conference on Mathematical Finance, we adopt a very general deep learning framework and we examine the gains it provides with respect to more classical approaches.

For this reason, in the next two sections of this introduction we provide background for the corresponding two approaches, review some of the relevant literature in the context of pairs trading and statistical arbitrage, and discuss in more detail the work presented in the next two chapters, which may be read independently and are each self-contained.

1.1 The stochastic control approach

As mentioned previously, in chapter 2 of this dissertation we approach the modeling and trading parts of the problem from the perspective of stochastic control, building on a line of work that dates back to the pioneering work of Robert Merton in classical portfolio theory [61, 62] and which has recently attracted considerable interest in the context of algorithmic trading in the mathematical finance literature (see, for example, [13] and the references therein).

The key idea of this approach is to assume the market and the investor's wealth dynamics follow a certain stochastic process depending on the investor's trading strategy, which corresponds to the system's control. The investor then chooses how to trade in order to maximize the expected value of some functional of the process and, by applying the dynamic programming principle, the corresponding stochastic optimization problem is reduced to the study of an associated Hamilton-Jacobi-Bellman partial differential equation or a variational inequality.

In the pairs trading literature, this line of research started with the pioneering works by [41] and [65], which studied the dynamic optimal allocation problem in a spread whose price follows a univariate Ornstein-Uhlenbeck process under constant relative risk aversion (CRRA) and Epstein-Zin utilities. These studies were then extended by further works, such as [82], which considers the case of trading a pair of two cointegrated and correlated stocks; [60], which includes recurring and non-recurring arbitrage opportunities; or [55], which includes stochastic volatility. Simultaneously, a different stream of the literature has also studied the problem of optimal timing when trading a couple of cointegrated assets. For instance, [54] analyzes the optimal entry and exit points of a pairs trading strategy, [49] considers multiple entry and exit-points during a trading period, and [68] investigates a regime switching model between three regimes: holding no stocks, long one and short the other stock, and vice-versa. In the multidimensional case, however, very limited research has been conducted, and, to the best of our knowledge, only two studies ([59] and [14]¹) address the problem with stochastic control techniques, both extending the dynamic optimal allocation problem in [82] to allow for multiple cointegrated assets.

The work that we present in the second chapter of this dissertation is essentially an initial extension in several directions of this last line of research, to which we make the following contributions: (1) instead of considering a specific cointegration relation in the logarithm of the asset prices (which is not straightforward to estimate in our high-dimensional setting), we propose a new framework by showing how to trade the residuals of a general statistically-constructed factor model; (2) we discuss both exponential utility and a related mean-variance objective, providing interpretable closed-form optimal strategies; (3) we incorporate transaction costs into the model, which are a crucial element in successful statistical arbitrage strategies; (4) we allow for common statistical arbitrage constraints in the investor's portfolio with the example of dollar neutrality; (5) we also study a more realistic discrete-time version of the problem and establish convergence of the solution to the continuous-time

¹to which a third one, [56], was added after completion of the second chapter of this dissertation.

version as a limiting case; and (6) we provide a simulation study that deals explicitly with a high-dimensional number of assets, and which offers a comprehensive analysis of the qualitative behavior of the different strategies under various parameter regimes.

A central motivation of our approach throughout this chapter is the derivation of solutions as analytical as possible to the corresponding high-dimensional Hamilton-Jacobi-Bellman partial differential equations, with the simultaneous exploration of how far it is possible to go in this direction with control techniques while incorporating important aspects like market frictions, allocation constraints, and discrete-time trading. This has two important advantages. First, it avoids numerical methods that (classically) have been prohibitive for the number of variables that are considered in empirical applications, where the number of assets often ranges from hundreds to thousands. Second, it allows for economic interpretation of the found optimal policies and its dependence on the model parameters, which provides insights about the structure of the problem and its solutions.

On the other hand, major limitations of this approach are, like in most of the control literature for algorithmic trading, (1) the imposition of very strong assumptions on the modeling of the time-series structure of the traded assets, which in our study corresponds to the assumption that the residuals follow a general matrix Ornstein-Uhlenbeck process, and (2) the impossibility of solving with classical methods the associated high-dimensional Hamilton-Jacobi-Bellman equations with more realistic models, objectives, or constraints. These two limitations motivate the data-based approach that we employ in Chapter 3 of this dissertation, and which we discuss next.

1.2 The deep learning approach

In Chapter 3 of the dissertation, we approach the modeling and trading problem using deep learning techniques, following a general trend in quantitative finance of the last few years. The central advantage of these methods compared to the classical techniques lies in their ability to model the interactions between a big number of variables through very flexible functional forms that are learned from large amounts of data, which places less rigid assumptions on the distribution of the data or of the function to be estimated. This kind of techniques has recently achieved remarkable empirical results in a number of important financial problems, inspired by their success in other disciplines like natural language processing, computer vision, and reinforcement learning. See, for example, [34] for applications of machine learning methods to empirical asset pricing, [76] for the study of universal features of price formation in financial markets, [78] for modeling the joint distribution of best ask and best bid prices in limit order books, [77] for the estimation of multi-period mortgage risk, and [48] and [40] for optimal execution in high-frequency algorithmic trading.

Within the pairs trading and statistical arbitrage literature, a still very limited stream of the literature has started applying machine and deep learning methods to the problem. One part of the current studies has focused on relative return forecasting to generate trading signals, generally

followed by equally-weighted trading rules based on signal rankings or thresholds. For example, [47] train feedforward neural networks, random forests, and gradient boosted trees on S&P 500 return data to predict the probability that each stock outperforms the index, and execute a long-short equally weighted strategy on the top and bottom performers. Similar studies include [27], [16], [39], and [22]. A second stream of the literature has instead approached the problem from a value-based reinforcement learning perspective, using deep Q-learning as in [74] and [43] to learn the optimal value function for the allocation control problem in a data-driven way. However, this approach requires strong Markovian assumptions, the discretization of the action space, and a temporally additive structure for the objective function. This complicates modeling the temporal dynamics of financial assets and does not allow for moderately complex trading rules or common financial objectives. Finally, a last approach is proposed in [67], which uses feedforward neural networks to partially parametrize the trading policy in the presence of proportional transactions costs, using a policy found with stochastic control as starting point for the training process of the networks.

The work that we present in the third chapter of this dissertation takes a different route with respect to these methods, and approaches the problem from a much more general perspective and without prior knowledge about the structure of the solution. Recognizing the two distinct conceptual steps of finding signals and of trading based on them, we formulate the problem as a general double optimal control problem, in which, given the residuals, the investor finds the optimal functions to obtain signals and allocations in order to maximize the risk-adjusted return of the strategy under constraints and market frictions. The resulting optimization problem is then solved numerically by restricting the controls to a series of functional classes, that range from classical parametric models to the most advanced neural network architectures adapted to our problem.

In an extensive out-of-sample empirical study, we test these methods with 19 years of daily high-capitalization U.S. equity data over the main families of factor models. This provides a comprehensive analysis of the importance of the different elements of a statistical arbitrage strategy, and of where machine learning methods provide the largest gain with respect to conventional techniques. Our empirical results show that the key component of a good strategy is a flexible model to capture time-series signals and that, given an appropriate model to extract the signals, different choices of factor models or classes of allocation functions give roughly the same performance. We also examine in depth important aspects like robustness of the strategies to different parameter regimes, time stability of the found signals, interpretation of the neural networks output, and the impact of market frictions and transaction costs.

Chapter 2

Stochastic control for statistical arbitrage

2.1 Introduction

¹Modeling of pairs trading based on stochastic control has been an active research topic in mathematical finance for the last few years. After the papers by [41] and [65], an increasing number of models have been proposed in this framework (see, for example, [17], [82], [60]), in which generally they assume that some statistically-designed relation between the prices of two assets is a mean-reverting stochastic process and find a dynamic optimal allocation in continuous time in some version of the classical Merton framework. More recently, a number of papers have also studied the optimal entry and exit points when trading a couple of cointegrated assets, such as [54], [49], [68], and [44].

In the high-dimensional case, however, relatively little model-based research has been conducted. [14] and [59] investigate a multidimensional generalization of the model in [82] and apply stochastic control to solve a Merton-like problem in continuous time on a collection of cointegrated assets, with exponential utility and finite horizon. In a different direction which is not exactly statistical arbitrage, [12] addresses an optimal execution problem on a basket of multiple cointegrated assets, which they also solve with control techniques. Finally, without using stochastic control, the paper by [2] carries out a data-based study of statistical arbitrage in the US equity market by proposing a factor model with mean-reverting residuals and a threshold-based strategy. This model is further analyzed and extended by [83], who discuss risk control and develop an optimization method to allocate the investments given the trading signals.

The previous papers in this high-dimensional framework thus either apply stochastic control to

¹This chapter is based on the publication [36].

a given mean-reverting process or use a factor model to construct this process and then choose the trading signals based on threshold rules, but none of them considers the combination of these two techniques nor do they consider important aspects like transaction costs, statistical arbitrage portfolio constraints like dollar neutrality, or discrete-time trading in the control case. The present chapter aims to fill this gap by providing a study of statistical arbitrage in a high-dimensional setting that combines factor models and the tools from stochastic control, considers transaction costs and statistical arbitrage constraints, and obtains closed-form optimal strategies in continuous and discrete time which are interpretable and do not require prohibitive numerical methods.

More precisely, in our framework an investor observes the returns of a high-dimensional collection of risky assets and, similar to [2] and [83], uses historical data to statistically construct a factor model such that the cumulative residuals are assumed to be mean-reverting and following an Ornstein-Uhlenbeck process. However, unlike these previous studies, these residuals may be correlated and interdependent and, based on their behavior, the investor must decide how to optimally allocate her wealth in the risky assets and a riskless security so that the expected utility of her terminal wealth is maximized and she is market-neutral². There are four main results in this chapter:

First, for a big class of statistically-constructed factor models that includes PCA we show how the investor may theoretically construct market-neutral portfolios without solving any optimization problem (unlike the approach followed in [83] or [8], for example) provided that the factor model holds, and we show how this makes the optimal allocation problem analytically tractable and guarantees market-neutrality by construction. These portfolios are explicitly computable and depend quadratically on the factor model loadings and, to the best of our knowledge, using this construction to connect factor models and stochastic control theory is new.

Second, using these explicit market-neutral portfolios as control variables, we show how the investor should trade optimally in continuous time to maximize either an exponential utility or a mean-variance objective, obtaining explicit analytic forms of the optimal strategies in both cases in this high-dimensional setting. The structure of these optimal strategies is related to the classical solution of the Merton problem and is affine in the deviation of the residuals from their statistical mean, thus giving a precise estimate of how much we should buy when the assets are underpriced and how much we should sell when they are overpriced, as in classical pairs trading. The coefficients are given by the solution of matrix Riccati differential equations and depend quadratically on the factor model loadings, and the strategies in both the exponential and the mean-variance case are surprisingly similar except for a non-myopic correction term that does not appear in the classical framework under a geometric Brownian motion. This arises from the fact that in our case the drift of the underlying Ornstein-Uhlenbeck process is stochastic.

The structure and the techniques to find these affine strategies are thus similar in spirit to those in the affine process literature in finance (see [21] for a broad survey), to the more recent affine

²In this chapter we use the expression “market-neutral” as in [2] to refer to factor neutrality.

control literature in algorithmic trading (see, for example, [13] and the references therein), and to the literature on extensions of the Merton problem (see, for example, [4], [57], [28] and [64], which deal with a single risky asset in the context of the Schwarz model or in geometric Brownian motion with stochastic drift or volatility; and [9] and [7], which consider the multiasset case in the setting of geometric Brownian motion with uncertain drift). While the techniques that we use to find the optimal strategies are therefore classical, the framework and the results are new because the mean-reverting behavior of the underlying stochastic process arises from the residuals of a factor model and in the context of statistical arbitrage, and we consider the general case of an arbitrary number of assets with a market-neutrality restriction and a general matrix Ornstein-Uhlenbeck process. Moreover, the explicit solutions allow us to understand the dependence of the optimal strategies on specific elements of a statistical arbitrage strategy (such as the factor model, its loadings matrix and its connection with market-neutrality, and the mean-reversion speed of the residuals and their correlation structure), and to compare arbitrageurs with exponential and mean-variance utilities.

Third, we extend the previous results in two directions by discussing how to incorporate into the model soft constraints frequently imposed by arbitrageurs such as dollar-neutrality, and also market frictions in the form of quadratic transaction costs, inspired by [30] and [31] and also by the more general quadratic transaction cost and linear price impact literature in portfolio theory (see, for example, [63] and [66] for some new research directions, and [69], [75], [1] and [5] for some classical papers). In both extensions, we again find explicit analytic strategies which are easily interpretable, and which quantitatively correspond to quadratic corrections in the structure of the original optimal strategies (when adding soft constraints like dollar neutrality) or to “tracking” averages of the future original optimal portfolios (when adding quadratic transactions costs). Moreover, in both cases these new strategies depend quadratically on the loadings of the factor model. Again, the novelty of the results comes from the study of these questions (dollar neutrality, transaction costs, etc.) in a new context in which they are crucial (statistical arbitrage with an arbitrary number of assets following a general matrix Ornstein-Uhlenbeck process and a market-neutrality restriction, in particular using control techniques and a factor model), and this framework and the strategies that we find are new to the best of our knowledge.

Fourth, we finally consider a more realistic discrete-time version of the problem, in which instead of trading in continuous time the investor solves a double optimal control problem by finding the optimal times to rebalance the portfolio and also the new allocations at the given times. Within this new framework, our main results are the analytic description of the optimal strategies in the mean-variance case for fixed times, a proof of its almost-sure convergence to the continuous-time strategies of the previous sections, and a study of the optimal strategies when optimizing over both times and allocations.

To conclude the chapter with a more empirical analysis, we also perform extensive numerical

simulations with a high-dimensional number of assets. This gives further insights about the behavior of the previous strategies that are not obvious when looking at the corresponding equations, and allows us to understand the sensitivity of the model parameters and the dependence on the underlying factor model. This high-dimensional numerical study is also new with respect to the existing literature, and the main conclusions are that (1) the exponential-utility strategies are more profitable than the mean-variance strategies and they also take more extreme positions, (2) after some initial up and downs the sample paths of the different wealth processes progressively stabilize due to the asymptotic properties of the Ornstein-Uhlenbeck process, (3) increasing the risk-control parameters consistently produces a concentration of the distribution of the terminal wealth around smaller values, and (4) imposing market neutrality when the loadings of the factor model get bigger leads to more aggressive strategies whose terminal wealth has a higher variance.

The remainder of the chapter is organized as follows. In section 2.2 we introduce our model, construct the market-neutral portfolios that make the problem analytically tractable, and formulate the control problems. Next, in section 2.3 we present the basic results under the exponential and the mean-variance frameworks, whereas section 2.4 extends these results by considering the addition of soft constraints and of quadratic transaction costs. Section 2.5 considers the discrete-time version of the problem. Section 2.6 contains Monte Carlo simulations that provide further insight about the qualitative behavior of the found strategies, and section 2.7 presents the main conclusions and proposes future new directions of research. Finally, the proofs are contained in section 2.8.

2.2 The model

2.2.1 Set-up and assumptions

In the remainder of this chapter we will consider the following general framework. We will assume that an investor observes the returns of a large number N of risky assets and, like in classical portfolio theory based on stochastic control, she must decide how to dynamically allocate her wealth by investing in them or in a riskless asset with constant interest rate r so that the expected utility of her wealth at a finite terminal time T is maximized. However, unlike the classical framework and the existing literature, to do so she will execute a statistical arbitrage strategy based on a factor model, in which instead of trading depending on the state of the original returns she will trade depending on the behavior of the residuals, which will be the trading signals. For example, in the case of two assets, this is equivalent to classical pairs trading, in which the investor may perform a simple linear regression on the returns of two historically correlated securities and, depending on how far the oscillation of the residual is from its historical average, she decides if there is a mispricing and opens and closes long and short positions in the original assets in a market-neutral way. In this chapter, we will study the generalization of this to the high-dimensional case of an arbitrary number of assets, in which we substitute the simple linear regression by a statistical factor model and we

study the optimal allocations under the framework of stochastic control, assuming a mean-reverting stochastic model for the behavior of the residuals.

More precisely, we make the following three general assumptions on how the investor will generate these residuals and what dynamics they will have:

1. **Assumption 1:** The investor has computed a factor model for the returns of the risky assets, which will hold during the investment finite horizon and is given in differential form by

$$dR_t = \Lambda dF_t + dX_t, \quad (2.1)$$

where R_t is the cumulative asset return process, Λ is the (constant-in-time³) loadings matrix, F_t is the cumulative factor returns process, and X_t is the cumulative residual returns process⁴.

2. **Assumption 2:** This factor model has been computed statistically by using some version of PCA⁵, so the rows of Λ are the largest eigenvectors of some square matrix and the discrete-time version of dF_t (i.e., the daily, hourly, etc. factors returns) is computed by linearly regressing the discrete-time version of dR_t (i.e., the daily, hourly, etc. assets returns) on some rescaling of Λ , so

$$dF_t = \tilde{\Lambda} dR_t \quad (2.2)$$

for some matrix $\tilde{\Lambda}$. However, the only fact we will need about this assumption is that (2.2) holds for some matrix $\tilde{\Lambda}$, which allows for a bigger class of factor models than classical PCA.

3. **Assumption 3:** The process X_t given by the cumulative residuals is mean-reverting. In particular, for analytic tractability we assume that it is a matrix N -dimensional Ornstein-Uhlenbeck process satisfying the following stochastic differential equation with known parameters

$$dX_t = A(\mu - X_t)dt + \sigma dB_t,$$

where A is a constant N -dimensional square matrix, μ is a constant N -dimensional vector, B_t is a vector of m independent Brownian motions in the usual complete filtered probability space $(\Omega, \mathcal{F}, \mathbb{P}, (\mathcal{F}_t)_{0 \leq t \leq T})$, and σ is a constant $N \times m$ matrix such that the instantaneous covariance matrix $\sigma\sigma'$ is invertible.

³We make this assumption for analytic tractability in our control framework, and given that the trading frequency is in general higher than the frequency at which these loadings will change significantly.

⁴Here we have written the factor model in a somewhat unusual differential form in terms of the cumulative residuals and returns because of notational simplicity for this section of the chapter. In practice, however, the factor model will be estimated in discrete time, by replacing the differentials by the corresponding discrete increments (so, for instance, dR_t should be replaced by the daily, hourly, etc. asset returns, dF_t would be just the corresponding daily, hourly, etc. factors returns, and so forth). In any case we will only use this notation and framework in this section of the chapter, and the reader may look at [2] for essentially the same continuous/discrete time framework and some estimation techniques.

⁵See [50] for some new versions of PCA that might be particularly interesting for this problem.

The previous framework thus combines high-dimensional statistical arbitrage, factor models and stochastic control in a way which is new to the best of our knowledge, and it extends several models in the existing literature. For example, statistical arbitrage models based on a more particular case of Assumptions 1, 2, 3 (in which the residuals are assumed to be independent one-dimensional Ornstein-Uhlenbeck processes, so A and σ are diagonal) and in which no stochastic control methods are applied have been studied empirically in the US equity market by [2] and [83]. In a different direction, if we consider the particular case of removing the factor model by making $\Lambda = 0$, we have the situation in which the returns are globally mean-reverting following a matrix Ornstein-Uhlenbeck process, which has also been studied empirically and analytically using stochastic control techniques in the context of optimal execution in [12].

2.2.2 Making the model tractable and imposing market-neutrality

Unlike the classical literature on portfolio choice based on stochastic control, choosing as control variables the amount of capital that the agent invests in each of the N risky assets of the previous framework might make the optimal allocation problem analytically intractable. Indeed, since we only have information about the dynamics of the residuals and not directly about the returns like in the classical framework, these residuals are not independent, and the factors depend on the returns, the classical approach would lead to complicated interdependencies. Moreover, since the investor is executing a statistical arbitrage strategy, we would need to incorporate additional market neutrality constraints⁶ so that the returns of the strategy do not depend on the model factors, but just on the idiosyncratic component of the model given by the residuals. This would complicate the problem further, and would require numerical optimization methods as done in [83].

In this chapter, on the contrary, we approach both problems simultaneously and we solve them analytically by following a new approach. This is based on the following proposition, which shows that, by using the original N risky assets, it is possible to construct analytically N market-neutral portfolios whose returns only depend on one coordinate of X , which simplifies the complexity of the problem and makes it analytically tractable:

Proposition 2.2.1. *Under the previous assumptions, it is possible to construct explicitly N market-neutral portfolios such that investing any real number π_{it} of dollars in the i -th one at time t yields an instantaneous return of $\pi_{it} dX_{ti}$ (and hence a combined return of $\pi_t \cdot dX_t$).*

Moreover, the total amount of capital invested at time t by doing so is $\pi_t \cdot p$ for an explicit constant-in-time vector $p \in \mathbb{R}^N$, which depends quadratically on the factor model loadings.

Proof. The mathematical construction of the market-neutral portfolios under the given assumptions

⁶Here we use market-neutrality in the sense of [2].

is surprisingly straightforward and involves just a linear projection. Indeed, (2.1) implies that

$$dR_{ti} = \sum_j \Lambda_{ij} dF_{tj} + dX_{ti},$$

whereas (2.2) yields

$$dF_{tj} = \sum_k \tilde{\Lambda}_{jk} dR_{tk}.$$

Combining the two previous equations we find that, for $c_{ik} := \sum_j \tilde{\Lambda}_{jk} \Lambda_{ij}$,

$$dR_{ti} = \sum_k \left(\sum_j \tilde{\Lambda}_{jk} \Lambda_{ij} \right) dR_{tk} + dX_{ti} = \sum_k c_{ik} dR_{tk} + dX_{ti}.$$

Thus, if at time t we hold the (explicitly constructible) constant-in-time portfolio given by

$$\tilde{p}_i := (-c_{i1}, -c_{i2}, \dots, -c_{i,i-1}, 1 - c_{ii}, -c_{i,i+1}, \dots, -c_{iN})$$

(i.e., we invest $-c_{i1}$ dollars in the first asset, $-c_{i2}$ dollars in the second one, and so on), we automatically obtain an instantaneous return of dX_{ti} , which is market neutral and depends only on the i th coordinate of the process X_t . Further, from the above equations it is also obvious that for any real number π_{it} , $\pi_{it}\tilde{p}_i$ will also be market-neutral and yielding a return of $\pi_{it}dX_{ti}$, and the same applies to $\sum_i \pi_{it}\tilde{p}_i$, which will have a return of $\sum_i \pi_{it}dX_{ti} = \pi_t \cdot dX_t$.

Finally, regarding the last part of the statement just observe that the total amount of capital invested in the strategy $\pi_t = (\pi_{it})_{1 \leq i \leq N}$ at time t is simply

$$\sum_i (\pi_{it}\tilde{p}_i) \cdot \mathbb{1} = \sum_i \pi_{it}(\tilde{p}_i \cdot \mathbb{1}) = \pi_t \cdot p$$

where $p := (\tilde{p}_i \cdot \mathbb{1})_{1 \leq i \leq N}$, which concludes our proof. \square

Remark 2.2.1. Note in particular that, if Λ or $\tilde{\Lambda}$ are sparse matrices, then most of the c_{ik} in the above construction will be 0, so the investor will be investing in a few number of assets in each market-neutral portfolio and this could significantly reduce his transaction costs while rebalancing his positions. In particular, [71] discusses a way of obtaining this kind of sparse factor model.

The key consequence of the above proposition is that, if we choose as control variables the amount of capital π_t that we wish to invest in these N market-neutral portfolios (instead of directly in the original assets) at time t , the dynamics of the problem get simpler, they only depend separately on the coordinates of X , and we have market-neutrality by construction. This solves simultaneously the two problems we discussed before and allows us to connect stochastic control and the factor model in a simple way, and it is therefore the approach which we will adopt in the remainder of this

chapter.

Note also that, under these new control variables, all the information about the factor model and in particular about its loadings matrix is now encoded in the vector p , which will play an important role in the remaining sections. Moreover, some statements about the strategies must be rewritten in terms of it within this new framework. For instance, in the new setting a strategy $(\pi_t)_{0 \leq t \leq T}$ is dollar-neutral at t if $p \cdot \pi_t = 0$, since as we mentioned before $p \cdot \pi_t$ is the total capital spent at time t .

2.2.3 Formulation of the control problems

Under the previous framework, now we formulate rigorously the control problems we will study in the chapter. We suppose that the investor executes the following trading strategy: at each time $t \in [0, T]$, she invests π_t dollars in the risky market-neutral portfolios we constructed in Proposition 2.2.1, and she invests her remaining wealth in the risk-free asset with constant interest rate r , so that the resulting strategy is self-financing. Thus, assuming for the moment no market frictions or constraints (which will be both considered in section 2.4), the evolution of her wealth is given by the equation

$$dW_t = \pi_t \cdot dX_t + (W_t - \pi_t \cdot p)r dt \quad (2.3)$$

and she aims to choose π_t to maximize the expected utility of her terminal wealth, which is given by $u(W_T)$ for a fixed utility function u .

Supposing further that she trades continuously in time, this means that mathematically she must solve the high-dimensional non-linear stochastic optimization problem given by

$$H(t, x, w) = \sup_{\pi \in \mathcal{A}_{[t, T]}} \mathbb{E}_{t, x, w} [u(W_T)] \quad (2.4)$$

subject to

$$dW_t = (\pi'_t A(\mu - X_t) + (W_t - \pi'_t p)r) dt + \pi'_t \sigma dB_t$$

$$dX_t = A(\mu - X_t)dt + \sigma dB_t,$$

where ' indicates transposition, and the admissible set $\mathcal{A}_{[t, T]}$ is the set of all the \mathcal{F}_s -predictable and adapted processes $(\pi_s)_{s \in [t, T]}$ in \mathbb{R}^N with the minimal technical restrictions that $\mathbb{E}[\int_t^T \|\pi_s\|^2 ds] < \infty$ (so Ito's formula may be applied and doubling strategies are excluded) and the above SDEs have a unique strong solution.

Finally, the associated dynamic programming equation of the problem is non-linear and $(N+2)$ -dimensional, and is given by

$$0 = \partial_t H + (\mu - x)' A' \nabla_x H + \frac{1}{2} \text{Tr}(\sigma \sigma' \nabla_{xx} H) + \sup_{\pi} \left((\pi' A(\mu - x) + (w - \pi' p)r) \partial_w H + \frac{1}{2} \pi' \sigma \sigma' \pi \partial_{ww} H + \pi' \sigma \sigma' \nabla_{xw} H \right) \quad (2.5)$$

with terminal condition $H(T, x, w) = u(w)$.

The problem is therefore formally related to the classical Merton framework, but instead of a geometric Brownian motion there is a multidimensional Ornstein-Uhlenbeck process which makes it impossible to combine the dynamics of W and X into a single equation and to get rid of the N -dimensional state variable x . Moreover, unlike the previous studies on extensions of the Merton problem with an Ornstein-Uhlenbeck process discussed in section 2.1, in (2.4) and (2.5) the mean-reverting behavior of the underlying stochastic process arises in the context of statistical arbitrage and from the residuals of a factor model (which is encoded in the vector p of the equations above and which will play an important role in the following sections), and we consider the general case of an arbitrary number of assets with a market neutral restriction. Furthermore, the model will be extended in section 2.4 to incorporate other important features of statistical arbitrage strategies, like dollar neutrality restrictions and transaction costs, and we will analyze the impact of the factor model on these extensions.

Quite surprisingly, the previous problems admit interpretable closed-form solutions – which is computationally useful in this high-dimensional setting, and which allows us to understand the influence of the model parameters and especially of the factor model – in the cases in which the utility is exponential or of a Markowitz-inspired mean-variance type, but not for other usual choices of utility functions, like the HARA family. We show this in the following two sections, first for the simple setup of (2.4) and (2.5) in section 2.3, and then extending the model in section 2.4 to incorporate soft constraints on the investor’s portfolio and quadratic transaction costs.

2.3 The frictionless results

In this section we present the closed-form, optimal strategies for the problem given by (2.4) and (2.5) in the cases in which the utility is exponential or of a mean-variance type, discussing the former in the first subsection and the latter in the second one.

2.3.1 The exponential utility case

In the first setting, the explicit description of the optimal strategy is given by the following main theorem (see [14] and [59] for related results with an exponential utility):

Theorem 2.3.1. *Under an exponential utility (so $u(w) = -e^{-\gamma w}$ for some $\gamma > 0$) and the technical condition described in our verification theorem (Proposition 2.3.2 below), the optimal portfolio to have at time t according to (2.4) is explicitly computable and given by*

$$\pi_t^* = (\sigma\sigma')^{-1} \frac{A(\mu - X_t) - pr}{\gamma e^{r(T-t)}} + \frac{A'(\sigma\sigma')^{-1}}{\gamma e^{r(T-t)}} \left((A(\mu - X_t) - pr)(T-t) - Apr \frac{(T-t)^2}{2} \right).$$

The result follows from the following two propositions, whose proof is given in section 2.8.1 using classical stochastic control techniques:

Proposition 2.3.1 (Solving the PDE). *The value function H associated to (2.4) and (2.5) when $u(w) = -e^{-\gamma w}$ is explicitly computable and admits the probabilistic representation $H(t, x, w) = -\exp(-\gamma(we^{r(T-t)} + h(t, x)))$ where*

$$h(t, x) = \mathbb{E}_{t,x}^* \left[\int_t^T \frac{1}{2\gamma} (A(\mu - Y_s) - pr)' (\sigma\sigma')^{-1} (A(\mu - Y_s) - pr) ds \right]$$

and $dY_t = rpdt + \sigma dB_t^*$ for a new Brownian motion B^* under a new probability law \mathbb{P}^* . The associated optimal control in feedback form is then

$$\pi^* = -(\sigma\sigma')^{-1} \frac{\mathcal{D}H}{\partial_{ww} H} \quad (2.6)$$

where $\mathcal{D}H = (A(\mu - x) - pr) \partial_w H + \sigma\sigma' \nabla_{xw} H$.

Proposition 2.3.2 (Verification). *The strategy given in Theorem 2.3.1. is indeed optimal if*

$$4 \max_{0 \leq s \leq T} \|\Lambda_0(s)\| < 1 \quad \text{and} \quad 32 \max_{0 \leq s \leq T} \|\Lambda_1(s)\| < 1,$$

where $\Lambda_0(s)$ and $\Lambda_1(s)$ are the diagonal matrices containing, respectively, the eigenvalues of $\Omega^{1/2}(C_0 + C'_0)\Omega^{1/2}(s)$ and $\Omega^{1/2}C_1C'_1\Omega^{1/2}(s)$, for

$$C_0(s) = A'(\sigma\sigma')^{-1}A(I_N + A(T-s)), \quad C_1(s) = A'(\sigma\sigma')^{-1}(I_N + A(T-s))\sigma$$

$$\Omega(s) = \int_0^s e^{-A(s-u)} \sigma\sigma' e^{-A'(s-u)} du.$$

Besides being a closed-form strategy which is easily implementable in our high-dimensional setting, the above optimal portfolio is also interpretable. Indeed, the first term of the optimal policy is Merton-like since it represents the drift of the underlying process (which here is stochastic unlike

in the classical geometric Brownian motion) minus the adjusted risk-free rate (which here depends on the loadings of the factor model via p). This is divided by a measure of the volatility (which is given by $\sigma\sigma'$, the instantaneous quadratic covariation of X) and the Arrow-Pratt coefficient of absolute risk-aversion of the value function with respect to the wealth w (i.e., $-\partial_{ww}H/\partial_wH$), which is the product $\gamma e^{r(T-t)}$, where γ is the risk aversion parameter of the utility function and the factor $e^{r(T-t)}$ measures the gains from interest between t and T .

On the other hand, the second summand is a non-myopic correction term which again depends linearly on the drift of X , and whose effect vanishes when we approach the terminal time T . Moreover, while the first term does not depend explicitly on the terminal time T , this correction term does, reflecting the fact that, since there are non-zero interest rates and moreover the behavior of the residuals is oscillating, the investor must keep in mind the final horizon to decide if she bets on the mean-reversion cycle before that time. Finally, as the risk-aversion parameter γ , the instantaneous volatility $\sigma\sigma'$, or the interest rate r increase, the optimal portfolio vector π_t^* gets closer to 0, implying that the investor will simply invest most of her wealth in the riskless asset.

The above strategy is also intuitive within our framework of statistical arbitrage with a factor model, and sheds further light on the problem. Indeed, note that the current state of the residual process X_t only appears in the strategy through the terms in $A(\mu - X_t)$, which essentially tells us to invest more in the risky assets the further their residuals are from their historical mean μ and in a way proportional to the historical mean reversion speed given by A , like in classical pairs trading. Moreover, all the remaining terms depend jointly on the factor model and the interest rate through the term pr , which reflects the cost of the leverage associated to imposing market-neutrality through the loadings of the factor model. In particular, note that, the bigger the loadings of the factor model are and hence the bigger p is, the more the agent will need to invest to achieve market neutrality (again like in pairs trading with a big beta) and the bigger her leverage will be, and this will affect the optimal strategy depending on the interest rate r .

Finally, regarding the technical optimality conditions, intuitively they arise from the fact that $H(t, X_t, W_t^*)$, the value function evaluated at the wealth process W_t^* corresponding to the optimal strategy, may blow up because of the exponential function coming from the exponential utility. In particular, since W_t^* ends up being an Ito process depending quadratically on X_t and X_t is Gaussian, the term $\exp(-\gamma W_t^* e^{r(T-t)})$ is related to the moment generating function of a chi-squared distribution, which blows up far away from 0. Thus, these technical conditions are just ensuring that the corresponding functions are integrable. Interestingly, this does not depend on the risk-aversion parameter γ , the interest rate r , or the factor model used (captured by p), but just on the parameters of X and the terminal time T .

2.3.2 The mean-variance case

In the second, Markowitz-inspired mean-variance framework, the investor tries to maximize her expected terminal wealth, but at the same time she continuously penalizes at each instant the instantaneous variance of her wealth process according to a time-dependent volatility-aversion function $\gamma(t)$. The optimal strategy in this case is again available in closed form and interpretable and, for an appropriate choice of this volatility-aversion function, we obtain the same optimal policy as in the exponential case but without the correction term. This is shown in the following theorem, whose proof is given in section 2.8.2:

Theorem 2.3.2. *If $\gamma(t)$ is continuous and positive on $[0, T]$, the problem in (2.4) with the following mean-variance objective function*

$$H(t, x, w) = \sup_{\pi \in \mathcal{A}_{t,T}} \mathbb{E}_{t,x,w} \left[W_T - \int_t^T \frac{\gamma(s)}{2} \frac{d}{d\tau} \text{Var}_s(W_\tau)|_{\tau=s} ds \right]$$

has explicit optimal portfolio at t given by

$$\pi_t^* = (\gamma(t)\sigma\sigma')^{-1} (A(\mu - X_t) - pr) e^{r(T-t)}.$$

In particular, for $\gamma(t) = \gamma e^{2r(T-t)}$, the above optimal policy is the same as the first term of the corresponding one in Theorem 2.3.1.

Regarding the interpretation of the mean-variance strategy within our context of statistical arbitrage and its connection with the exponential-utility arbitrageur, there are two important remarks.

First, as we mentioned, the new optimal strategy is the same as the myopic part of the exponential case modulo the value of $\gamma(t)$. In particular, this means that, unlike the exponential arbitrageur, the mean-variance arbitrageur will not take into account the expected number of mean-reversion cycles until the terminal time T . Moreover, for a non-zero interest rate and a constant volatility aversion $\gamma(t)$, the mean-variance arbitrageur is more aggressive than the corresponding exponential investor with the same γ , since she will invest significantly more capital (quantitatively, by a factor of $e^{2r(T-t)}$) in going long or short, taking more aggressive positions the higher the interest rate is and the sooner it is with respect to the terminal date.

Second, the optimal strategy has two components like in section 2.3.1: one term in $A(\mu - X_t)$ which measures how far the residuals are from their historical mean and how fast they will mean-revert (like in classical pairs trading), and a second term in pr linked to the factor model, which measures the cost of the leverage associated to imposing market neutrality. In particular, note that, the bigger the loadings of the factor model are (and hence the bigger p is), the more aggressive the positions will be and the more leverage the investor will have if $r \neq 0$.

2.4 Two extensions

In this section of the chapter, we complete the picture described in the previous two sections by considering two important extensions within the context of statistical arbitrage with a factor model. In the first subsection, we show how to incorporate in the above strategies soft constraints frequently imposed by arbitrageurs with the example of dollar-neutrality. In the second one, we introduce market frictions in the form of quadratic transaction costs. In both cases, we obtain again closed-form analytic solutions which are interpretable, convenient from a computational perspective in our high-dimensional setting, and which shed further light on the influence of the factor model and its connection with market neutrality.

2.4.1 Incorporating soft constraints on the admissible portfolios

While imposing restrictions on the portfolios by introducing hard constraints directly on the admissible set $\mathcal{A}_{t,T}$ leads in general to control problems that must be solved numerically (and hence potentially unfeasible in a high-dimensional setting since in applications the number of assets ranges from hundreds to thousands), it is still possible to impose many additional soft constraints in the two frameworks of section 2.3 without significantly increasing the difficulty of the problems, by just adding a carefully chosen penalty term to the corresponding objective function.

As an illustration of this, we give in the next corollary the corresponding optimal strategies when a dollar-neutrality restriction is softly enforced. To do so, recall that, within the framework of section 2.2 that imposed market-neutrality within the factor model, a strategy π_t is dollar neutral if $p \cdot \pi_t = 0$, which means that the total amount of capital invested at time t is 0. Hence, we can softly enforce dollar neutrality by replacing the wealth process of Theorems 2.3.1 and 2.3.2 by the penalized wealth process defined by $d\tilde{W}_t := dW_t - \alpha(t)(\pi_t \cdot p)^2/2dt$ for a certain general time-dependent penalty function $\alpha(t)$. This penalizes non dollar-neutrality (i.e., $\pi_t \cdot p \neq 0$) at each time and is quadratic to simplify the optimization process.

The proof follows the same lines as in the previous two cases and is obtained from them by small modifications, so we will omit it for the sake of brevity.

Corollary 2.4.1. *Suppose that dollar neutrality is softly enforced by replacing the wealth process of Theorems 2.3.1 and 2.3.2 by the penalized wealth process defined by $d\tilde{W}_t := dW_t - \alpha(t)(\pi_t \cdot p)^2/2dt$. Then*

1. *The problem with mean-variance utility has optimal portfolio at t given by*

$$\pi_t^* = (\gamma(t)\sigma\sigma' + \alpha(t)pp')^{-1} (A(\mu - X_t) - pr) e^{r(T-t)}.$$

2. The problem with exponential utility has optimal portfolio at t given by

$$\pi_t^* = (\gamma e^{r(T-t)} \sigma \sigma' + \alpha(t) pp')^{-1} (A(\mu - X_t) - pr - \gamma \sigma \sigma' (b(t) + c(t)X_t)).$$

where $c(t)$ is an $N \times N$ symmetric matrix and $b(t)$ is an N -dimensional vector, vanishing when $t \rightarrow T$, and with coordinates depending on A, σ, rp, γ and $\alpha(t)$. In particular, $c(t)$ is given by the solution of the matrix Riccati ODE

$$0 = \partial_t c - A'c - cA - \gamma c\sigma\sigma'c + e^{r(T-t)}(A + \gamma\sigma\sigma'c)'M(t)(A + \gamma\sigma\sigma'c)$$

and $b(t)$ is the solution of the linear system of ODEs

$$0 = \partial_t b - A'b + cA\mu - e^{r(T-t)}(A + \gamma\sigma\sigma'c)M(t)(A\mu - pr - \gamma\sigma\sigma'b) - \gamma c\sigma\sigma'b,$$

both with terminal conditions $b(T) = c(T) = 0$ and where $M(t) = (\gamma\sigma\sigma'e^{r(T-t)} + \alpha(t)pp')^{-1}$.

The resulting optimal policies have therefore the same structure as the two previous strategies of section 2.3, but now the additional term $\alpha(t)pp'$ has been introduced in the inverse to enforce the dollar-neutrality condition. This again depends on the factor model via p and is related to how extreme the capital positions will be because of the market-neutrality restriction, which depends directly on the loadings matrix and hence on p . Note in particular that, the bigger the loadings of the factor model are, the bigger $\alpha(t)pp'$ will be and hence the bigger the impact of the dollar neutrality restriction will be.

2.4.2 Incorporating quadratic transaction costs

In this subsection, we extend our model to incorporate market frictions in the form of transaction costs, which play a crucial role when executing statistical arbitrage strategies. We consider in particular quadratic transaction costs, which are in general a measure of price impact or illiquidity and which make the model analytically tractable.

To do so, rather than looking at the amount of capital π_t invested in the risky assets at time t as the control variables, we consider the trading intensity I_t at which these investments will be made at time t , which is given by $d\pi_t = I_t dt$. We can now adapt to our setting the model for transaction costs introduced in [31], who posit, providing market microstructural justification and referring to empirical research, that these transaction costs at time t may be represented quadratically as $I_t'CI_t$ for a certain symmetric positive-definite matrix C ⁷, which essentially comes from the assumption that the price impact of the investor's actions is linear on its trading intensity I_t .

⁷The assumption that C is symmetric is without loss of generality, since if the transaction costs are given by $I_t'\tilde{C}I_t$ for a non-symmetric \tilde{C} , then one can see that by considering the symmetric part of \tilde{C} (given by $C := (\tilde{C} + \tilde{C}')/2$) we have that $I_t'\tilde{C}I_t = I_t'C I_t$.

Under this framework, we rewrite the performance criteria of Theorem 2.3.2 by incorporating the adverse effect caused by these transaction costs on the investor's wealth as a running penalty, obtaining the stochastic optimization problem given by

$$H(t, x, w, \pi) = \sup_{I \in \mathcal{A}^*} \mathbb{E}_{t,x,w,\pi} \left[W_T - \int_t^T \frac{\gamma(s)}{2} \frac{d}{d\tau} \text{Var}_s(W_\tau)|_{\tau=s} ds - \frac{1}{2} \int_t^T I'_s C I_s ds \right]. \quad (2.7)$$

As we mentioned, in this new problem the control variable is I ; t, x, w, π are now state variables; and \mathcal{A}^* is the set of all \mathcal{F} -adapted predictable processes I_t such that the corresponding SDEs have a unique strong solution for any initial data and both I_t and the resulting π_t given by $d\pi_t = I_t dt$ are again in $L^2(\Omega \times [0, T])$. Thus, the investor aims to maximize her terminal wealth, but penalizing at each instant both for the risk of her strategy (measured by the instantaneous variance of her wealth process) and for the price impact caused by her actions (reflected in the quadratic transaction costs).

In this new setting, it is again possible to find explicitly the optimal strategy that the investor should follow, which is described in detail in the next theorem:

Theorem 2.4.1. *If $\gamma(t) \geq 0$ and is continuous, the optimal strategy in the above problem "tracks" a moving aim portfolio $\text{Aim}(t, X_t)$ with a tracking speed of $\text{Rate}(t)$, according to the following ODE describing the evolution of the optimal trading intensity $I_t = d\pi_t/dt$*

$$I_t = \text{Aim}(t, X_t) + \text{Rate}(t)\pi_t,$$

where $\text{Rate}(t)$ is a $N \times N$ negative-definite matrix tending to 0 when $t \rightarrow T^8$, and $\text{Aim}(t, X_t)$ admits the probabilistic representation

$$\text{Aim}(t, x) = \int_t^T f(s) \mathbb{E}_{t,x}[\text{Frictionless}(s)] ds$$

where $\text{Frictionless}(s)$ is the optimal portfolio at time s in the frictionless case of section 2.3.2. and $f(s)$ is a certain averaging function given in Proposition 2.4.3 below.

Furthermore, the optimal portfolio is then

$$\pi_s^* = \pi_t + \int_t^s : \exp \left(\int_u^s \text{Rate}(v) dv \right) : \text{Aim}(u, X_u) du,$$

where the notation $: \exp(\int_u^t \cdot ds) :$ represents the time-ordered exponential⁹.

⁸and given by the solution of a matrix Riccati ODE specified in the Porposition 2.4.2 below.

⁹Recall that the time-ordered exponential of a time-dependent matrix $A(s)$ is defined as $: \exp(\int_u^t A(s) ds) := \lim_{||\mathcal{P}|| \downarrow 0} \prod_{i=1}^n \exp(A(t_i) \Delta t_i)$, where $\mathcal{P} := \{u = t_0, t_1, \dots, t_n = t\}$ is a partition of $[u, t]$, $\Delta t_i := t_i - t_{i-1}$, and the product is ordered increasingly in time. If $A(s)$ is a scalar, then obviously $: \exp(\int_u^t A(s) ds) := \exp(\int_u^t A(s) ds)$.

Remark 2.4.1. If in particular the investor has constant volatility aversion (so $\gamma(t) = \gamma$), the matrix Riccati ODE is explicitly solvable and

$$\text{Rate}(t) = C^{-1/2} D \tanh(D(t-T)) C^{1/2}$$

for $D := (\gamma C^{-1/2} \sigma \sigma' C^{-1/2})^{1/2}$. Moreover, if the transaction costs are proportional to the volatility (i.e., $C = \lambda \sigma \sigma'$ for $\lambda > 0$, see [30, 31] for a market microstructural justification) then this rate is indeed a scalar given by $\sqrt{\frac{\gamma}{\lambda}} \tanh(\sqrt{\frac{\gamma}{\lambda}}(t-T))$ and $\exp(\int_u^s \text{Rate}(v) dv) := \cosh(\sqrt{\frac{\gamma}{\lambda}}(s-T)) / \cosh(\sqrt{\frac{\gamma}{\lambda}}(u-T))$.

The result follows from the following sequence of three propositions, which are proved in section 2.8.3:

Proposition 2.4.1 (Conjectured solution). *The solution of the HJB equation associated to the problem is $H(t, x, w, \pi) = e^{r(T-t)}w + \frac{1}{2}\pi'a(t)\pi + \pi'b(t, x) + d(t, x)$ if there exist a $N \times N$ symmetric matrix $a(t)$, a N -dimensional vector $b(t, x)$ and a scalar function $d(t, x)$ satisfying*

1. *The matrix Riccati ODE*

$$\partial_t a - \gamma(t) \sigma \sigma' + a C^{-1} a = 0 \quad (2.8)$$

with terminal condition $a(T) = 0$.

2. *The vector-valued and the scalar linear parabolic PDEs*

$$(\partial_t + \mathcal{L}_X)b + e^{r(T-t)}(A(\mu - x) - rp) + a'C^{-1}b = 0 \quad (2.9)$$

$$(\partial_t + \mathcal{L}_X)d + \frac{1}{2}b'C^{-1}b = 0 \quad (2.10)$$

with terminal conditions $b(T, x) = d(T, x) = 0$ and where \mathcal{L}_X is the infinitesimal generator of X , acting coordinatewise.

The hypothesized optimal trading intensity at (t, x, w, π) is then $I^ = C^{-1}(a(t)\pi + b(t, x))$.*

Proposition 2.4.2 (Existence of solutions). .

1. *If the volatility aversion $\gamma(t) \geq 0$ and is continuous, then the Riccati equation (2.8) has a symmetric, bounded and negative definite solution on all $[0, T]$. In particular, for $\gamma(t) = \gamma$, the solution is*

$$a(t) = C^{1/2} D \tanh(D(t-T)) C^{1/2}$$

for $D := (\gamma C^{-1/2} \sigma \sigma' C^{-1/2})^{1/2}$.

2. Moreover, under this condition the parabolic PDEs (2.9) and (2.10) have a unique solution satisfying a polynomial growth condition in x , and this solution admits the probabilistic representation

$$b(t, x) = \mathbb{E}_{t,x} \left[\int_t^T : \exp \left(\int_t^s a'(u) C^{-1} du \right) : e^{r(T-s)} (A(\mu - X_s) - rp) ds \right] \quad (2.11)$$

$$d(t, x) = \frac{1}{2} \mathbb{E}_{t,x} \left[\int_t^T b(t, X_s)' C^{-1} b(t, X_s) ds \right] \quad (2.12)$$

Furthermore, b has linear growth in x and d has quadratic growth in x , both uniformly in t .

Proposition 2.4.3 (Verification). *Under the assumptions of the previous proposition, the trading intensity given in Theorem 2.4.1 is indeed optimal with the choices*

$$\text{Aim}(t, x) = C^{-1} b(t, x), \quad \text{Rate}(t) = C^{-1} a(t), \quad f(s) = C^{-1} : \exp \left(\int_t^s \text{Rate}(u)' du \right) : \gamma(s) \sigma \sigma'.$$

The interpretation of the above strategy, which is again explicit and hence easily implementable in practice, is intuitive and complementary to the infinite-horizon model of [31]: in the presence of quadratic transactions costs, the investor trades with a certain decreasing rate $\text{Rate}(t)$ towards an aim portfolio $\text{Aim}(t, X_t)$ depending on the time and the mean-reversion state of the signals X_t . This aim portfolio is given by a weighted average of the future optimal strategies in the frictionless case, reflecting the fact that now trading is not free and thus to enter a trade the investor must weight the future outcomes derived from the strategy. Moreover, as shown in the above remark, the trading rate is bounded by 1 because of the properties of tanh, depends on t unlike the infinite-horizon case, and is naturally decreasing in the transaction cost parameter λ (or in general in C) and increasing in the volatility aversion parameter γ .

Finally, regarding the influence of the factor model and the imposition of market neutrality in this new setting, note that $\text{Rate}(t)$ is insensitive to it, since it only depends on the risk aversion parameter γ , the volatility of the residual process $\sigma \sigma'$, and the transaction cost matrix C . Similarly, in $\text{Aim}(t, x)$ it only appears through the term $\mathbb{E}_{t,x}[\text{Frictionless}(s)]$ and hence only when considering the future optimal strategies in the frictionless case, which has been described in section 2.3. Likewise, the residual process X_s only affects the strategy through this same term and hence, as seen in section 2.3 when studying these frictionless cases, only through the distance between this residual and its historical mean, like in classical pairs trading.

2.5 A discrete-time version of the problem

We conclude the theoretical section of this chapter by studying a more realistic, discrete-time version of the problem inspired by the literature initiated in the pairs trading case by [54]. Here, instead of trading continuously in time, the investor may only rebalance her portfolio at an optimally chosen finite sequence of deterministic times. More precisely, she must then choose the increasing sequence of deterministic times $t = t_0 \leq t_1 \leq \dots \leq t_L = T$ at which she wants to change her positions, and the sequence of portfolios π_l that she will have between times t_l and t_{l+1} for $1 \leq l \leq L - 1$.

Mathematically, this means that the investor must solve the doubly controlled stochastic optimization problem

$$H(x, w) = \sup_{\mathbf{t}, \pi \in \mathcal{A}_{\mathbf{t}}} \mathbb{E}_{x, w} [u(W_T)] \quad (2.13)$$

where $\mathbf{t} = (t_1, \dots, t_L)$ is an increasing deterministic sequence of times in $[0, T]$,

$$W_{t_{l+1}} = W_{t_l} + \pi_l \cdot (X_{t_{l+1}} - X_{t_l}) + (W_{t_l} - \pi_l \cdot p)r(t_{l+1} - t_l)$$

$$dX_t = A(\mu - X_t)dt + \sigma dB_t,$$

and $\mathcal{A}_{\mathbf{t}}$ is now the set of all the L -tuples $(\pi_l)_{1 \leq l < L}$ of vectors of \mathbb{R}^N such that π_l is \mathcal{F}_{t_l} -adapted.

In this section, we first solve the simplified problem in which the discrete sequence of times at which the investor may rebalance her portfolio has been fixed beforehand, so she does not have to optimize over these times. This is done first in the case in which there are only soft constraints, and then in the general case with hard constraints and arbitrary admissible sets. Finally, in a second subsection we discuss the general problem in which the investor also optimizes over the deterministic times.

2.5.1 The fixed-time setting

The first result of this section provides the solution to the corresponding fixed-time problem, for the mean variance objective¹⁰ that we analyzed in sections 2.3.2 and 2.4 and in which we have also introduced a dollar-neutrality soft constraint for the sake of illustration. As the next theorem shows, the optimal portfolios may be found again in an explicit, closed-form expression which is straightforward to implement numerically in our high-dimensional setting. Moreover, the solution of this discretized problem converges almost surely to the corresponding solution of the continuous-time case, regardless of the choice of the investing times.

Theorem 2.5.1. *The problem in (2.13) for a fixed sequence of times $\mathbf{t} = (t_1, \dots, t_L)$ and with the*

¹⁰The exponential case is more difficult in this discrete-time framework and involves the maximization of the moment generating function of quadratic forms in X , which in general cannot be done explicitly.

following objective function (mean-variance utility plus a running penalty for non dollar-neutrality) and no frictions

$$H_l(x, w; t) = \sup_{\pi \in \mathcal{A}_t} \mathbb{E}_{t_l, x, w} \left[W_{t_L} - \sum_{k=l}^{L-1} \left(\frac{\gamma(t_k)}{2} \text{Var}_{t_k}(W_{t_{k+1}}) + \frac{\alpha(t_k)}{2} (\pi_k \cdot p)^2 \Delta t_k \right) \right]$$

has, for $1 \leq l \leq L-1$, explicit optimal portfolio at time t_l given by

$$\pi_l^* = (\gamma(t_l)\Sigma(\Delta t_l) + \alpha(t_l)pp'\Delta t_l)^{-1} ((I_N - e^{-A\Delta t_l})(\mu - X_{t_l}) - pr\Delta t_l) \prod_{i=l+1}^{L-1} (1 + r\Delta t_i)$$

where $\Delta t_l = t_l - t_{l-1}$, and $\Sigma(\Delta t) = \int_0^{\Delta t} e^{-A(\Delta t-s)} \sigma \sigma' e^{-A'(\Delta t-s)} ds$.

Furthermore, independently of the chosen times $t = (t_1, \dots, t_L)$, as $\sup_{1 \leq l \leq L} |t_l - t_{l-1}| \rightarrow 0$ the optimal strategy described above converges almost surely to the corresponding one in the continuous-time case of Theorem 2.3.2 and Corollary 2.4.1.1.

Proof. See section 2.8.4. □

An important remark is that, by looking more closely at the proof of the above theorem, one can see that all the calculations involved do not require knowing what the admissible sets are, except for the computations of the closed-form expressions for the maxima π_l^* , which are never used in the proof. Hence, we obtain as a corollary the solution of the problem with arbitrary hard constraints:

Corollary 2.5.1. *For a general admissible set \mathcal{A}_{t_l} at time t_l , the optimal portfolio π_l^* to have in the above framework is given by*

$$\underset{\pi_l \in \mathcal{A}_{t_l}}{\operatorname{argmax}} \left(\pi_l'(\mu(X_{t_l}, \Delta t_l) - X_{t_l} - pr\Delta t_l) - \left(\frac{\gamma(t_l)}{2} \pi_l' \Sigma(\Delta t_l) \pi_l + \alpha(t_l) \pi_l' pp' \pi_l \Delta t_l \right) \right) \prod_{i=l+1}^{L-1} (1 + r\Delta t_i). \quad (2.14)$$

where $\mu(X_t, \Delta t) = e^{-A\Delta t} X_t + (I - e^{-A\Delta t})\mu$.

The previous corollary also provides a way of numerically solving the general continuous-time problem with both soft and hard constraints, since in the continuous-time case we could approximate the solution by discretizing the time grid with small Δt 's and then solve the problem in the discrete-time setting, whose convergence to the continuous-time case is partially justified by the last part of Theorem 2.5.1. Moreover, from a computational perspective, solving the problem would just require to solve L independent explicit deterministic optimization problems with a quadratic objective function over N variables, and in which the set of constraints would normally consist of convex functions¹¹. This computational simplicity should be contrasted with the much harder problem of

¹¹These include most of the commonly used constraints, such as long-short restrictions, bounds on the amount of

numerically solving the $(N + 2)$ -dimensional PDE given by the HJB equation corresponding to the continuous-time framework.

2.5.2 Optimizing over both times and allocations

To conclude this section, we discuss the general discrete-time problem in which, apart from optimizing her portfolios π_l at each of the discrete times $\mathbf{t} = (t_1, \dots, t_L)$, the investor also chooses the optimal deterministic times t_l at which to do so. Mathematically, this reduces to the double stochastic optimization problem with controls π and \mathbf{t} which is given by equation (2.13) above.

To address the new problem, we note that $t_i = t_l + \sum_{l+1 \leq j \leq i} \Delta t_j$ for $\Delta t_j = t_j - t_{j-1}$ and, using the Markov property of (X, W) and the equivalent controls π_l and Δt_l , we solve the corresponding stochastic optimization problem through the following Bellman equations, mimicking the proof of Theorem 2.5.1. above:

$$H_l(x, w) = \sup_{\pi_l \in \mathcal{A}_{t_l}, \Delta t_l \geq 0} \mathbb{E}_{t_l, x, w} [H_{l+1}(X_{t_{l+1}}, W_{t_{l+1}})]$$

for $1 \leq l \leq L - 1$, with $H_L(x, w) = u(w)$.

In particular, for the mean-variance framework with a dollar-neutrality soft constraint that we discussed in the last section, the previous equation is

$$H_l(x, w) = \sup_{\pi_l \in \mathcal{A}_{t_l}, \Delta t_l \geq 0} \mathbb{E}_{t_l, x, w} \left[H_{l+1}(X_{t_{l+1}}, W_{t_{l+1}}) - \left(\frac{\gamma(t_l)}{2} \pi_l' \Sigma(\Delta t_l) \pi_l + \frac{\alpha(t_l)}{2} \pi_l' p p' \pi_l \Delta t_l \right) \right]$$

for $1 \leq l \leq L - 1$ and $H_L(x, w) = w$, and where we have supposed that $r = 0$ for notational simplicity, although the general case $r \neq 0$ presents no additional mathematical difficulty.

Optimizing first with respect to π_l for fixed Δt_l 's and then with respect to the Δt_l 's in the above equations, the optimal portfolio $\pi_l^* = \pi_l^*(\Delta t_l, X_{t_l})$ may be found as in Theorem 2.5.1 for fixed Δt_l 's. Finally, looking again at the proof of Theorem 2.5.1 to see the resulting objective function and using one more time the Markov property of X , we get that the optimal Δt_l^* 's may be found via the equations

$$\sup_{\Delta t_l^* \geq 0} \sum_{i=l}^{L-1} \mathbb{E}_{0, X_{t_l}} \left[\pi_i^{*'} (\mu(X_{\sum_{j=l+1}^i \Delta t_j^*}, \Delta t_i^*) - X_{\sum_{j=l+1}^i \Delta t_j^*}) - \left(\frac{\gamma(t_l)}{2} \pi_i^{*'} \Sigma(\Delta t_i^*) \pi_i^* + \frac{\alpha(t_l)}{2} \pi_i^{*'} p p' \pi_i^* \Delta t_i^* \right) \right].$$

Although the previous equations give theoretically the optimal solution of the general problem, it should be noted that (1) even in this simplified mean-variance setting the optimal times must be computed numerically, and (2) unlike the case of π_l^* , which only depends on the current value of X and which may be computed without paying attention to the future values, the problem of

capital invested on each asset, leverage restrictions, etc., but not directly others such as cardinality restrictions on the total number of different assets.

simultaneously finding the optimal Δt_l^* 's requires a forward-backward simulation and optimization process, which makes the problem much harder even in this simplified setting in which the times are deterministic.

2.6 Monte Carlo simulations

We conclude the chapter by providing some high-dimensional numerical simulations that give new insights about the behavior of the previously discussed strategies and their sensitivity to the different parameters, emphasizing in a separate simulation the role of the factor model and its connection with market-neutrality. To this end, we first simulate X in the case of $N = 100$ residuals by using exact Monte Carlo sampling along a discrete time grid, and we execute the previous strategies for some parametric choices of X and some values of p to compute sample paths of π_t and W_t and histograms of the terminal Profit & Loss (P&L). We have however opted to defer experiments with real data to a separate paper, since examining carefully the delicate issues of asset selection, rebalance frequency, construction of the factor models and obtention of X , high-dimensional parameter estimation and updating, risk control, etc. that the problem requires would be impossible to consider here without prohibitively extending the length of the paper.

During all this section, we therefore fix the following parameters for our model:

$$N = 100, \quad \mu = 0, \quad X_0 = \mu,$$

A is diagonal with entries drawn i.i.d from a normal distribution of mean 0.5 and standard deviation 0.1, and the coefficients of σ are drawn i.i.d from a uniform distribution in $[-0.3, 0.3]$, except for the diagonal elements which are drawn from a uniform distribution in $[0, 0.5]$. Furthermore $p = \mathbf{1}$ for the first simulations, and we will also perturb it later to study different factor model regimes and the impact of imposing market neutrality. We also fix a finite horizon of $T = 20$ and a temporal grid $0 = t_0 < t_1 < \dots < t_L = T$ obtained by discretizing $[0, T]$ with constant $\Delta t = T/L = 20/400 = 0.5$.

From a financial perspective, the above choice of parameters means that the 100 coordinates of X are correlated and mean-revert with similar speeds (given by the eigenvalues of A) to an equilibrium of 0, describing an average number of approximately 10 oscillation cycles of ups and downs before the terminal time (given by the product of T and the average mean-reversion speed). The choice of $p = \mathbf{1}$ arises when the asset returns themselves are mean-reverting and may be modeled directly by X so we can take $\Lambda = 0$ in our factor model, while the perturbations of p will imply departing from this assumption to factor models with heavier loadings, in which imposing market neutrality leads to more leveraged positions. As an illustration, we plot some sample paths of the first three coordinates of such a process X in Figure 2.1 below.

We then sample $M = 1,000$ paths of X on this grid exactly with standard Monte Carlo techniques

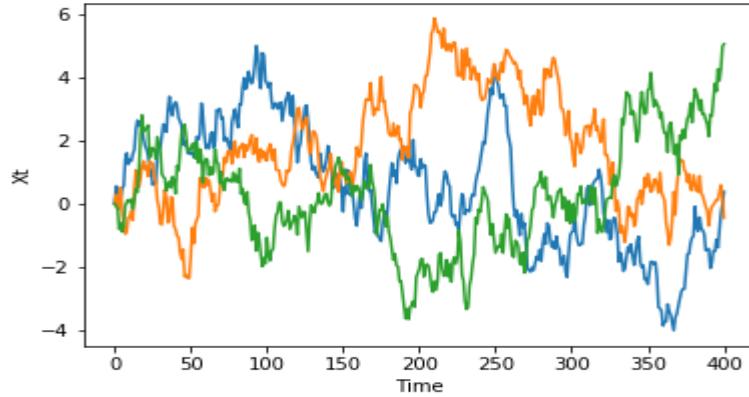


Figure 2.1: Sample paths of the first three coordinates of X in $[0, T]$

by using the fact that, since

$$X_{t+\Delta t} = e^{-A\Delta t} X_t + (I - e^{-A\Delta t})\mu + \int_t^{t+\Delta t} e^{-A(\Delta t+s-t)} \sigma dB_s,$$

$X_{t+\Delta t}|X_t \sim N(\mu(X_t, \Delta t), \Sigma(\Delta t))$, where

$$\mu(X_t, \Delta t) = e^{-A\Delta t} X_t + (I - e^{-A\Delta t})\mu, \quad \Sigma(\Delta t) = \int_0^{\Delta t} e^{-A(\Delta t-s)} \sigma \sigma' e^{-A'(\Delta t-s)} ds,$$

and execute the following strategies¹² at the corresponding times t_l 's, with $W_0 = \pi_0 = 0$ and π_t constant between consecutive times:

1. The exponential utility strategy of Theorem 2.3.1 with $\gamma(t) = 1, 2, 3, 4$ and $r = 0, 2\%$.
2. The mean-variance utility strategy with dollar-neutrality penalty of Corollary 2.4.1.1 with $\gamma(t) = 1, 2, 3, 4$, $\alpha(t) = 0, 20, 50$ and $r = 2\%$.
3. The mean-variance utility strategy with quadratic transaction costs of Theorem 2.4.1 with $\gamma(t) = 1, 2, 3, 4$, $r = 2\%$, and $C = \lambda \sigma \sigma'$ for $\lambda = 0.1, 0.5, 1$.

Moreover, to study the result of imposing market-neutrality through the market-neutral portfolios constructed in section 2.2 under different factor model regimes, we perform the following additional simulation in which we experiment with the parameter p , which encapsulates all the factor model information and which we perturb to simulate the effect of going away from the case where the

¹²We have just simulated some simple cases of the previously discussed strategies for space limitation reasons, but it would be interesting as well to execute the strategies with some perturbations of the real parameters to simulate possible microstructural noise and imperfect estimation.

returns themselves are mean-reverting (which corresponds to the previous case $p = \mathbf{1}$) and of having progressively more leveraged market-neutral portfolios:

4. The three strategies above with $\gamma(t) = 1, \alpha(t) = 0$ and $r = 2\%$ (and $\lambda = 1$ for the third strategy) for $p = \mathbf{1} + \epsilon_a$ and $a = 1, 2, 4, 8$, where ϵ_a is a N -dimensional vector whose components are drawn i.i.d. from a uniform distribution in $[-a, a]$.

We present the simulated path of a sample wealth process $(W_t)_{t \in [0, T]}$, the simulated path of the first coordinate of a sample allocation process $(\pi_t)_{t \in [0, T]}$, and the histogram for the terminal wealth W_T for each of the above cases in the following four subsections, along with a final analysis:

2.6.1 Simulations of the exponential-utility strategy

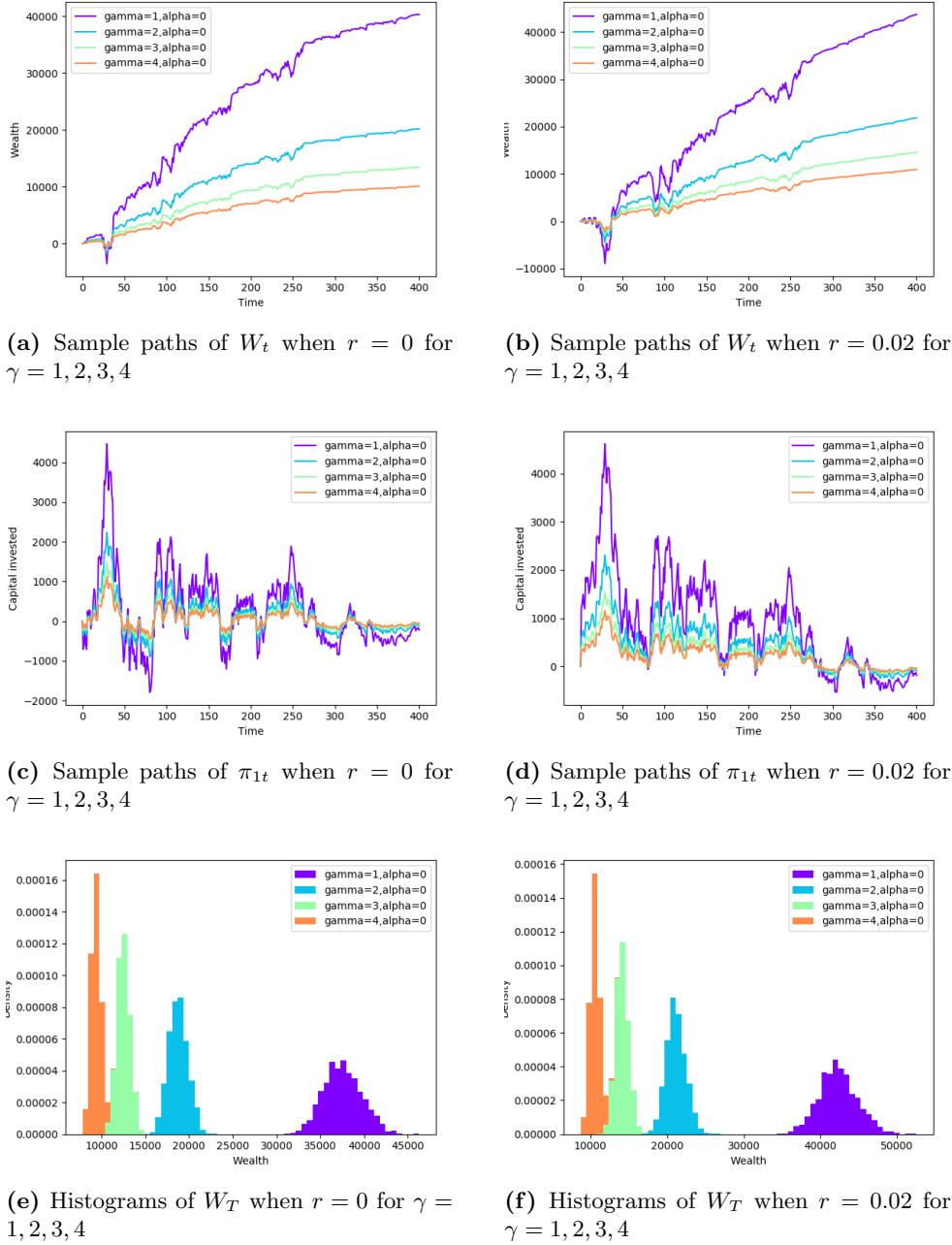


Figure 2.2: Results for the exponential utility

2.6.2 Simulations of the mean-variance strategy with dollar neutrality

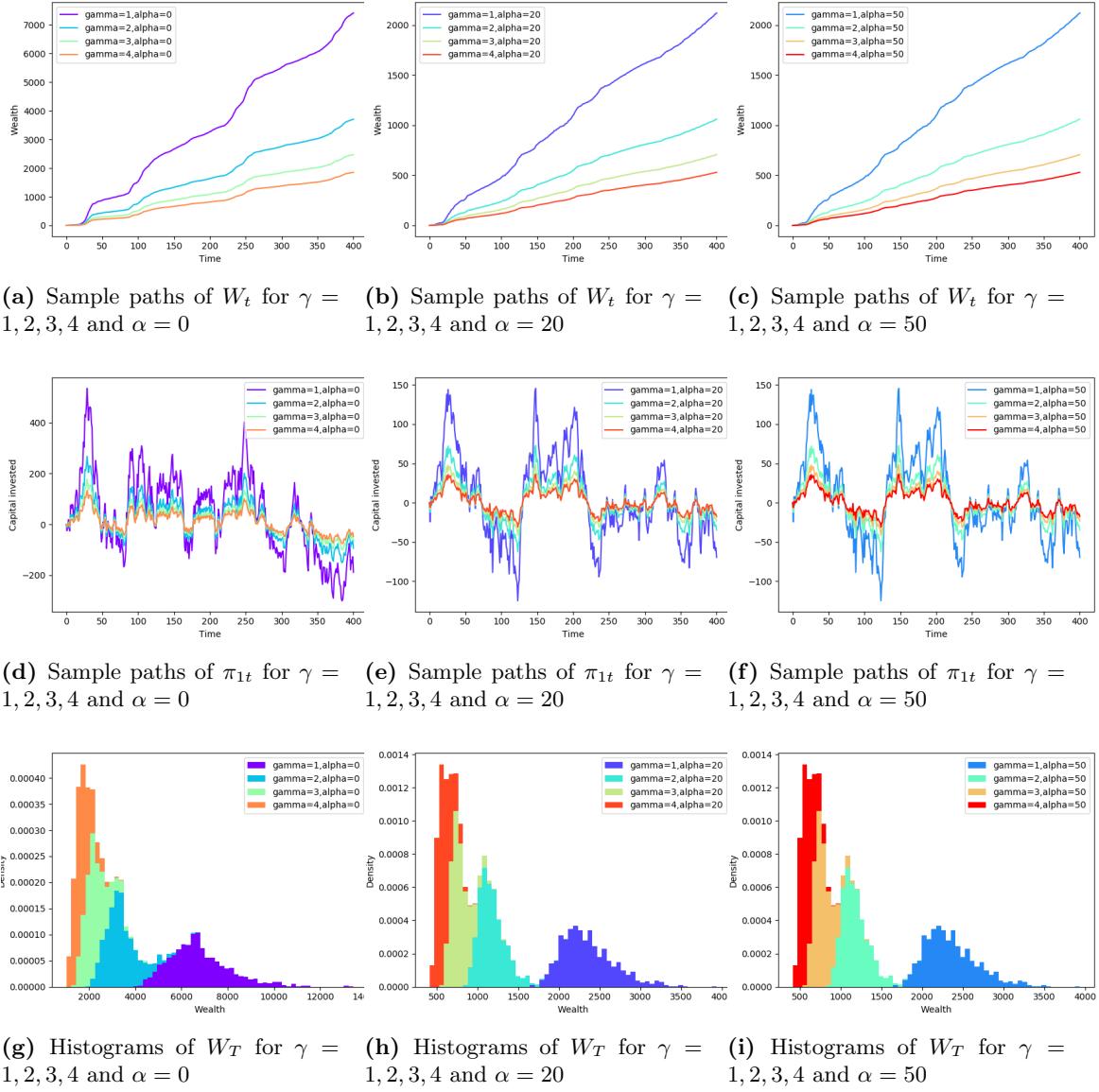


Figure 2.3: Results for the mean-variance utility when $r = 0.02$ with different dollar-neutrality restrictions

2.6.3 Simulations of the mean-variance strategy with quadratic transaction costs

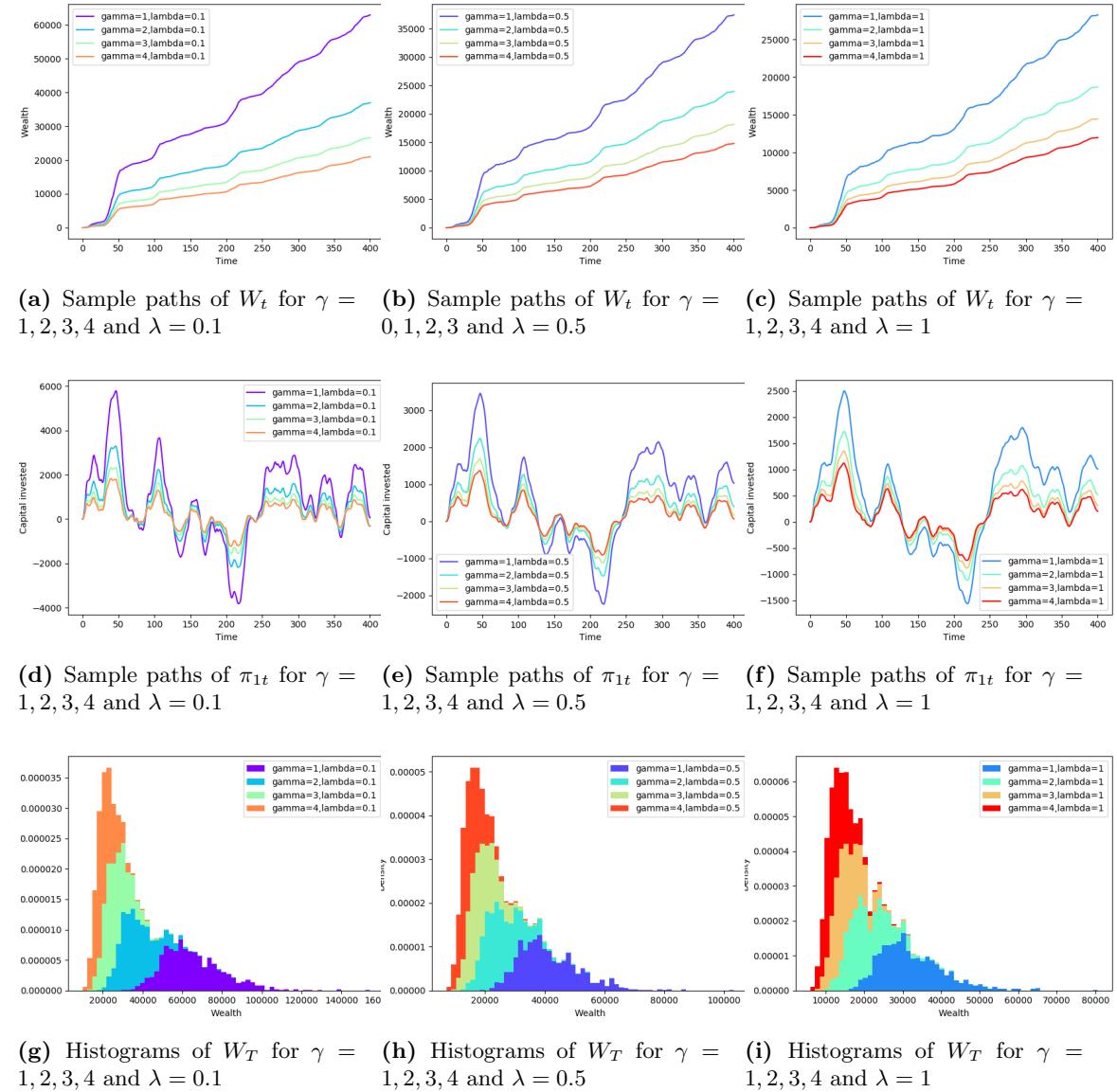


Figure 2.4: Results for the mean-variance utility when $r = 0.02$ with different quadratic transaction costs $C = \lambda\sigma\sigma'$

2.6.4 Simulations for different factor model and market-neutrality regimes

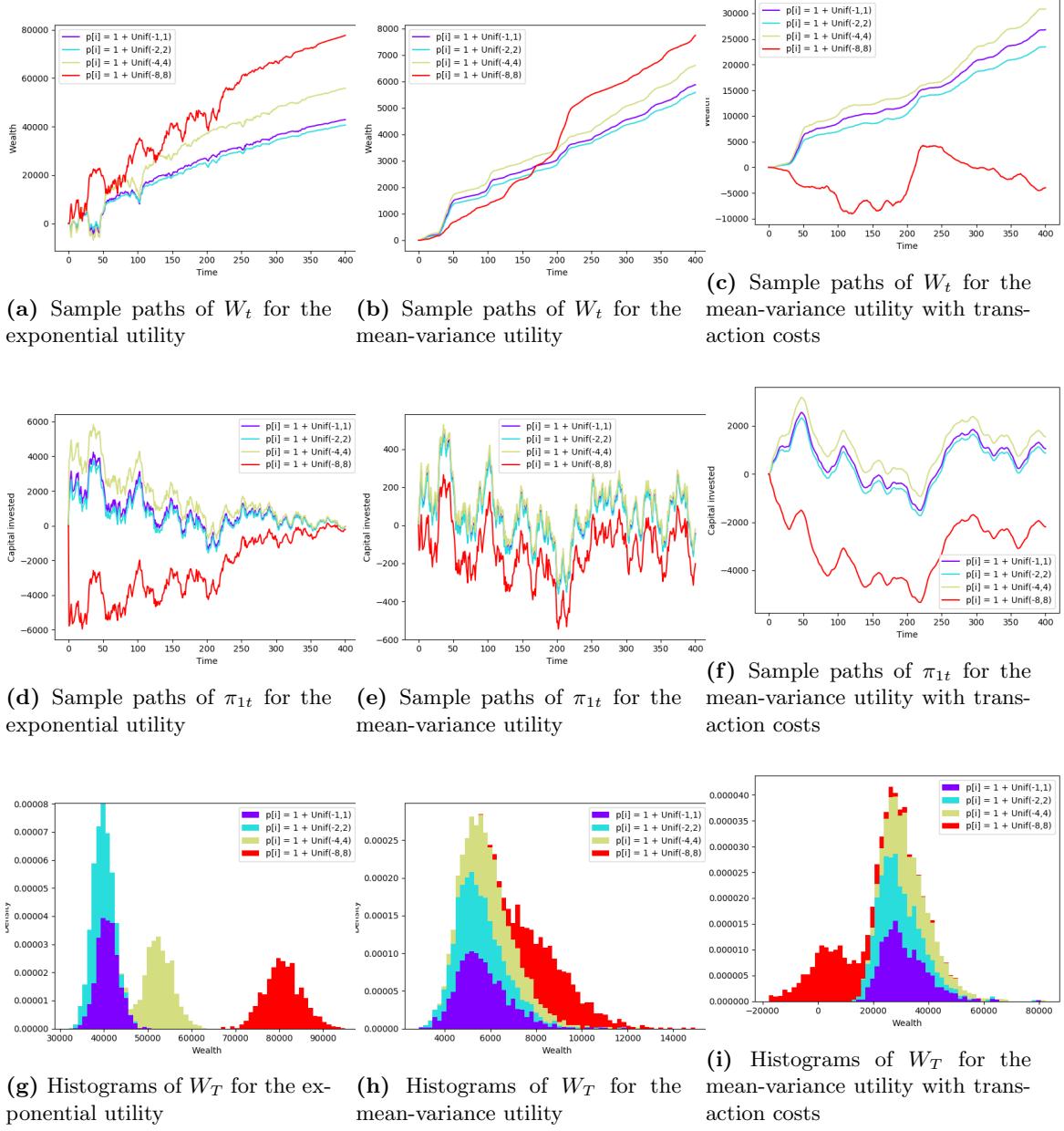


Figure 2.5: Results for the three strategies above with different p 's, when $r = 0.02, \alpha = 0, \gamma = 1, \lambda = 1$

2.6.5 Comparison of the simulated strategies

We now present our main conclusions after observing the previous plots, analyzing the behavior of the histograms of the final wealth W_T , the sample paths of the wealth process $(W_t)_{t \in [0, T]}$, and the sample paths of the positions $(\pi_{1t})_{t \in [0, T]}$, with a final subsubsection discussing the effect of imposing market-neutrality under different p 's.

Histograms of the final wealth

Looking first at the above histograms (Figures 2.3-2.4 (g)-(i), and 2.2 (e)-(f)), we see that, for our parametric choice and our setting in which X is effectively a multidimensional Ornstein-Uhlenbeck process with known parameters,

1. The most profitable strategy is the one derived from the exponential utility (Figure 2.2, (e) and (f)) with the lowest risk-aversion parameter γ , even in the most adverse scenarios of the histogram and both with and without zero interest rates. Moreover, even for bigger values of γ this strategy significantly performs better under any regime of r and α than the mean-variance strategy (Figure 2.3, (g)-(i)).
2. We observe the following outcomes when changing one of the parameters for each of the three strategies (Figures 2.3-2.4 (g)-(i), and 2.2 (e)-(f)):
 - Increasing the value of the risk-aversion parameter γ produces a concentration of the density of W_T around smaller values, i.e., the expected wealth decreases and so does the dispersion around it.
 - Increasing the dollar-neutrality penalty α has this same negative effect, but makes little difference unless the increments in α are considerable.
 - Increasing the value of the interest-rate r has an overall positive effect, which is more pronounced in the mean-variance case since, as we mentioned at the end of section 2.2.2, the investor is then more aggressive than the exponential agent.
 - Increasing the transaction cost parameter λ decreases the expected terminal wealth, but it also skews its distribution producing a considerable right-tail (whereas all the other distributions are essentially symmetric).

These outcomes have a natural interpretation: since the model is perfectly specified and the parameters are known, the derived strategies will always produce benefits by construction, and they will be bigger the fewer additional constraints we impose (such as risk-aversion, dollar-neutrality, and transaction costs) and the more we can take advantage of previous success (by increasing r). This situation, however, might not apply under parameter misspecification, where the additional constraints would help the investor mitigate the model risk.

Sample paths of the wealth process W_t

Examining next the sample paths for the particular simulation which is plotted (Figures 2.3,2.4 (a)-(c), and 2.2 (a)-(b)), we observe the same patterns as discussed in the previous paragraph when modifying the parameters γ, α, r and γ . There are, however, two new observations:

1. In the three strategies, after an initial period of ups and downs and similarity between the different strategies, there is a tendency towards stabilization because of the asymptotic properties of the Ornstein-Uhlenbeck process, and of differentiation depending on the parametric choices.
2. This phenomenon is especially pronounced with the exponential utility and with bigger values of r (Figure 2.2 (a)-(b)) since it takes more aggressive positions, reflecting the fact that sometimes the agent will invest more capital than what she will make at that moment (and sometimes even having temporary negative wealth and borrowing aggressively) to continue executing the strategy.

Sample paths of the positions π_{1t}

Considering now the plots of the sample paths of the positions π_{1t} (Figures 2.3,2.4 (d)-(f), and 2.2 (c)-(d)), we similarly notice that

1. The positions become more extreme when decreasing γ, α and λ (i.e., the risk-aversion parameter, the non-dollar-neutrality penalty and the transaction cost parameter) and when increasing r (the interest rate). The greatest overall impact is produced by γ and λ and then r , especially in the mean-variance case for the same reasons as before.
2. The exponential utility strategy takes more extreme positions than the mean-variance strategies, which in this idealized setting of perfect estimation partially explains why the exponential agent obtains a greater wealth at the terminal time.
3. The cycles in the positions π_{1t} match the oscillations of X_{1t} depicted in Figure 2.1, as described theoretically in the corresponding equations.

Effects of imposing market neutrality

Finally, looking separately at the effect of imposing market neutrality under various factor model regimes depending on p (which, as we mentioned, depends quadratically on the factor model loadings), we observe the following (Figure 2.5):

1. As the parameter p gets bigger, the market neutral portfolios of section 2.2 become more extreme and the adopted positions π_t also become more aggressive, especially in the exponential utility case (Figure 2.5, (d)-(f)).

2. Since the strategy is more aggressive but we have perfect estimation, with bigger p the mean-variance and especially the exponential strategy become more profitable. However, the wealth process also has more ups and downs (Figure 2.5, (a)-(c)), the standard deviation of the terminal wealth increases considerably (Figure 2.5, (g)-(h)), especially in the mean-variance case, and with the biggest p there are also heavy losses when transaction costs are incorporated (Figure 2.5, (c),(i)). The strategies are therefore riskier, but a relatively large value of p is needed to appreciate its effect.
3. Lastly, note that the influence of p on the strategies also depends most of the time on the value of r , since they normally appear combined as a factor of rp in the equations describing the strategies. In particular, when $r = 0$ there is no theoretical effect associated to p (apart from possible model risk and high leverage in a real-world setting) unless the dollar-neutrality parameter $\alpha(t) \neq 0$.

2.7 Conclusions and further research

In this chapter we have aimed to provide a systematic study of high-dimensional statistical arbitrage combining stochastic control and factor models. To this end, we have first proposed a general framework based on a statistically-constructed factor model, and then shown how to obtain analytically explicit market-neutral portfolios and rephrase our problem in terms of them to make it tractable and get market neutrality by construction. Using this insight, we have studied the question of optimizing the expected utility of the investor's terminal wealth in continuous and discrete time under both an exponential and a mean-variance objective. In both cases, we have obtained explicit closed-form solutions that avoid potentially unfeasible high-dimensional numerical methods, analyzed the corresponding strategies from the perspective of statistical arbitrage and the underlying factor model, and discussed extensions involving the addition of soft constraints on the admissible portfolios (like dollar-neutrality) and the presence of temporary quadratic transaction costs. Finally, we have run Monte Carlo simulations to explore the behavior of the previous strategies, and analyzed their qualitative aspects and their sensitivity to the relevant parameters and the underlying factor model.

There are four natural extensions to the work presented in this chapter. First, one could investigate the more realistic version of the discrete-time problem in which, rather than being deterministic, the rebalancing times are stochastic, generalizing the literature initiated by [54]. Second, it would be interesting to study the effects of different scales or partial information within the residual portfolios constructed in section 2.2. Third, on a more empirical side and as we mentioned at the start of the section 2.6, one should consider in this setting the problems of construction of the factor models and of high-dimensional parameter estimation, along with out-of-sample experiments with real market data under the strategies developed in this chapter. Fourth, one could study a more data-driven

version of the problem, where the fixed stochastic model is replaced by new tools from deep learning. These last two avenues of research will be part of the subject of the next chapter of this dissertation, as explained in more detail in Chapter 1.

2.8 Proofs

2.8.1 Proof of Theorem 2.3.1

Proof of Proposition 2.3.1:

The dynamic programming principle suggests that the value function H should satisfy the dynamic programming equation (2.5) with terminal condition $H(T, x, w) = -e^{-\gamma w}$. The optimal control may then be found in feedback form by looking at the first order condition of the term inside the supremum, since the corresponding function is quadratic and concave in π (if $\partial_{ww}H < 0$, i.e., if there is risk aversion). The first order condition gives that

$$0 = \sigma\sigma'\partial_{ww}H\pi + (A(\mu - x) - pr)\partial_wH + \sigma\sigma'\nabla_{xw}H,$$

and solving for π we find the control given in the proposition's statement. Putting it back into (2.5) we get the following non-linear and $(N + 2)$ -dimensional PDE

$$0 = \partial_tH + (A(\mu - x))'\nabla_xH + \frac{1}{2}\text{Tr}(\sigma\sigma'\nabla_{xx}H) + wr\partial_wH - \frac{\mathcal{D}H'(\sigma\sigma')^{-1}\mathcal{D}H}{2\partial_{ww}H}. \quad (2.15)$$

Now, looking at the terminal condition, we guess that the solution of this PDE will be of the form $H(t, x, w) = -\exp(-\gamma(w e^{r(T-t)} + h(t, x)))$ for some function $h(t, x)$ to be determined and such that $h(T, x) = 0$. Some easy computations then show that

$$\begin{aligned} \partial_tH &= -\gamma H(-rwe^{r(T-t)} + \partial_t h) & \partial_wH &= -\gamma e^{r(T-t)}H & \partial_{ww}H &= \gamma^2 e^{2r(T-t)}H & \nabla_{xw}H &= \gamma^2 e^{r(T-t)}H\nabla_xh \\ \nabla_xH &= -\gamma H\nabla_xh & \nabla_{xx}H &= -\gamma H(\nabla_{xx}h - \gamma\nabla_xh\nabla_xh') & \mathcal{D}H &= -\gamma e^{r(T-t)}H(A(\mu-x) - pr - \gamma\sigma\sigma'\nabla_xh). \end{aligned}$$

Plugging all this into (2.15), dividing everything by $-\gamma H$, and doing some simple algebra to expand the last term yields

$$\begin{aligned} 0 &= -rwe^{r(T-t)} + \partial_t h + (A(\mu - x))'\nabla_xh + \frac{1}{2}\text{Tr}(\sigma\sigma'(\nabla_{xx}h - \gamma\nabla_xh\nabla_xh')) + wre^{r(T-t)} + \\ &\quad \frac{1}{2\gamma}(A(\mu - x) - pr)'(\sigma\sigma')^{-1}(A(\mu - x) - pr) + \frac{\gamma}{2}\nabla_xh'\sigma\sigma'\nabla_xh - (A(\mu - x) - pr)'\nabla_xh \end{aligned}$$

and we can see that the non-linear terms in h , the terms in w , and the third and part of the last term of the PDE get cancelled and the equation gets considerably simplified, obtaining

$$0 = \partial_t h + \frac{1}{2} \text{Tr}(\sigma\sigma' \nabla_{xx} h) + rp' \nabla_x h + \frac{1}{2\gamma} (A(\mu - x) - pr)' (\sigma\sigma')^{-1} (A(\mu - x) - pr).$$

This is now a parabolic linear PDE in h and we can find explicitly its solution by using the Feynman-Kac formula. Indeed, if we consider the stochastic process given by

$$dY_t = rpdt + \sigma dB_t^* \quad (2.16)$$

we can rewrite the above equation in terms of the infinitesimal generator \mathcal{L}^* of Y as

$$0 = (\partial_t + \mathcal{L}^*)h + \frac{1}{2\gamma} (A(\mu - x) - pr)' (\sigma\sigma')^{-1} (A(\mu - x) - pr)$$

and then we can express its solution via the following conditional expectation, which is the probabilistic representation given in the proposition's statement:

$$\begin{aligned} h(t, x) &= \mathbb{E}_{t,x}^* \left[\int_t^T \frac{1}{2\gamma} (A(\mu - Y_s) - pr)' (\sigma\sigma')^{-1} (A(\mu - Y_s) - pr) ds \right] \\ &= \frac{1}{2\gamma} (A\mu - pr)' (\sigma\sigma')^{-1} (A\mu - pr)(T - t) \\ &\quad - \frac{1}{\gamma} (A\mu - pr)' (\sigma\sigma')^{-1} A \mathbb{E}_{t,x}^* \left[\int_t^T Y_s ds \right] + \frac{1}{2\gamma} \mathbb{E}_{t,x}^* \left[\int_t^T Y_s' A' (\sigma\sigma')^{-1} A Y_s ds \right]. \end{aligned}$$

Finally, to find h explicitly, notice that we can easily solve the SDE (2.16), obtaining, for $s \geq t$,

$$Y_s = x + rp(s - t) + \sigma(B_s^* - B_t^*).$$

and this allows us to compute the two expectations in our expression for h above. Indeed, Fubini's theorem and elementary facts about the Brownian motion immediately yield

$$\mathbb{E}_{t,x}^* \left[\int_t^T Y_s ds \right] = \int_t^T \mathbb{E}_{t,y}^* [Y_s] ds = x(T - t) + rp \frac{(T - t)^2}{2}$$

and, interchanging integral and expectation again and noticing that

$$\mathbb{E}^*[(B_s^* - B_t^*)' \sigma' A' (\sigma\sigma')^{-1} A \sigma (B_s^* - B_t^*)] = (s - t) \text{Tr}(\sigma' A' (\sigma\sigma')^{-1} A \sigma),$$

we similarly find out that

$$\begin{aligned} \mathbb{E}_{t,x}^* \left[\int_t^T Y_s' A'(\sigma\sigma')^{-1} A Y_s ds \right] &= \int_t^T (x + rp(s-t))' A'(\sigma\sigma')^{-1} A (y + rp(s-t)) + (s-t) \text{Tr}(\sigma' A'(\sigma\sigma')^{-1} A \sigma) ds \\ &= x' A'(\sigma\sigma')^{-1} A x (T-t) + (2x' A'(\sigma\sigma')^{-1} A r p + \text{Tr}(\sigma' A'(\sigma\sigma')^{-1} A \sigma)) \frac{(T-t)^2}{2} + r^2 p' A'(\sigma\sigma')^{-1} A p \frac{(T-t)^3}{3}, \end{aligned}$$

which gives us the complete explicit solution of the DPE, and hence the explicit form of the optimal strategy π^* by using equation (2.6). \square

Proof of Proposition 3.2:

Since in the previous proof we have found explicitly the classical smooth solution H of the dynamic programming equation, we just have to check that $\pi^* \in \mathcal{A}_{[0,T]}$ and that the usual regularity conditions hold for the classical proof to apply. More precisely, this means that the local martingale $dH - \mathcal{L}_{t,x,w}^\pi H dt$ is a supermartingale for any admissible π and a true martingale for π^* , where $\mathcal{L}_{t,x,w}^\pi$ is the infinitesimal generator of the controlled process (X, W^π) , or some sufficient condition for this like the one we stated in Proposition 2.3.2 in terms of the model parameters, which is what we will show here.

As for the first issue, it is easy to see that $\pi^* \in \mathcal{A}_{[0,T]}$. Indeed, it is obviously \mathcal{F}_t -adapted and predictable (in fact, it has continuous paths) and, using the trivial inequalities $\|x + y\|^2 \leq 2\|x\|^2 + 2\|y\|^2$ and $\|Ax\| \leq \|A\|\|x\|$ and the fact that X_t is a Gaussian process, it is easy to see that $\int_0^T \mathbb{E}[\|\pi_s^*\|^2] ds < \infty$. Moreover, applying Ito's formula to the process $e^{-rt}W_t$ yields

$$d(e^{-rt}W_t) = -re^{rt}W_t dt + e^{-rt}dW_t = \pi_t \cdot e^{-rt}dX_t - \pi_t \cdot e^{-rt}prdt$$

and, therefore,

$$W_t = w + e^{rt} \left(\int_0^t \pi_s \cdot e^{-rs}dX_s - \int_0^t \pi_s \cdot e^{-rs}prds \right) \quad (2.17)$$

for any $t \geq 0$ and any admissible control π . Thus, the SDE for W has a unique strong solution W^* for the particular case $\pi = \pi^*$ for any initial data, given by the above integral (note that the stochastic integral is well defined, since $dX_s = A(\mu - X_s)ds + \sigma dB_s$, π^* and X are continuous, and again $\int_0^T e^{-2rs}\mathbb{E}[\|\pi_s^*\sigma\|^2]ds < \infty$).

As for the regularity conditions, we adapt the proof of Theorem 2.1. of [59], which guarantee the uniform \mathbb{P} -integrability of the family of random variables $(H(\tau, X_\tau, W_\tau^*))_{\tau \in [0,T]}$ where τ is a \mathcal{F} -stopping time, and which we adapt to the parameters of the present model obtaining the sufficient conditions stated in Proposition 2.3.2.

The key observation to adapt their proof is that in our case we also have that the hypothesized value function is of the form $H(t, x, w) = -\exp(-\gamma w e^{r(T-t)} - \frac{1}{2}x' A_2(t)x - A_1(t)x - A_0(t))$ for some explicit smooth functions $A_i(t)$ that we computed in the proof of Proposition 2.3.1, and our X is

also a matrix Ornstein-Uhlenbeck process under \mathbb{P} with SDE $dX_t = A(\mu - X_t)dt + \sigma dB_t$, and

$$\gamma W_\tau^* e^{r(T-\tau)} = \gamma w e^{r(T-\tau)} + \gamma \int_0^\tau \pi_s^* \cdot e^{r(T-s)} (A(\mu - X_s) - pr) ds + \gamma \int_0^\tau \pi_s^* \cdot e^{r(T-s)} \sigma dB_s$$

as we showed in (2.17). Thus, using the Cauchy-Schwarz inequality as in their proof, the part corresponding to $-\frac{1}{2} X'_\tau A_2(\tau) X_\tau - A_1(\tau) X_\tau - A_0(\tau)$ in the above expression for $H(\tau, X_\tau, W_\tau^*)$ may be bounded following their reasoning. As for the part corresponding to $-\gamma W_\tau^* e^{r(T-\tau)}$, we can again repeat their argument, but noting that the quadratic term in X_s in the first integral above is now $X'_s C_0(s) X_s$ for the matrix $C_0(s)$ that we defined before, and likewise the term in X_s in the second integral is $X'_s C_1(s)$, which following their proof gives respectively the two explicit sufficient conditions that we stated in Proposition 2.3.2. \square

2.8.2 Proof of Theorem 2.3.2

The proof of this follows the same lines as the previous one and is simpler, so we just indicate the relevant changes. The HJB equation is now

$$0 = \partial_t H + (\mu - x)' A' \nabla_x H + \frac{1}{2} \text{Tr}(\sigma \sigma' \nabla_{xx} H) + \sup_{\pi} \left((\pi' A(\mu - x) + (w - \pi' p)r) \partial_w H + \frac{1}{2} \pi' \sigma \sigma' \pi \partial_{ww} H + \pi' \sigma \sigma' \nabla_{xw} H - \frac{\gamma(t)}{2} \pi' \sigma \sigma' \pi \right)$$

with terminal condition $H(T, x, w) = w$.

Guessing that the value function will now be of the form $H(t, x, w) = w e^{r(T-t)} + a(t)' x + \frac{1}{2} x' c(t) x$ for a scalar $a(t)$, an N -dimensional vector $b(t)$, and a symmetric $N \times N$ matrix $c(t)$, and plugging this into the above equation, we obtain the hypothesized optimal control given in the statement of the theorem and the above PDE gets reduced to the following system of three first-order linear matrix ODEs

$$\begin{aligned} 0 &= \partial_t c - A' c - c A + e^{2r(T-t)} A' (\gamma(t) \sigma \sigma')^{-1} A \\ 0 &= \partial_t b - A' b + c A \mu - e^{2r(T-t)} A' (\gamma(t) \sigma \sigma')^{-1} (A \mu - pr) \\ 0 &= \partial_t a + \frac{1}{2} (\mu' A' b + b' A \mu) + \frac{1}{2} \text{Tr}(\sigma \sigma' c) + \frac{e^{2r(T-t)}}{2} (A \mu - pr)' (\gamma(t) \sigma \sigma')^{-1} (A \mu - pr) \end{aligned}$$

with terminal conditions $a(T) = b(T) = c(T) = 0$.

This system has an explicit bounded solution in $[0, T]$, since the classical solution of the general first-order linear matrix ODE $\partial_t y + u y + v(t) = 0$ with $y(T) = 0$ is given by

$$y(t) = \int_t^T \exp((s-t)u) v(s) ds,$$

if $v(s)$ is continuous on $[0, T]$, in which case it is automatically bounded on $[0, T]$ as well; and similarly

the classical solution of $\partial_t y + uy + yu' + v(t) = 0$ with $y(T) = 0$ for a symmetric v is given by

$$y(t) = \int_t^T \exp((s-t)u) v(s) \exp((s-t)u') ds.$$

Thus, the HJB equation has an explicit classical solution which has *quadratic growth in the state variables uniformly in t*. A classical verification result (cf. for example Theorem 4.3 of [38]) then yields that our hypothesized optimal control is indeed optimal provided that it is admissible, which may be checked exactly as in the proof of Theorem 2.3.1. \square

2.8.3 Proof of Theorem 2.4.1

Proof of Proposition 2.4.1:

The corresponding dynamic programming equation is in this case

$$\begin{aligned} 0 = & (\partial_t + \mathcal{L}_X)H + (\pi' A(\mu - x) + (w - \pi' p)r) \partial_w H + \frac{1}{2} \pi' \sigma \sigma' \pi \partial_{ww} H + \\ & + \pi' \sigma \sigma' \nabla_{xw} H - \frac{\gamma(t)}{2} \pi' \sigma \sigma' \pi + \sup_I \left(I' \nabla_\pi H - \frac{1}{2} I' C I \right) \end{aligned}$$

with terminal condition $H(T, x, w, \pi) = w$ and where the supremum is obviously attained at $I^* = C^{-1} \nabla_\pi H$.

Substituting this back in the above equation and plugging the stated ansatz we obtain that

$$\begin{aligned} 0 = & \frac{1}{2} \pi' \partial_t a \pi + \pi' (\partial_t + \mathcal{L}_X) b + (\partial_t + \mathcal{L}_X) d + \pi' (A(\mu - x) - pr) e^{r(T-t)} \\ & - \frac{\gamma(t)}{2} \pi' \sigma \sigma' \pi + \frac{1}{2} (a\pi + b)' C^{-1} (a\pi + b). \end{aligned}$$

Matching the coefficients for $\pi'(\cdot)\pi$, $\pi'(\cdot)$, and the constant yields the above differential equations. \square

Before we prove the next proposition, we state here the following result for comparison and existence of solutions of matrix Riccati ODEs (cf. Theorem 2.2.2 in [46]), which we will use in our proof.

Theorem 2.8.1. *Let $L_1(t), L_2(t), L(t), N_1(t), N_2(t) \in \mathbb{R}^{d \times d}$ be piecewise continuous on \mathbb{R} . Moreover, suppose $L_1(t), L_2(t), N_1(t), N_2(t)$ and $S_1, S_2 \in \mathbb{R}^{d \times d}$ are symmetric. Let $T > 0$ and*

$$S_1 \geq S_2, L_1 \geq L_2 \geq 0, N_1 \geq N_2,$$

on $[0, T]$. Assume that the terminal value problem

$$\partial_t H_1 + H_1 L_1 H_1 + M H_1 + H_1 M + N_1 = 0, \quad H_1(T) = S_1,$$

has a (symmetric) solution H_1 on $[0, T]$. Then the terminal value problem

$$\partial_t H_2 + H_2 L_2 H_2 + M H_2 + H_2 M + N_2 = 0, \quad H_2(T) = S_2,$$

has a (symmetric) solution H_2 on $[0, T]$ and $H_1(t) \geq H_2(t)$ for all $t \in [0, T]$.

We are now in a position to give the following:

Proof of Proposition 2.4.2:

1. The first statement follows directly from the comparison Theorem 2.8.1 stated before, since the matrix Riccati ODE

$$\partial_t a + a C^{-1} a = 0$$

with terminal condition $a(T) = 0$ has the obvious symmetric solution $a(t) = 0$ defined on all $[0, T]$. Thus, (2.8) has a symmetric classical solution $a(t)$ defined on all $[0, T]$ with $a \leq 0$, which is bounded because $[0, T]$ is compact and a is differentiable hence continuous.

As for the particular solution when $\gamma(t) = \gamma$, simply note that pre- and post-multiplying (2.8) by $C^{-1/2}$ and defining $\tilde{a} := C^{-1/2} a C^{-1/2}$ gives the new Riccati

$$\partial_t \tilde{a} - \gamma C^{-1/2} \sigma \sigma' C^{-1/2} + \tilde{a}^2 = 0,$$

whose solution is $\tilde{a}(t) = D \tanh(D(t - T))$.

2. The existence of solutions with polynomial growth and their probabilistic representation in the above form follow from a vector-valued version of the Feynman-Kac theorem (see Appendix A.3 of [12] for a proof of how to adapt the one-dimensional case) provided that the appropriate regularity conditions hold. Using, for example, Condition 2 of Appendix E in [20], it is sufficient that all the functions of (t, x) $A(\mu - x)$, σ , $a(t)' C^{-1}$, $e^{r(T-t)}(A(\mu - x) - rp)$ (and $b(t, x)' C^{-1} b(t, x)$ for the existence of d) are uniformly Lipschitz in x , they and their first and second derivatives in x are continuous with polynomial growth in x uniformly in t , and $a(t) \leq 0$. All of these properties are straightforward to check in this case because all the corresponding functions are given explicitly and are simple, and the required properties for a follow from (1).

The fact that b has linear growth in x uniformly in t is then a consequence of the probabilistic representation (2.11). Indeed, Fubini's theorem implies that

$$b(t, x) = \int_t^T : \exp \left(\int_t^s a'(u) C^{-1} du \right) : e^{r(T-s)} (A \mathbb{E}_{t,x} [\mu - X_s] - rp) ds$$

whereas the fact that

$$X_{t+\Delta t} = e^{-A\Delta t} X_t + (I - e^{-A\Delta t}) \mu + \int_t^{t+\Delta t} e^{-A(\Delta t+s-t)} \sigma dB_s$$

yields

$$\mathbb{E}_{t,x} [\mu - X_s] = e^{-A(s-t)}(\mu - x).$$

Combining the two pieces and using the boundedness of a and the compactness of $[0, T]$ gives the desired uniform bound in t .

The quadratic growth of d in x uniformly in t is then obvious looking at its probabilistic representation and using the linear growth of b . \square

Proof of Proposition 2.4.3:

Combining the two previous propositions, we have already found an explicit classical solution of the associated HJB equation with quadratic growth in the state variables uniformly in t , so using again Theorem 4.3 in [38], we just have to verify that the candidate intensity given in Proposition 2.4.1 is admissible.

For this, first of all note that the corresponding SDEs controlled by the above intensity have a unique strong solution for any initial data. Indeed, given I^* and the definition of I as $d\pi = Idt$, we can solve explicitly the corresponding first-order linear matrix ODE for π^* yielding, for $s \geq t$,

$$\pi_s^* = \pi_t + \int_t^s : \exp \left(\int_u^s \text{Rate}(v) dv \right) : \text{Aim}(u, X_u) du,$$

and this π^* in turn defines W^* like in the proof of Theorem 2.3.1.

Finally, from the above construction it is obvious that both π_t^* and I_t^* are \mathcal{F}_t -adapted and predictable (in fact, they have continuous paths), and the property that π^* is in $L^2([0, T] \times \Omega)$ (i.e., that $\int_0^T \mathbb{E}[|\pi_s^*|^2] ds < \infty$) stems from the observation that $\text{Rate}(u)$ is deterministic and bounded (because of Proposition 2.4.2.1), $\text{Aim}(t, x)$ has linear growth in x uniformly in t (by Proposition 2.4.2.2), and X is a Gaussian process (so it is in $L^2([0, T] \times \Omega)$).

I_t^* is likewise in $L^2([0, T] \times \Omega)$ since, as we saw in Proposition 2.4.1, $I_t^* = C^{-1}(a(t)\pi_t^* + b(t, X_t))$ and we can therefore use the triangular inequality, the just shown fact that π_t^* is in $L^2([0, T] \times \Omega)$, and the same arguments as above that $a(t)$ is bounded (because of Proposition 2.4.2.1), that $b(t, x)$ has linear growth in x uniformly in t (by Proposition 2.4.2.2), and that X is a Gaussian process, to conclude. \square

2.8.4 Proof of Theorem 2.5.1

Solving the SDE for X with Ito's formula and the change of variables $Z_t = e^{At}(X_t + \mu)$ we get that

$$X_{t+\Delta t} = e^{-A\Delta t} X_t + (I - e^{-A\Delta t})\mu + \int_t^{t+\Delta t} e^{-A(\Delta t+t-s)} \sigma dB_s. \quad (2.18)$$

Hence $X_{t+\Delta t}|X_t \sim N(\mu(X_t, \Delta t), \Sigma(\Delta t))$ where $\mu(X_t, \Delta t) = e^{-A\Delta t} X_t + (I - e^{-A\Delta t})\mu$, and $W_{t+\Delta t}|(W_t = w, X_t = x) \sim N(w + \pi'_t(\mu(x, \Delta t) - x) + (w - \pi'_t p)r\Delta t, \pi'_t \Sigma(\Delta t) \pi_t)$.

Bellman's dynamic programming principle then gives that, for $1 \leq l < L$, π_l^* is the solution of

$$H_l(x, w; \mathbf{t}) = \sup_{\pi_l} \mathbb{E}_{t_l, x, w} \left[H_{l+1}(X_{t_{l+1}}, W_{t_{l+1}}; \mathbf{t}) - \left(\frac{\gamma(t_l)}{2} \pi_l' \Sigma(\Delta t_l) \pi_l + \frac{\alpha(t_l)}{2} \pi_l' p p' \pi_l \Delta t_l \right) \right] \quad (2.19)$$

with $H_L(x, w; \mathbf{t}) = w$.

For $l = L - 1$ the expectation is

$$(1+r\Delta t_{L-1})w + \pi_{L-1}'(\mu(x, \Delta t_{L-1}) - x - pr\Delta t_{L-1}) - \left(\frac{\gamma(t_{L-1})}{2} \pi_{L-1}' \Sigma(\Delta t_{L-1}) \pi_{L-1} + \frac{\alpha(t_{L-1})}{2} \pi_{L-1}' p p' \pi_{L-1} \Delta t_{L-1} \right)$$

whose maximum after computing the first order condition is attained at

$$\pi_{L-1}^* = (\gamma(t_{L-1})\Sigma(\Delta t_{L-1}) + \alpha(t_{L-1})pp'\Delta t_{L-1})^{-1} (\mu(x, \Delta t_{L-1}) - x - pr\Delta t_{L-1})$$

as desired, and $H_{L-1}(x, w; \mathbf{t})$ is thus equal to

$$(1+r\Delta t_{L-1})w + \pi_{L-1}^*(\mu(x, \Delta t_{L-1}) - x - pr\Delta t_{L-1}) - \left(\frac{\gamma(t_{L-1})}{2} \pi_{L-1}^* \Sigma(\Delta t_{L-1}) \pi_{L-1}^* + \frac{\alpha(t_{L-1})}{2} \pi_{L-1}^* p p' \pi_{L-1}^* \Delta t_{L-1} \right).$$

Now, if we consider the next optimization in (2.19) for $l = L - 2$, the only term above that depends on w (and hence on the controls) is the first one, so we do not have to worry about the remaining ones for the maximization over π_{L-2} . Thus, for optimization purposes the corresponding expectation will be the same as in the previous iteration but with the wealth W_{L-2} multiplied by $(1 + r\Delta t_{L-1})$, so we obtain that

$$\pi_{L-2}^* = (\gamma(t_{L-1})\Sigma(\Delta t_{L-1}) + \alpha(t_{L-1})pp'\Delta t_{L-1})^{-1} (\mu(x, \Delta t_{L-1}) - x - pr\Delta t_{L-1})(1 + r\Delta t_{L-1})$$

and that $H_{L-2}(x, w; \mathbf{t})$ is given (modulo the terms that we ignored before) by

$$(1+r\Delta t_{L-1})((1+r\Delta t_{L-2})w + \pi_{L-2}^*(\mu(x, \Delta t_{L-2}) - x - pr\Delta t_{L-2})) - \left(\frac{\gamma(t_{L-2})}{2} \pi_{L-2}^* \Sigma(\Delta t_{L-2}) \pi_{L-2}^* + \frac{\alpha(t_{L-2})}{2} \pi_{L-2}^* p p' \pi_{L-2}^* \Delta t_{L-2} \right).$$

It is now clear that iterating this same process for $1 \leq l \leq L - 3$ yields the remaining optimal portfolios described in Theorem 2.5.1.

The final statement in Theorem 2.5.1 is then obvious: we just have to multiply and divide π_l^* by Δt_l and note that $\Delta t^{-1}\Sigma(\Delta t) \rightarrow \sigma\sigma'$ and $\Delta t^{-1}(I - e^{-A\Delta t}) \rightarrow A$ when $\Delta t \rightarrow 0$, whereas $\prod_{i=l+1}^{L-1} (1 + r\Delta t_i) \rightarrow e^{r(t_L - t_l)}$. \square

Chapter 3

Deep learning for statistical arbitrage

3.1 Introduction

¹Statistical arbitrage is one of the pillars of quantitative trading, and has long been used by hedge funds and investment banks. The term statistical arbitrage encompasses a wide variety of investment strategies, which identify and exploit temporal price differences between similar assets using statistical methods. Its simplest form is known as “pairs trading”. Two stocks are selected that are “similar”, usually based on historical co-movement in their price time-series. When the spread between their prices widens, the arbitrageur sells the winner and buys the loser. If their prices move back together, the arbitrageur will profit. While Wall Street has developed a plethora of proprietary tools for sophisticated arbitrage trading, there is still a lack of understanding of how much arbitrage opportunity is actually left in financial markets. In this chapter, we attempt to answer two key questions about statistical arbitrage: What are the important elements of a successful arbitrage strategy and how much realistic arbitrage is in financial markets?

In general, any statistical arbitrage strategy needs to solve the following three fundamental problems. First, given a large universe of assets, what are long-short portfolios of similar assets? Second, given these portfolios, what are time series signals that indicate the presence of temporary price deviations? Third, given these signals, how should an arbitrageur trade them to optimize a trading objective while taking into account possible constraints and market frictions? Each of these three questions poses substantial challenges, that prior work has only partly addressed. First, it is a hard problem to find long-short portfolios for all stocks, as it is a priori unknown what constitutes “similarity”. This problem requires considering all the big data available for a large number of assets and

¹This chapter is based on the preprint [37], which is joint work with Markus Pelger and Greg Zanotti.

times, including not just conventional return data but also exogenous information like asset characteristics. Second, extracting the right signals requires detecting flexibly all the relevant patterns in the noisy, complex, low-sample-size time series of the portfolio prices. Third, optimal trading rules on a multitude of signals and assets are complicated and depend on the trading objective. All of these challenges fundamentally require flexible estimation tools that can deal with many variables. It is a natural idea to use machine learning techniques like deep neural networks to deal with the high dimensionality and complex functional dependencies of the problem. However, our problem is different from the usual prediction task, where machine learning tools excel. We show how to optimally design a machine learning solution to our problem that leverages the economic structure and objective.

In this chapter, we propose a unifying conceptual framework that generalizes common approaches to statistical arbitrage. Statistical arbitrage can be decomposed into three fundamental elements: (1) arbitrage portfolio generation, (2) arbitrage signal extraction and (3) the arbitrage allocation decision given the signal. By decomposing different methods into their arbitrage portfolio, signal and allocation element, we can compare different methods and study which components are the most relevant for successful trading. For each step we develop a novel machine learning implementation, which we compare with conventional methods. As a result, we construct a new deep learning statistical arbitrage approach. Our new approach constructs arbitrage portfolios with a conditional latent factor model, extracts the signals with the currently most successful machine learning time-series method and maps them into a trading allocation with a flexible neural network. These components are integrated and optimized over a global economic objective, which maximizes the risk-adjusted return under constraints. Empirically, our general model outperforms out-of-sample the leading benchmark approaches and provides a clear insight into the structure of statistical arbitrage.

To construct arbitrage portfolios, we introduce the economically motivated asset pricing perspective to create them as residuals relative to asset pricing models. This perspective allows us to take advantage of the recent developments in asset pricing and to also include a large set of firm characteristics in the construction of the arbitrage portfolios. We use fundamental risk factors and conditional and unconditional statistical factors for our asset pricing models. Similarity between assets is captured by similar exposure to those factors. Arbitrage Pricing Theory implies that, with an appropriate model, the corresponding factor portfolios represent the “fair price” of each of the assets. Therefore, the residual portfolios relative to the asset pricing factors capture the temporary deviations from the fair price of each of the assets and should only temporally deviate from their long-term mean. Importantly, the residuals are tradeable portfolios, which are only weakly cross-sectionally correlated and close to orthogonal to firm characteristics and systematic factors. These properties allow us to extract a stationary time-series model for the signal.

To detect time series patterns and signals in the residual portfolios, we introduce a filter perspective and estimate them with a flexible data-driven filter based on convolutional networks combined with transformers. In this way, we do not prescribe a potentially misspecified function to extract the time series structure, for example, by estimating the parameters of a given parametric time-series model, or the coefficients of a decomposition into given basis functions, as in conventional methods. Instead, we directly learn in a data-driven way what the optimal pattern extraction function is for our trading objective. The convolutional transformer is the ideal method for this purpose. Convolutional neural networks are the state-of-the-art AI method for pattern recognition, in particular in computer vision. In our case they identify the local patterns in the data and may be thought as a nonlinear and learnable generalization of conventional kernel-based data filters. Transformer networks are the most successful AI model for time series in natural language processing. In our model, they combine the local patterns to global time-series patterns. Their combination results in a data-driven flexible time-series filter that can essentially extract any complex time-series signal, while providing an interpretable model.

To find the optimal trading allocation, we propose neural networks to map the arbitrage signals into a complex trading allocation. This generalizes conventional parametric rules, for example fixed rules based on thresholds, which are only valid under strong model assumptions and a small signal dimension. Importantly, these components are integrated and optimized over a global economic objective, which maximizes the risk-adjusted return under constraints. This allows our model to learn the optimal signals and allocation for the actual trading objective, which is different from a prediction objective. The trading objective can maximize the Sharpe ratio or expected return subject to a risk penalty, while taking into account constraints important to real investment managers, such as restricting turnover, leverage, or proportion of short trades.

Our comprehensive empirical out-of-sample analysis is based on the daily returns of roughly the 550 largest and most liquid stocks in the U.S. from 1998 to 2016. We estimate the out-of-sample residuals on a rolling window relative to the empirically most important factor models. These are observed fundamental factors, for example the Fama-French 5 factors and price trend factors, locally estimated latent factors based on principal component analysis (PCA) or locally estimated conditional latent factors that include the information in 46 firm-specific characteristics and are based on the Instrumented PCA (IPCA) of [42]. We extract the trading signal with one of the most successful parametric models, based on the mean-reverting Ornstein-Uhlenbeck process, a frequency decomposition of the time-series with a Fourier transformation and our novel convolutional network with transformer. Finally, we compare the trading allocations based on parametric or nonparametric rules estimated with different risk-adjusted trading objectives.

Our empirical main findings are five-fold. First, our deep learning statistical arbitrage model substantially outperforms all benchmark approaches out-of-sample. In fact, our model can achieve an impressive annual Sharpe ratio larger than four. While respecting short-selling constraints we

can obtain annual out-of-sample mean returns of 20%. This performance is four times better than one of the best parametric arbitrage models, and twice as good as an alternative deep learning model without the convolutional transformer filter. These results are particularly impressive as we only trade the largest and most liquid stocks. Hence, our model establishes a new standard for arbitrage trading.

Second, the performance of our deep learning model suggests that there is a substantial amount of short-term arbitrage in financial markets. The profitability of our strategies is orthogonal to market movements and conventional risk factors including momentum and reversal factors and does not constitute a risk-premium. Our strategy performs consistently well over the full time horizon. The model is extremely robust to the choice of tuning parameters, and the period when it is estimated. Importantly, our arbitrage strategy remains profitable in the presence of realistic transaction and holdings costs. Assessing the amount of arbitrage in financial markets with unconditional pricing errors relative to factor models or with parametric statistical arbitrage models, severely underestimates this quantity.

Third, the trading signal extraction is the most challenging and separating element among different arbitrage models. Surprisingly, the choice of asset pricing factors has only a minor effect on the overall performance. Residuals relative to the five Fama-French factors and five locally estimated principal component factors perform very well with out-of-sample Sharpe ratios above 3.2 for our deep learning model. Five conditional IPCA factors increase the out-of-sample Sharpe ratio to 4.2, which suggests that asset characteristics provide additional useful information. Increasing the number of risk factors beyond five has only a marginal effect. Similarly, the other benchmark models are robust to the choice of factor model as long as it contains sufficiently many factors. The distinguishing element is the time-series model to extract the arbitrage signal. The convolutional transformer doubles the performance relative to an identical deep learning model with a pre-specified frequency filter. Importantly, we highlight that time-series modeling requires a time-series machine learning approach, which takes temporal dependency into account. An off-the-shelf nonparametric machine learning method like conventional neural networks, that estimates an arbitrage allocation directly from residuals, performs substantially worse.

Fourth, successful arbitrage trading is based on local asymmetric trend and reversion patterns. Our convolutional transformer framework provides an interpretable representation of the underlying patterns, based on local basic patterns and global “dependency factors”. The building blocks of arbitrage trading are smooth trend and reversion patterns. The arbitrage trading is short-term and the last 30 trading days seem to capture the relevant information. Interestingly, the direction of policies is asymmetric. The model reacts quickly on downturn movements, but more cautiously on uptrends. More specifically, the “dependency factors” which are the most active in downturn movements focus only on the most recent 10 days, while those for upward movements focus on the first 20 days in a 30-day window.

Fifth, time-series-based trading patterns should be extracted from residuals and not directly from returns. For an appropriate factor model, the residuals are only weakly correlated and close to stationary in both, the time and cross-sectional dimension. Hence, it is meaningful to extract a uniform trading pattern, that is based only on the past time-series information, from the residuals. In contrast, stock returns are dominated by a few factors, which severely limits the actual independent time-series information, and are strongly heterogenous due to their variation in firm characteristics. While the level of stock returns is extremely hard to predict, even with flexible machine learning methods, residuals capture relative movements and remove the level component. These properties make residuals analyzable from a purely time-series based perspective and, unlike the existing literature, they allow us to incorporate alternative data into the portfolio construction process. This also highlights a fundamental difference with most of the existing financial machine learning literature: We do not use characteristics to get features for prediction, but rather to generate new data orthogonal to these features.

Related Literature

Our work builds on the classical statistical arbitrage literature, in which the three main problems of portfolio generation, pattern extraction, and allocation decision have traditionally been considered independently. Classical statistical methods of generating arbitrage portfolios have mostly focused on obtaining multiple pairs or small portfolios of assets, using techniques like the distance method of [32], the cointegration approach of [80], or copulas as in [73]. In contrast, more general methods that exploit large panels of stock returns include the use of PCA factor models, as in [2] and its extension in [83], and the maximization of mean-reversion and sparsity statistics as in [19]. We include the model of [83] as the parametric benchmark model in our study as it has one of the best empirical performances among the class of parametric models. Our work contributes to this literature by introducing a general asset pricing perspective to obtain the arbitrage portfolios as residuals. This allows us to take advantage of conditional asset pricing models, that include time-varying firm characteristics in addition to the return time-series, and provide a more disciplined, economically motivated approach. The signal extraction step for these classical statistical arbitrage methods generally assumes parametric time series models for the arbitrage portfolios, whereas the allocations are often decided from the estimated parameters by using stochastic control methods or given threshold rules and one-period optimizations. Some representative papers of the first approach include [41], [65], [14], [59] and [54], whereas the second one is illustrated by [24] and [83]. Both approaches may be considered special cases of our more general framework. On the other hand, [67] and [43] are examples of including machine learning elements within the parametric statistical arbitrage framework, by either solving a stochastic control problem with neural networks or by estimating a time-varying threshold rule with reinforcement learning.

Our work in this chapter is complementary to the emerging literature that uses machine learning

methods for asset pricing. While the asset pricing literature aims to explain the risk premia of assets, our focus is on the residual component which is not explained by the asset pricing models. [15], [10] and [45] estimate the stochastic discount factor (SDF), which explains the risk premia of assets, with deep neural networks, decision trees or elastic net regularization. These papers employ advanced statistical methods to solve a conditional method of moment problem in the presence of many variables. The workhorse models in equity asset pricing are based on linear factor models exemplified by [25, 26]. Recently, new methods have been developed to extract statistical asset pricing factors from large panels with various versions of principal component analysis (PCA). The Risk-Premium PCA in [52, 51] includes a pricing error penalty to detect weak factors that explain the cross-section of returns. The high-frequency PCA in [70] uses high-frequency data to estimate local time-varying latent risk factors and the Instrumented PCA (IPCA) of [42] estimates conditional latent factors by allowing the loadings to be functions of time-varying asset characteristics. [33] generalizes IPCA to allow the loadings to be nonlinear functions of characteristics.

Our work is also related to the growing literature on return prediction with machine learning methods, which has shown the benefits of regularized flexible methods. In their pioneering work, [35] conduct a comparison of machine learning methods for predicting the panel of individual U.S. stock returns based on the asset-specific characteristics and economic conditions in the previous period. In a similar spirit, [6] predicts bond returns and [29] uses different methods for predicting stock returns. This literature is fundamentally estimating the risk premia of assets, while our focus is on understanding and exploiting the temporal deviations thereof. This different goal is reflected in the different methods that are needed. These return predictions estimate a nonparametric model between current returns and large set of covariates from the last period, but do not estimate a time-series model. In contrast, the important challenge that we solve is to extract a complex time-series pattern. A related stream of this literature forecasts returns using past returns, generally followed by some long-short investment policy based on the prediction. For example, [47] uses various machine learning methods for this type of prediction.² However, they use general nonparametric function estimates, which are not specifically designed for time-series data. [58] shows that it is important for machine learning solutions to explicitly account for temporal dependence when they are applied to time-series data. Forecasting returns and building a long-short portfolio based on the prediction is different from statistical arbitrage trading as it combines a risk premium and potential arbitrage component. It is not based on temporary price differences and also in general not orthogonal to common risk factors and market movements. In this chapter, we highlight the challenge of inferring complex time-series information and argue that using returns directly as an input to a time-series machine learning method, is suboptimal as returns are dominated by a few factor time-series and heterogeneous due to cross-sectionally and time-varying characteristics. In contrast, appropriate residuals are locally stationary and hence allow the extraction of a complex time-series pattern.

²Similar studies include [27], [16], [39], and [22].

Our work also overlaps with the literature on using machine learning tools for investment. The SDF estimated by asset pricing models, like in [15] and [10], directly maps into a conditionally mean-variance efficient portfolio and hence an attractive investment opportunity. However, by construction this investment portfolio is not orthogonal but fully exposed to systematic risk, which is exactly the opposite for an arbitrage portfolio. Prediction approaches also imply investment strategies, typically long-short portfolios based on the prediction signal. However, estimating a signal with a prediction objective is not necessarily providing an optimal signal for investment. [10] and [15] illustrate that machine learning models that use a trading objective can result in a substantially more profitable investment than models that estimate a signal with a prediction objective, while using the same information as input and having the same flexibility. This is also confirmed in [18], which uses an investment objective and reinforcement learning to construct machine learning investment portfolios. This chapter contributes to this literature by estimating investment strategies that are orthogonal to systematic risk and are based on a trading objective with constraints.

Finally, our approach is also informed by the recent deep learning for time series literature. The transformer method was first introduced in the groundbreaking paper by [79]. We are the first to bring this idea into the context of statistical arbitrage and adapt it to the economic problem.

Organization of the chapter

The remainder of the chapter is organized as follows. Section 3.2 introduces our general modeling framework and describes the specific classes of models we consider in our empirical study. Section 3.3 describes the models' implementation and contains our main empirical results. Section 3.4 concludes. Finally, Appendix A contains additional empirical results and implementation details.

3.2 Model

The fundamental problem of statistical arbitrage consists of three elements: (1) The identification of similar assets to generate arbitrage portfolios, (2) the extraction of time-series signals for the temporary deviations of the similarity between assets, and (3) a trading policy on the arbitrage portfolios based on the time-series signals. We discuss each element separately and in generality in the next three subsections, and we discuss specific models for the signal and trading components in subsection 3.2.4.

3.2.1 Arbitrage portfolios

We consider a panel of excess returns $R_{n,t}$, that is the return minus risk free rate of stock $n = 1, \dots, N_t$ at time $t = 1, \dots, T$. The number of available assets at time t can be time-varying. The excess return vector of all assets at time t is denoted as $R_t = (R_{1,t} \quad \dots \quad R_{N_t,t})^\top$.

We use a general asset pricing model to identify similar assets. In this context, similarity is defined as the same exposure to systematic risk, which implies that assets with the same risk exposure should have the same fundamental value. We assume that asset returns can be modeled by a conditional factor model:

$$R_{n,t} = \beta_{n,t-1}^\top F_t + \epsilon_{n,t}.$$

The K factors $F \in \mathbb{R}^{T \times K}$ capture the systematic risk, while the risk loadings $\beta_{t-1} \in \mathbb{R}^{N_t \times K}$ can be general functions of the information set at time $t - 1$ and hence can be time-varying. This general formulation includes the empirically most successful factor models. In our empirical analysis we will include fundamental factors, e.g. the Fama-French 5 factor model, latent factors based on the principal components analysis (PCA) of stock returns and conditional latent factors estimated with Instrumented Principal Component Analysis (IPCA).

Without loss of generality, we can treat the factors as excess returns of traded assets. Either the factors are traded, for example a market factor, in which case we include them in the returns R_t . Otherwise, we can generate factor mimicking portfolios by projecting them on the asset space, as for example with latent factors. In any case, we have that

$$F_t = w_{t-1}^F {}^\top R_t$$

for a certain matrix $w_{t-1}^F \in \mathbb{R}^{K \times N_t}$ dependent on the factor model.

We define *arbitrage portfolios* as the residual portfolios $\epsilon_{n,t} = R_{n,t} - \beta_{n,t-1}^\top F_t$. As the factors are traded assets, the arbitrage portfolio is itself a traded portfolio, since

$$\epsilon_t = R_t - \beta_{t-1} w_{t-1}^F {}^\top R_t = \underbrace{\left(I_{N_t} - \beta_{t-1} w_{t-1}^F {}^\top \right)}_{\Phi_{t-1}} R_t = \Phi_{t-1} R_t \quad (3.1)$$

for the matrix $\Phi_{t-1} = I_{N_t} - \beta_{t-1} w_{t-1}^F {}^\top \in \mathbb{R}^{N_t \times N_t}$.

The arbitrage portfolios are projections on the return space that annihilate systematic asset risk. For an appropriate asset pricing model, the residual portfolios should not earn a risk premium. This is the fundamental assumption behind any arbitrage argument. As deviations from a mean of zero have to be temporary, arbitrage trading bets on the mean revision of the residuals. In particular, for an appropriate factor model the residuals should have the following properties:

1. The unconditional mean of the arbitrage portfolios is zero: $\mathbb{E}[\epsilon_{n,t}] = 0$.
2. The arbitrage portfolios are only weakly cross-sectionally dependent.

We denote by \mathcal{F}_t the filtration generated by the returns R_t , which include the factors, and the information set that captures the risk exposure β_t , which is typically based on asset specific characteristics

or past returns.

3.2.2 Arbitrage signal

The *arbitrage signal* is extracted from the time-series of the arbitrage portfolios. These time-series signals are the input for a trading policy. An example for an arbitrage signal would be the parameters of a parametric model for mean reversion that is estimated for each arbitrage portfolio. The trading strategy for each arbitrage portfolio would depend on its speed of mean reversion and its deviation from the long run mean. More generally, the arbitrage signal corresponds to the estimation of a time-series model, which can be parametric or non-parametric. Conceptually, time-series models are multivariate functional mappings between sequences which take into account the temporal order of the elements and potentially complex dependencies between the elements of the input sequence.

We apply the signal extraction to the time-series of the last L lagged residuals, which we denote in vector notation as

$$\epsilon_{n,t-1}^L := \begin{pmatrix} \epsilon_{n,t-L} & \cdots & \epsilon_{n,t-1} \end{pmatrix}.$$

The arbitrage signal function is a mapping $\boldsymbol{\theta} \in \Theta$ from \mathbb{R}^L to \mathbb{R}^p , where Θ defines an appropriate function space:

$$\boldsymbol{\theta}(\cdot) : \epsilon_{n,t-1}^L \mapsto \theta_{n,t-1}.$$

The signals $\theta_{n,t-1} \in \mathbb{R}^p$ for the arbitrage portfolio n at time t only depend on the time-series of lagged returns $\epsilon_{n,t-1}^L$. Note that the dimensionality of the signal can be the same as for the input sequence. Formally, the function $\boldsymbol{\theta}$ is a mapping from the filtration $\mathcal{F}_{n,t-1}^{\epsilon,L}$ generated by $\epsilon_{n,t-1}^L$ into the filtration $\mathcal{F}_{n,t-1}^\theta$ generated by $\theta_{n,t-1}$ and $\mathcal{F}_{n,t-1}^\theta \subseteq \mathcal{F}_{n,t-1}^{\epsilon,L}$. We use the notation of evaluating functions elementwise, that is $\boldsymbol{\theta}(\epsilon_{t-1}^L) = (\theta_{1,t-1} \ \cdots \ \theta_{N_t,t-1}) = \theta_{t-1} \in \mathbb{R}^{N_t}$ with $\epsilon_{t-1}^L = (\epsilon_{1,t-1} \ \cdots \ \epsilon_{N_t,t-1})$.

The arbitrage signal $\theta_{n,t-1}$ is a sufficient statistic for the trading policy; that is, all relevant information for trading decisions is summarized in it. This also implies that two arbitrage portfolios with the same signal get the same weight in the trading strategy. More formally, this means that the arbitrage signal defines equivalence classes for the arbitrage portfolios. The most relevant signals summarize reversal patterns and their direction with a small number of parameters. A potential trading policy could be to hold long positions in residuals with a predicted upward movement and go short in residuals that are in a downward cycle.

This problem formulation makes two implicit assumptions. First, the residual time-series follow a stationary distribution conditioned on its lagged returns. This is a very general framework that includes the most important models for financial time-series. Second, the first L lagged returns are

a sufficient statistic to obtain the arbitrage signal $\theta_{n,t-1}$. This reflects the motivation that arbitrage is a temporary deviation of the fair price. The lookback window can be chosen to be arbitrarily large, but in practice it is limited by the availability of lagged returns.

3.2.3 Arbitrage trading

The trading policy assigns an investment weight to each arbitrage portfolio based on its signal. The allocation weight is the solution to an optimization problem, which models a general risk-return tradeoff and can also include trading frictions and constraints. An important case are mean-variance efficient portfolios with transaction costs and short sale constraints.

Formally, an arbitrage allocation is a mapping from \mathbb{R}^p to \mathbb{R} in a function class \mathbf{W} , that assigns a weight $w_{n,t-1}^\epsilon$ for the arbitrage portfolio $\epsilon_{n,t-1}$ in the investment strategy using only the arbitrage signal $\theta_{n,t}$:

$$\mathbf{w}^\epsilon : \theta_{n,t-1} \mapsto w_{n,t-1}^\epsilon.$$

Given a concave utility function $U(\cdot)$, we find the signal and the allocation functions as the solutions to

$$\max_{\mathbf{w}^\epsilon \in \mathbf{W}, \boldsymbol{\theta} \in \Theta} \mathbb{E}_{t-1} [U(w_{t-1}^R R_t - \text{cost}(w_{t-1}^R, w_{t-2}^R))] \quad (3.2)$$

$$\text{s.t.} \quad w_{t-1}^R = \frac{w_{t-1}^\epsilon \top \Phi_{t-1}}{\|w_{t-1}^\epsilon \top \Phi_{t-1}\|_1} \quad \text{and} \quad w_{t-1}^\epsilon = \mathbf{w}^\epsilon(\boldsymbol{\theta}(\epsilon_{t-1}^L)), \quad (3.3)$$

where the trading cost function $\text{cost}(\cdot, \cdot)$ can capture the transaction costs from frequent rebalancing and the higher costs of short selling compared to long positions. The stock weights w_{t-1}^R are normalized to add up to one in absolute value, which implicitly imposes a leverage constraint. The conditional expectation uses the general filtration \mathcal{F}_{t-1} .

This is a combined optimization problem, which simultaneously solves for the optimal allocation function and arbitrage signal function. As the weight is a composition of the two functions, i.e. $w_{t-1}^\epsilon = \mathbf{w}^\epsilon(\boldsymbol{\theta}(\epsilon_{t-1}^L))$, the decomposition into a signal and allocation function is in general not uniquely identified. This means there can be multiple representations of $\boldsymbol{\theta}$ and \mathbf{w}^ϵ , that will result in the same trading policy. We use a decomposition that allows us to compare the problem to classical arbitrage approaches, for which this separation is uniquely identified. The key feature of the signal function $\boldsymbol{\theta}$ is that it models a time-series, that means, it is a mapping that explicitly models the temporal order and the dependency between the elements of ϵ_{t-1}^L . The allocation function \mathbf{w}^ϵ can be a complex nonlinear function, but does not explicitly model time-series behavior. This means that \mathbf{w}^ϵ is implicitly limited in the dependency patterns of its input elements that it can capture.

In our empirical analysis, we will consider arbitrage trading that maximizes the Sharpe ratio or

achieves the highest average return for a given level of variance risk. More specifically we will solve for

$$\max_{\mathbf{w}^\epsilon \in \mathbf{W}, \boldsymbol{\theta} \in \Theta} \frac{\mathbb{E}[w_{t-1}^R R_t]}{\sqrt{\text{Var}(w_{t-1}^R R_t)}} \quad \text{or} \quad \max_{\mathbf{w}^\epsilon \in \mathbf{W}, \boldsymbol{\theta} \in \Theta} \mathbb{E}[w_{t-1}^R R_t] - \gamma \text{Var}(w_{t-1}^R R_t) \quad (3.4)$$

$$\text{s.t.} \quad w_{t-1}^R = \frac{w_{t-1}^{\epsilon \top} \Phi_{t-1}}{\|w_{t-1}^{\epsilon \top} \Phi_{t-1}\|_1} \quad \text{and} \quad w_{t-1}^{\epsilon \top} = \mathbf{w}^\epsilon(\boldsymbol{\theta}(\epsilon_{t-1}^L)). \quad (3.5)$$

for some risk aversion parameter γ . We will consider this formulation with and without trading costs.³

Many relevant models estimate the signal and allocation function separately. The arbitrage signals can be estimated as the parameters of a parametric time-series model, the serial moments for a given stationary distribution or a time-series filter. In these cases, the signal estimation solves a separate optimization problem as part of the estimation. Given the signals, the allocation function is the solution of

$$\max_{\mathbf{w}^\epsilon \in \mathbf{W}} \mathbb{E}_{t-1} [U(w_{t-1}^R R_t)] \quad \text{s.t.} \quad w_{t-1}^R = \frac{w^\epsilon(\theta_{t-1})^\top \Phi_{t-1}}{\|w^\epsilon(\theta_{t-1})^\top \Phi_{t-1}\|_1}. \quad (3.6)$$

We provide an extensive study of the importance of the different elements in statistical arbitrage trading. We find that the most important driver for profitable portfolios is the arbitrage signal function; that is, a good model to extract time-series behavior and to time the predictable mean reversion patterns is essential. The arbitrage portfolios of asset pricing models that are sufficiently rich result in roughly the same performance. Once an informative signal is extracted, parametric and nonparametric allocation functions can take advantage of it. We find that the key element is to consider a sufficiently general class of functions Θ for the arbitrage signal and to estimate the signal that is the most relevant for trading. In other words, the largest gains in statistical arbitrage come from flexible time-series signals $\boldsymbol{\theta}$ and a joint optimization problem.

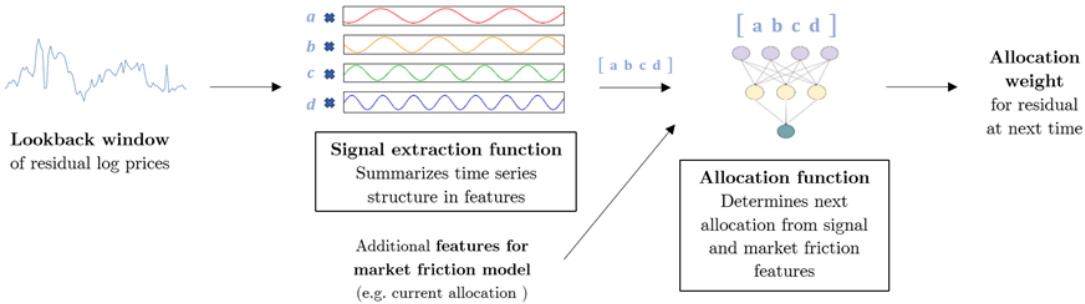
3.2.4 Models for Arbitrage Signal and Allocation Functions

In this section we introduce different functional models for the signal and allocation functions. They range from the most restrictive assumptions for simple parametric models to the most flexible model, which is our deep neural network architecture. The general problem is the estimation of a signal and allocation function given the residual time-series. Here, we take the residual returns as given, i.e. we have selected an asset pricing model. In order to illustrate the key elements of the allocation functions, we consider trading the residuals directly. Projecting the residuals back into the original return space is identical for the different methods and discussed in the empirical part.

³While the conventional mean-variance optimization applies a penalty to the variance, we found that it is numerically beneficial to use the standard deviation instead. Both formulations describe the same fundamental trade-off.

The conceptual steps are illustrated in Figure 3.1.

Figure 3.1: Conceptual Arbitrage Model



This figure illustrates the structure of our model for trading residuals. The model takes as input a lookback window of the last L cumulative returns or log prices of a residual portfolio at a given time and outputs the predicted optimal allocation weight for that residual for the next time. The model is composed of a signal extraction function and an allocation function, whose purposes are explained in the figure.

The input to the signal extraction functions are the last L cumulative residuals. We simplify the notation by dropping the time index $t - 1$ and the asset index n and define the generic input vector

$$x := \text{Int}(\epsilon_{n,t-1}^L) = (\epsilon_{n,t-L} \quad \sum_{l=1}^2 \epsilon_{n,t-L-1+l} \quad \cdots \quad \sum_{l=1}^L \epsilon_{n,t-L-1+l}).$$

Here the operation $\text{Int}(\cdot)$ simply integrates a discrete time-series. We can view the cumulative residuals as the residual ‘‘price’’ process. We discuss three different classes of models for the signal function θ that vary in the degree of flexibility of the type of patterns that they can capture. Similarly, we consider different classes of models for the allocation function w^ϵ .

Parametric Models

Our first benchmark method is a parametric model and corresponds to classical mean-reversion trading. In this framework, the cumulative residuals x are assumed to be the discrete realizations of a continuous time model:

$$x = (X_1 \quad \cdots \quad X_L).$$

Following [2] and [83] we model X_t as an Ornstein-Uhlenbeck (OU) process

$$dX_t = \kappa(\mu - X_t)dt + \sigma dB_t$$

for a Brownian motion B_t . These are the standard models for mean-reversion trading and [2] among others have shown their good empirical performance.

The parameters of this model are estimated from the moments of the discretized time-series, as

described in detail along with the other implementation details in Appendix A.2.2. The parameters for each residual process, the last cumulative sum and a goodness of fit measure form the signals for the Ornstein-Uhlenbeck model:

$$\theta^{\text{OU}} = \begin{pmatrix} \hat{\kappa} & \hat{\mu} & \hat{\sigma} & X_L & R^2 \end{pmatrix}.$$

Following [83] we also include the goodness of fit parameter R^2 as part of the signal. R^2 has the conventional definition of the ratio of squared values explained by the model normalized by total squared values. If the R^2 value is too low, the predictions of the model seem to be unreliable, which can be taken into account in a trading policy. Hence, for each cumulative residual vector $\epsilon_{n,t-1}^L$ we obtain the signal

$$\theta_{n,t-1}^{\text{OU}} = \begin{pmatrix} \hat{\kappa}_{n,t-1} & \hat{\mu}_{n,t-1} & \hat{\sigma}_{n,t-1} & \sum_{l=1}^L \epsilon_{n,t-1+l} & R_{n,t-1}^2 \end{pmatrix}.$$

[2] and [83] advocate a classical mean-reversion thresholding rule, which in our framework corresponds to the following allocation function:

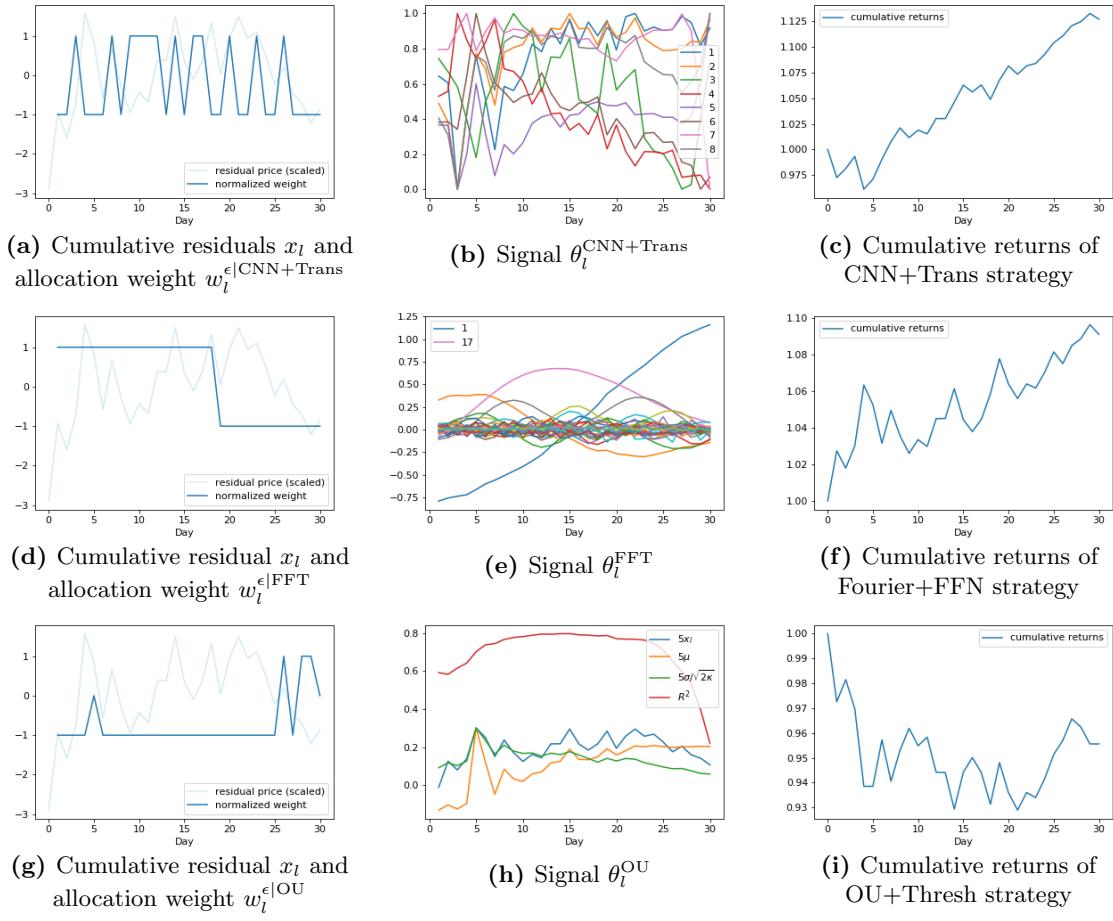
$$\mathbf{w}^{\epsilon|\text{OU}}(\theta^{\text{OU}}) = \begin{cases} -1 & \text{if } \frac{X_L - \mu}{\sigma/\sqrt{2\kappa}} > c_{\text{thresh}} \text{ and } R^2 > c_{\text{crit}} \\ 1 & \text{if } \frac{X_L - \mu}{\sigma/\sqrt{2\kappa}} < -c_{\text{thresh}} \text{ and } R^2 > c_{\text{crit}} \\ 0 & \text{otherwise} \end{cases}$$

The threshold parameters c_{thresh} and c_{crit} are tuning parameters. The strategy suggests to buy or sell residuals based on the ratio $\frac{X_L - \mu}{\sigma/\sqrt{2\kappa}}$. If this ratio exceeds a threshold, it is likely that the process reverts back to its long term mean, which starts the trading. If the R^2 value is too low, the predictions of the model seem to be unreliable, which stops the trading. This will be our parametric benchmark model. It has a parametric model for both the signal and allocation function.

Figure 3.2 illustrates this model with an empirical example. In this figure we show the allocation weights and signals of the Ornstein-Uhlenbeck with threshold model as well as the more flexible models that we are going to discuss next. The models are estimated on the empirical data, and the residual is a representative empirical example. In more detail, we consider the residuals from five IPCA factors and estimate the benchmark models as explained in the empirical Section 3.3.10. The left subplots display the cumulative residual process along with the out-of-sample allocation weights w_l^ϵ that each model assigns to this specific residual. The evaluation of this illustrative example is a simplification of the general model that we use in our empirical main analysis. In this example, we consider trading only this specific residual and hence normalize the weights to $\{-1, 0, 1\}$. In our empirical analysis we trade all residuals and map them back into the original stock returns. The middle column shows the time-series of estimated out-of-sample signals for each model, by applying the θ_l arbitrage signal function to the previous $L = 30$ cumulative returns of the residual. The right

column displays the out-of-sample cumulative returns of trading this particular residual based on the corresponding allocation weights.

Figure 3.2: Illustrative Example of Allocation Weights and Signals for Different Methods



These plots are an illustrative example of the allocation weights and signals of the Ornstein-Uhlenbeck with Threshold (OU+Thres), Fast Fourier Transform (FFT) with Feedforward Neural Network (FFN), and Convolutional Neural Network (CNN) with Transformer models for a specific cumulative residual. The models are estimated on the empirical data, and the residual is a representative empirical example. In more detail, we consider the residuals from five IPCA factors and estimate the benchmark models as explained in Section 3.3.10. The left subplots display the cumulative residual process along with the out-of-sample allocation weights $w_l^{\epsilon| \cdot}$ that each model assigns to this specific residual. In this example, we consider trading only this specific residual and hence normalize the weights to $\{-1, 0, 1\}$. The middle column plots show the time-series of estimated out-of-sample signals for each model, by applying the θ_l^{\cdot} arbitrage signal function to the previous L cumulative returns of the residual. The right column plots display the out-of-sample cumulative returns of trading this particular residual based on the corresponding allocation weights. We use a rolling lookback window of $L = 30$ days to estimate the signal and allocation, which we evaluate for the out-of-sample on the next 30 days. The plots only show the out-of-sample period. The evaluation of this illustrative example is a simplification of the general model that we use in our empirical main analysis, where we trade all residuals and map them back into the original stock returns.

The last row in Figure 3.2 shows the results for the OU+Threshold model. The cumulative return

of trading this residual is negative, suggesting that the parametric model fails. The residual time-series with the corresponding allocation weights in subplot (g) explain why. The trading allocation does not assign a positive weight during the uptrend and wrongly assigns a constant negative weight, when the residual price process follows a mean-reversion pattern with positive and negative returns. A parametric model can break down if it is misspecified. This is not only the case for trend patterns, but also if there are multiple mean reversion patterns of different frequencies. Subplot (h) shows the signal.⁴ We see that changes in the allocation function are related to sharp changes in at least one of the signals, but overall, the signal does not seem to capture the complex price patterns of the residual.

A natural generalization is to allow for a more flexible allocation function given the same time-series signals. We will consider for all our models also a general feedforward neural network (FFN) to map the signal into an allocation weight. FFNs are nonparametric estimators that can capture very general functional relationships.⁵ Hence, we also consider the additional model that restricts the signal function, but allows for a flexible allocation function:

$$\mathbf{w}^{\epsilon|\text{OU-FFN}}(\theta^{\text{OU}}) = \mathbf{g}^{\text{FFN}}(\theta^{\text{OU}}).$$

We will show empirically that the gains of a flexible allocation function are minor relative to the very simple parametric model.

Pre-Specified Filters

As a generalization of the restrictive parametric model of the last subsection, we consider more general time-series models. Many relevant time-series models can be formulated as filtering problems. Filters are transformations of time-series that provide an alternative representation of the original time-series which emphasizes certain dynamic patterns.

A time-invariant linear filter can be formulated as

$$\theta_l = \sum_{j=1}^L W_j^{\text{filter}} x_j,$$

which is a linear mapping from \mathbb{R}^L into \mathbb{R}^L with the matrix $W^{\text{filter}} \in \mathbb{R}^{L \times L}$. The estimation of causal ARMA processes is an example for such filters. A spectral decomposition based on a frequency filter is the most relevant filter for our problem of finding mean reversion patterns.

A Fast Fourier Transform (FFT) provides a frequency decomposition of the original time-series

⁴For better readability we have scaled the parameters of the OU process by a factor of five, but this still represents the same model as the scaling cancels out in the allocation function. As a minor modification, we use the ratio $\sigma/\sqrt{2\kappa}$ as a signal instead of two individual parameters, as the conventional regression estimator of the OU process directly provides the ratio, but requires additional moments for the individual parameters. However, this results in an equivalent presentation of the model as only the ratio enters the allocation function.

⁵Appendix A.2.1 provides the details for estimating a FFN as a functional mapping $\mathbf{g}^{\text{FFN}} : \mathbb{R}^p \rightarrow \mathbb{R}$.

and separates the movements into mean reverting processes of different frequencies. FFT applies the filter $W_j^{\text{FFT}} = e^{\frac{2\pi i}{L} j}$ in the complex plane, but for real-valued time-series it is equivalent to fitting the following model:

$$x_l = a_0 + \sum_{j=1}^{L/2-1} \left(a_j \cdot \cos\left(\frac{2\pi j}{L} l\right) + b_j \cdot \sin\left(\frac{2\pi j}{L} l\right) \right) + a_{L/2} \cos(\pi l).$$

The FFT representation is given by coefficients of the trigonometric representation

$$\theta^{\text{FFT}} = \begin{pmatrix} a_0 & \dots & a_{L/2} & b_1 & \dots & b_{L/2-1} \end{pmatrix} \in \mathbb{R}^L.$$

The coefficients a_l and b_l can be interpreted as “loadings” or exposure to long or short-term reversal patterns. Note that the FFT is an invertible transformation. Hence, it simply represents the original time-series in a different form without losing any information. It is based on the insight that not the magnitude of the original data but the relative relationship in a time-series matters.

We use a flexible feedforward neural network for the allocation function

$$\mathbf{w}^{\epsilon|\text{FFT}}(\theta^{\text{FFT}}) = \mathbf{g}^{\text{FFN}}(\theta^{\text{FFT}}).$$

The usual intuition behind filtering is to use the frequency representation to cut off frequencies that have low coefficients and therefore remove noise in the representation. The FFN is essentially implementing this filtering step of removing less important frequencies.

We illustrate the model within our running example in Figure 3.2. The middle row shows the results for the FFT+FFN model. The cumulative residual in subplot (d) seems to be a combination of low and high-frequency movements with an initial trend component. The signal in subplot (e) suggests that the FFT filter seems to capture the low frequency reversal pattern. However, it misses the high-frequency components as indicated by the simplistic allocation function. The trading strategy takes a long position for the first half and a short position for the second part. While this simple allocation results in a positive cumulative return, in this example it neglects the more complex local reversal patterns.

While the FFT framework is an improvement over the simple OU model as it can deal with multiple combined mean-reversion patterns of different frequencies, it fails if the data follows a pattern that cannot be well approximated by a small number of the prespecified basis functions.

For completeness, our empirical analysis will also report the case of a trivial filter, which simply takes the residuals as signals, and combines them with a general allocation function:

$$\begin{aligned} \boldsymbol{\theta}^{\text{ident}}(x) &= x = \theta^{\text{ident}} \\ \mathbf{w}^{\epsilon|\text{FFN}}(\theta^{\text{ident}}) &= \mathbf{g}^{\text{FFN}}(x). \end{aligned}$$

This is a good example to emphasize the importance of a time-series model. While FFNs are flexible in learning low dimensional functional relationships, they are limited in learning a complex dependency model. For example, the FFN architecture we consider is not sufficiently flexible to learn the FFT transformation and hence has a worse performance on the original time-series compared to frequency-transformed time-series. While [81] have shown that FFNs are “universal approximators” of low-dimensional functional relationships, they also show that FFN can suffer from a curse of dimensionality when capturing complex dependencies between the input. Although the time domain and frequency domain representations of the input are equivalent under the Fourier transform, clearly the time-series model implied by the frequency domain representation allows for a more effective learning of an arbitrage policy. However, the choice of the pre-specified filter limits the time-series patterns that can be exploited. The solution is our data driven filter presented in the next section.

Convolutional Neural Network with Transformer

Our benchmark machine learning model is a Convolutional Neural Network (CNN) combined with a Transformer. It uses the most advanced machine learning tools for time series tailored to our problem. Convolutional networks are the most successful networks for computer vision, i.e. for pattern detection. Transformers have rapidly become the model of choice for sequence modeling such as Natural Language Processing (NLP) problems, replacing older recurrent neural network models such as the Long Short-Term Memory (LSTM) network.

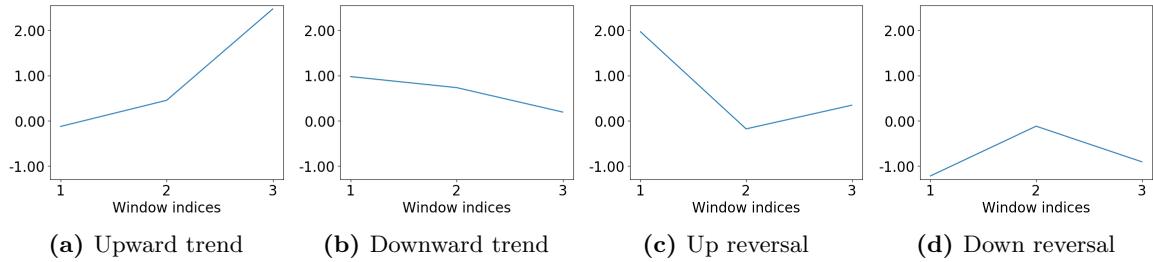
The CNN and transformer framework has two key elements: (1) Local filters and (2) the temporal combination of these local filters. The CNN can be interpreted as a set of data driven flexible local filters. A transformer can be viewed as a data driven flexible time-series model to capture complex dependencies between local patterns. We use the CNN+Transformer to generate the time-series signal. The allocation function is then modeled as a flexible data driven allocation with an FFN.

The CNN estimates D local filters of size D_{size} :

$$y_l^{(0)} = \sum_{m=1}^{D_{\text{size}}} W_m^{(0)} x_{l-m+1}$$

for a matrix $W^{(0)} \in \mathbb{R}^{D_{\text{size}} \times D}$. The local filters are a mapping from $x \in \mathbb{R}^L$ to $y^{(0)} \in \mathbb{R}^{L \times D}$ given by the convolution $y^{(0)} = W^{(0)} * x$. Figure 3.3 shows examples of these local filters for $D_{\text{size}} = 3$. The values of $y^{(0)}$ can be interpreted as the “loadings” or exposure to local basis patterns. For example, if x represents a global upward trend, its filtered representation should have mainly large values for the local upward trend filter.

The convolutional mapping can be repeated in multiple layers to obtain a multi-layer CNN. First, the output of the first layer of the CNN is transformed nonlinearly by applying the $\text{ReLU}(\cdot)$

Figure 3.3: Examples of Local Filters

These figures show the most important local filters estimated for the benchmark model in our empirical analysis. These are projections of our higher dimensional nonlinear filter from a 2-layer CNN into two-dimensional linear filters.

function:

$$x_{l,d}^{(1)} = \text{ReLU} \left(y_{l,d}^{(0)} \right) := \max(y_{l,d}^{(0)}, 0).$$

The second layer is given by a higher dimensional filtering projection:

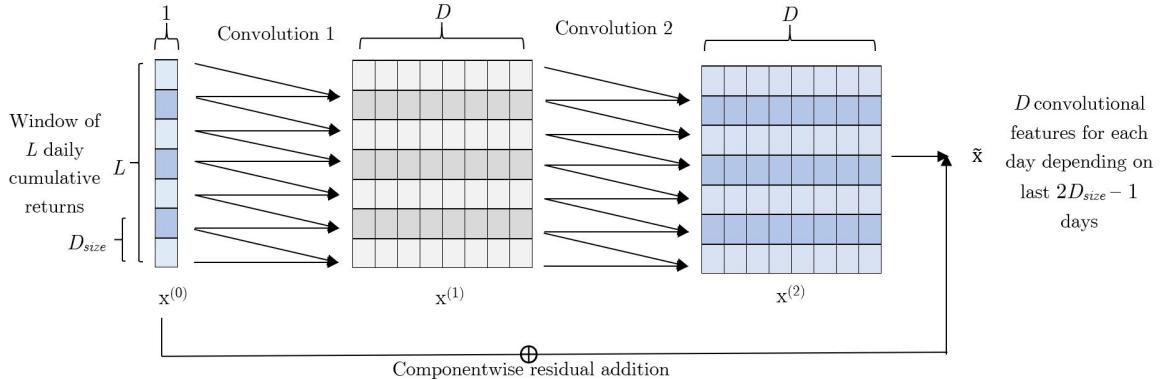
$$\begin{aligned} y_{l,d}^{(2)} &= \sum_{m=1}^{D_{\text{size}}} \sum_{j=1}^D W_{d,j,m}^{(1)} x_{l-m+1,j}^{(1)}, \\ x_{l,d}^{(2)} &= \text{ReLU} \left(y_{l,d}^{(1)} \right). \end{aligned}$$

The final output of the CNN is $\tilde{x} \in \mathbb{R}^{L \times D}$. Our benchmark model is a 2-layered convolutional neural network. The number of layers is a hyperparameter selected on the validation data. Figure 3.4 illustrates the structure of the 2-layer CNN. While this description captures all the conceptual elements, the actual implementation includes additional details, such as bias terms, instance normalization and residual connection to improve the implementation as explained in Appendix A.2.3.

For a 1-layer CNN without the final nonlinear transformation, i.e. for a simple local linear filter, the patterns can be visualized by the vectors $W_m^{(0)}$. In our case of a 2-layer CNN the local filter can capture more complex patterns as it applies a 3-dimensional weighting scheme in the array $W^{(1)}$ and nonlinear transformations. In order to visualize the type of patterns, we project the local filter into a simple local linear filter. We want to find the basic patterns that activate only one of the D filters, but none of the others, i.e. we are looking for an orthogonal representation of the projection.⁶

⁶A local filter can be formalized as a mapping from the local D_{size} points of a sequence to the activation of the D filters: $\phi : \mathbb{R}^{D_{\text{size}}} \rightarrow \mathbb{R}^D$. Denote by $e_d \in \mathbb{R}^D$ a vector that is 0 everywhere except for the value 1 at position d , i.e. $e_d = (0 \ \dots \ 1 \ \dots 0)$. Fundamentally, we want to invert the local filter to obtain $\phi^{-1}(e_d)$ to find the local sequences that only activates filter d . In general, the inverse is a set and not unique. Our example basic patterns in Figure 3.3 solve

$$\operatorname{argmin}_{x_{\text{loc},d} \in \mathbb{R}^{D_{\text{size}}}} \|\phi(x_{\text{loc},d}) - e_d\|_2 \quad \text{for } d = 1, \dots, D.$$

Figure 3.4: Convolutional Network Architecture

This figure shows the structure of our convolutional network. The network takes as input a window of L consecutive daily cumulative returns, and outputs D features for each block of D_{size} days. Each of the features is a nonlinear function of the observations in the block, and captures a common pattern.

The example plots for local filters in Figure 3.3 are projections of our higher dimensional nonlinear filter into two-dimensional linear filters. The examples show some of the most important local filters for our empirical benchmark model. While these projections are of course not complete representations of the nonlinear filters of the CNN, they provide an intuition for the type of patterns which are activated by specific filters. Our 2-layer CNN network has a local window size of $D_{size} = 2$, but because of the 2-layer structure it captures information from two neighboring points. Hence, the projection on a one-dimensional linear filter has a local window size of three as depicted in Figure 3.3.

The output of the CNN $\tilde{x} \in \mathbb{R}^{L \times D}$ is used as an input to the transformer. The CNN projection provides a more informative representation of the dynamics than the original time-series as it captures the relative local dependencies between data points. However, by construction the CNN is only a local representation, and we need the transformer network to detect the global patterns. A transformer network is a model of temporal dependencies between local filters. Given the local structure \tilde{x} the transformer estimates the temporal interactions between the L different blocks by computing a ‘‘global pattern projection’’:

Assume there are H different global patterns. The transformer will calculate projections on these H patterns with the ‘‘attention weights’’. We first introduce a simplified linear projection model before extending it to the actual transformer. For each of the $i = 1, \dots, H$ patterns we have projections defined by $\alpha_i \in \mathbb{R}^{L \times L}$:

$$h_{i,l}^{\text{simple}} = \sum_{j=1}^L \alpha_{i,l,j} \tilde{x}_j \quad \text{for } l = 1, \dots, L \text{ and } i = 1, \dots, H.$$

The “attention function” $\alpha_i(.,.) \in [0, 1]$ captures dependencies between the local patterns:

$$\alpha_{i,l,j} = \alpha_i(\tilde{x}_l, \tilde{x}_j) \quad \text{for } i = 1, \dots, H.$$

Each projection $h_{i,l}^{\text{simple}}$ is called an “attention head”. These attention heads $h_{i,l}^{\text{simple}}$ could be interpreted as “loadings” or “exposure” for a specific “pattern factor” α_i . For example, a global upward trend can be captured by an attention function that puts weight on subsequent local upward trends. Another example would be sinusoidal mean reversion patterns which would put weights on alternating “curved” local basis patterns. The projection on these weights captures how much a specific time-series \tilde{x} is exposed to this global pattern. Hence, $h_{i,l}^{\text{simple}}$ measures the exposure to the global pattern i at time l of the time-series \tilde{x} . Each attention head can focus on a specific global pattern, which we then combine to obtain our signal.

The fundamental challenge is to learn attention functions that can model complex dependencies. The crucial innovation in transformers is their modeling of the attention functions α_i and attention heads h_i . In order to deal with the high dimensionality of the problem, transformers consider lower dimensional projections of \tilde{x} into $\mathbb{R}^{D/H}$ and use the lower-dimensional scaled dot product attention mechanism for α_i as explained in Appendix A.2.3. More specifically, each attention head $h_i \in \mathbb{R}^{L \times D/H}$ is based on⁷ the projected input $\tilde{x}W_i^V$ with $W_i^V \in \mathbb{R}^{D \times D/H}$ and $\alpha_i \in \mathbb{R}^{L \times L}$:

$$h_i = \alpha_i \tilde{x} W_i^V \quad \text{for } i = 1, \dots, H.$$

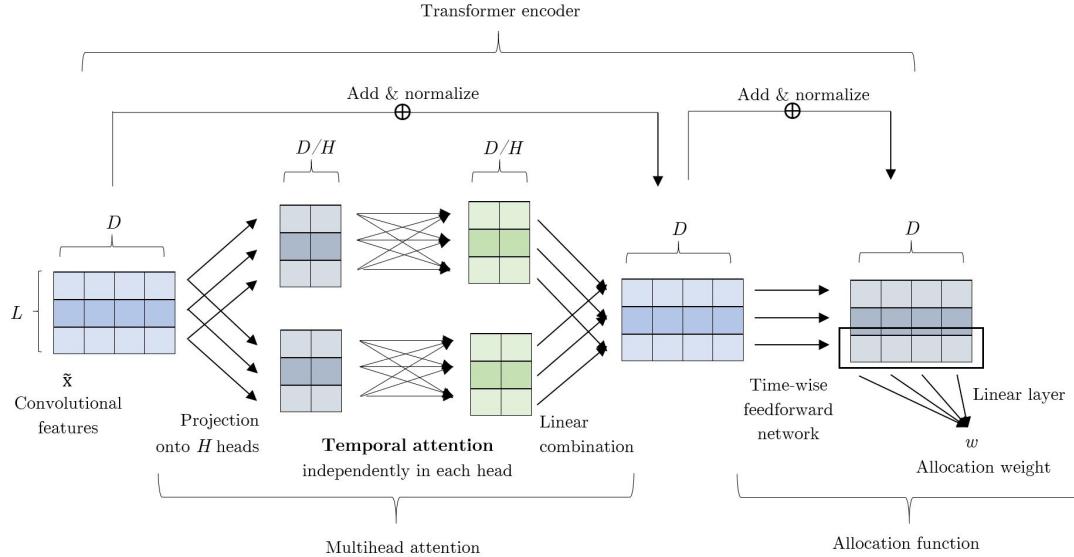
The projection on all global basis patterns $h \in \mathbb{R}^{L \times D/H}$ is given by a weighted linear combination of the different attention heads

$$h^{\text{proj}} = \begin{pmatrix} h_1 & \dots & h_H \end{pmatrix} W^O$$

with $W^O \in \mathbb{R}^{D \times D}$. This final projection can, for example, model a combination of a global trend and mean reversion patterns. In conclusion, h^{proj} represents the time-series in terms of the H global patterns. This is analogous to a Fourier filter, but without pre-specifying the global patterns a priori. All parts of the CNN+Transformer network, i.e. the local patterns, the attention functions and the projections on global patterns, are estimated from the data.

The trading signal $\theta^{\text{CNN+Trans}}$ equals the global pattern projection for the final cumulative return

⁷The actual implementation also includes bias terms which we neglect here for simplicity. The Appendix provides the implementation details.

Figure 3.5: Transformer Network Architecture

This figure shows the structure of our transformer network. The model takes as input the matrix $\tilde{x} \in \mathbb{R}^{L \times D}$ that we obtain as output of the convolutional network depicted in Figure 3.4, which contains D features for each of the L blocks of the original time series. These features are projected onto H attention heads, which independently quantify the temporal relation between the blocks and aggregate them into hidden states. These hidden states are finally combined by a feedforward network and a linear layer to predict the optimal allocation for the residual in the next day.

projection⁸ h_L^{proj} :

$$\theta^{\text{CNN+Trans}} = h_L^{\text{proj}} \in \mathbb{R}^H,$$

which is then used as input to a time-wise feedforward network allocation function

$$\boldsymbol{w}^{\epsilon|\text{CNN+Trans}}(\theta^{\text{CNN+Trans}}) = \boldsymbol{g}^{\text{FFN}}(\theta^{\text{CNN+Trans}}).$$

The separation between signal and allocation is not uniquely identified as we use a joint optimization problem. We have chosen a separation that maps naturally into the classical examples considered in the previous subsections. Figure 3.5 illustrates the transformer network architecture. We have presented a 1-layer transformer network, which is our benchmark model. The transformed data can be used as input in more iterations of the transformer to obtain a multi-layer transformer.

We illustrate the CNN+Transformer model in the first row of Figure 3.2 for an empirical residual example. First, it is apparent that the cumulative returns of the strategy in subplot (c) outperforms

⁸In principle, we can use the complete matrix $h \in \mathbb{R}^{L \times D}$ as the signal. However, conceptually the global pattern at the end of the time period should be the most relevant for the next realization of the process. We have also implemented a transformer that uses the full matrix, with similar results and the variable importance rankings suggest that only h_L is selected in the allocation function.

the previous two models. This is because the allocation weights in subplot (a) capture not only the low frequency reversal patterns, but also the high-frequency cycles and trend components. This also implies that the allocation weights change more frequently to capture the higher frequency components. This more sophisticated allocation function requires a more complex signal as illustrated in subplot (b). Each change in the allocation can be traced back to changes in at least one of the signals. While the signals themselves are hard to interpret, we will leverage the transformer structure to extract interpretable “global dependency factors” in our main analysis. Figure A.2 in the Appendix provides another example to illustrate the differences between the three models. This example has a strong negative trend component with a superimposed mean-reversion. Only the CNN+Transformer captures both type of patterns.

3.3 Implementation and Empirical Analysis

3.3.1 Data

We collect daily equity return data for the securities on CRSP from January 1978 through December 2016. We use the first part of the sample to estimate the various factor models, which gives us the residuals for the time period from January 1998 to December 2016 for the arbitrage trading. The arbitrage strategies trade on a daily frequency at the close of each day. We use the daily adjusted returns to account for dividends and splits and the one-month Treasury bill rates from the Kenneth French Data Library as the risk-free rate. In addition, we complement the stock returns with the 46 firm-specific characteristics from [15], which are listed in Table A.1. All these variables are constructed either from accounting variables from the CRSP/Compustat database or from past returns from CRSP. The full details on the construction of these variables are in the Internet Appendix of [15].

Our analysis uses only the most liquid stocks in order to avoid trading and market friction issues. More specifically, we consider only the stocks whose market capitalization at the previous month was larger than 0.01% of the total market capitalization at that previous month, which is the same selection criterion as in [45]. On average this leaves us with approximately the largest 550 stocks, which correspond roughly to the S&P 500 index. This is an unbalanced dataset, as the stocks that we consider each month need not be the same as in the next month, but it is essentially balanced on a daily frequency in rolling windows of up to one year in our trading period from 1998 through 2016. For each stock we have its cross-sectionally centered and rank-transformed characteristics of the previous month. This is a standard transformation to deal with the different scales which is robust to outliers and time-variation, and has also been used in [15], [45], [42], and [29].

Our daily residual time-series start in 1998 as we have a large number of missing values in daily individual stock returns prior to this date, but almost no missing daily values in our sample.⁹ We

⁹Of all the stocks that have daily returns observed in a the local lockback window of $L = 30$ days, only 0.1% have

want to point out that the time period after 1998 also seems to be more challenging for arbitrage trading or factor trading, and hence our results can be viewed as conservative lower bounds.

3.3.2 Factor model estimation

As discussed in Section 3.2.1, we construct the statistical arbitrage portfolios by using the residuals of a general factor model for the daily excess returns of a collection of stocks. In particular, we consider the three empirically most successful families of factor models in our implementation. For each family, we conduct a rolling window estimation to obtain daily residuals out of sample from 1998 through 2016. This means that the residual composition matrix Φ_{t-1} of equation (3.1) depends only on the information up to time $t - 1$, and hence there is no look-ahead bias in trading the residuals. The rolling window estimation is necessary because of the time-variation in risk exposure of individual stocks and the unbalanced nature of a panel of individuals stock returns.

The three classes of factor models consists of pre-specified factors, latent unconditional factors and latent conditional factors:

1. **Fama-French factors:** We consider 1, 3, 5 and 8 factors based on various versions and extensions of the Fama-French factor models and downloaded from the Kenneth French Data Library. We consider them as tradeable assets in our universe. Each model includes the previous one and adds additional characteristic-based risk factors:
 - (a) $K = 1$: CAPM model with the excess return of a market factor
 - (b) $K = 3$: Fama-French 3 factor model includes a market, size and value factor
 - (c) $K = 5$: Fama-French 3 factor model + investment and profitability factors
 - (d) $K = 8$: Fama-French 5 factor model + momentum, short-term reversal and long-term reversal factors.

We estimate the loadings of the individual stock returns daily with a linear regression on the factors with a rolling window on the previous 60 days and compute the residual for the current day out-of-sample. This is the same procedure as in [11]. At each day we only consider the stocks with no missing observations in the daily returns within the rolling window, which in any window removes at most 2% of the stocks given our market capitalization filter.

2. **PCA factors:** We consider 1, 3, 5, 8, 10, and 15 latent factors, which are estimated daily on a rolling window. At each time $t - 1$, we use the last 252 days, or roughly one trading year, to estimate the correlation matrix from which we extract the PCA factors.¹⁰ Then, we

a missing return the next day for the out-of-sample trading, in which case we do not trade this stock. Hence, our data set of stocks with market capitalization higher than 0.01% of the total market capitalization, has essentially no missing daily values on a local window for the time period after 1998.

¹⁰This is the same procedure as in [2].

use the last 60 days to estimate the loadings on the latent factors using linear regressions, and compute residuals for the current day out-of-sample. At each day we only consider the stocks with no missing observations in the daily returns during the rolling window, which in any window removes at most 2% of the stocks given our market capitalization filter.

3. IPCA factors: We consider 1, 3, 5, 8, 10, and 15 factors in the Instrumented PCA (IPCA) model of [42]. This is a conditional latent factor model, in which the loadings β_{t-1} are a linear function of the asset characteristics at time $t - 1$. As the characteristics change at most each month, we reestimate the IPCA model on rolling window every year using the monthly returns and characteristics of the last 240 months. The IPCA provides the factor weights and loadings for each stock as a function of the stock characteristics. Hence, we do not need to estimate the loadings for individual stocks with an additional time-series regression, but use the loading function and the characteristics at time $t - 1$ to obtain the out-of-sample residuals at time t . The other details of the estimation process are carried out in the way outlined in [42].

In addition to the factor models above, we also include the excess returns of the individual stocks without projecting out the factors. This “zero-factor model” simply consists of the original excess returns of stocks in our universe and is denoted as $K = 0$. For each factor model, in our empirical analysis we observe that the cumulative residuals exhibit consistent and relatively regular mean-reverting behavior, with some occasional jumps. After taking out sufficiently many factors, the residuals of different stocks are only weakly correlated.

3.3.3 Implementation

Given the daily out-of-sample residuals from 1998 through 2016 we estimate the trading signal and policy on a rolling window to obtain the out-of-sample returns of the strategy. For each strategy we calculate the annualized sample mean μ , annualized volatility σ and annualized Sharpe ratio¹¹ $SR = \frac{\mu}{\sigma}$. The Sharpe ratio represents a risk-adjusted average return. Our main models estimate arbitrage strategies to maximize the Sharpe ratio without transaction costs. In Section 3.3.5 we also consider a mean-variance objective and in Section 3.3.11 we include transaction costs in the estimation and evaluation.

Our strategies trade the residuals of all stocks, which are mapped back into positions of the original stocks. We use the standard normalization that the absolute values of the individual stock portfolio weights sums up to one, i.e. we use the normalization $\|\omega_{t-1}^R\|_1 = 1$. This normalization implies a leverage constraint as short positions are bounded by one. The trading signal is based on a local lookback window of $L = 30$ days. We show in Section 3.3.9, that the results are robust to this choice and are very comparable for a lookback window of $L = 60$ days. Our main results use a

¹¹We obtain the annualized metrics from the daily returns using the standard calculations $\mu = \frac{252}{T} \sum_{t=1}^T R_t$ and $\sigma = \sqrt{\frac{252}{T} \sum_{t=1}^T (R_t - \mu)^2}$.

rolling window of 1,000 days to estimate the deep learning models. For computational reasons we re-estimate the network only every 125 days using the previous 1,000 days. Section 3.3.9 shows that our results are robust to this choice. Our main results show the out-of-sample trading performance from January 2002 to December 2016 as we use the first four years to estimate the signal and allocation function.

The hyperparameters for the deep learning models are based on the validation results summarized in Appendix A.3.1. Our benchmark model is a 2-layer CNN with $D = 8$ local convolutional filters and local window size of $D_{size} = 2$ days. The transformer has $H = 4$ attention heads, which can be interpreted as capturing four different global patterns. The results are extremely robust to the choice of hyperparameters. Appendix A.2.4 includes all the technical details for implementing and estimating the deep learning models.

3.3.4 Main Results

Table 3.1 displays the main results for various arbitrage models. It reports the annualized Sharpe ratio, mean return and volatility for our principal deep trading strategy CNN+Transformer and the two benchmark models, Fourier+FFN and OU+Threshold, for every factor model described in Section 3.3.2. The CNN+Transformer model and Fourier+FFN model are estimated with a Sharpe ratio objective. We obtain the daily out-of-sample residuals for different number of factors K for the time period January 1998 to December 2016. The daily returns of the out-of-sample arbitrage trading is then evaluated from January 2002 to December 2016, as we use a rolling window of four years to estimate the deep learning models.

First, we confirm that it is crucial to apply arbitrage trading to residuals and not individual stock returns. The stock returns, denoted as the $K = 0$ model, perform substantially worse than any type of residual within the same model and factor family. This is not surprising as residuals for an appropriate factor model are expected to be better described by a model that captures mean reversion. Importantly, individual stock returns are highly correlated and a substantial part of the returns is driven by the low dimensional factor component.¹² Hence, the complex nonparametric models are actually not estimated on many weakly dependent residual time-series, but most time-series have redundant information. In other words, the models are essentially calibrated on only a few factor time-series, which severely limits the structure that can be estimated. However, once we extract around $K = 5$ factors with any of the different factor models, the performance does not substantially increase by adding more factors. This suggests that most of commonality is explained by a small number of factors.

Second, the CNN+Transformer model strongly dominates the other benchmark models in terms of Sharpe ratio and average return. The Sharpe ratio is approximately twice as large as for a comparable Fourier+FFN model and four times higher for the corresponding parametric OU+Threshold

¹²[70] shows that around one third of the individual stock returns is explained by a latent four-factor model.

Table 3.1: OOS Annualized Performance Based on Sharpe Ratio Objective

Model	K	Factors			Fama-French			PCA			IPCA		
		SR	μ	σ	SR	μ	σ	SR	μ	σ	SR	μ	σ
CNN + Trans	0	1.64	13.7%	8.4%	1.64	13.7%	8.4%	1.64	13.7%	8.4%	3.22	8.7%	2.7%
	1	3.68	7.2%	2.0%	2.74	15.2%	5.5%	3.93	8.6%	2.2%	4.16	8.7%	2.1%
	3	3.13	5.5%	1.8%	3.56	16.0%	4.5%	3.95	8.2%	2.1%	3.97	8.0%	2.0%
	5	3.21	4.6%	1.4%	3.36	14.3%	4.2%	4.17	8.4%	2.0%	-	-	-
	8	2.49	3.4%	1.4%	3.02	12.2%	4.0%	3.95	8.2%	2.1%	2.81	10.7%	3.8%
	10	-	-	-	2.81	10.7%	3.8%	3.97	8.0%	2.0%	-	-	-
	15	-	-	-	2.30	7.6%	3.3%	4.17	8.4%	2.0%	-	-	-
Fourier + FFN	0	0.36	4.9%	13.6%	0.36	4.9%	13.6%	0.36	4.9%	13.6%	1.24	6.3%	5.0%
	1	0.89	3.2%	3.5%	0.80	8.4%	10.6%	1.77	7.8%	4.4%	1.90	7.7%	4.1%
	3	1.32	3.5%	2.7%	1.66	11.2%	6.7%	1.94	7.8%	4.0%	1.94	7.8%	4.0%
	5	1.66	3.1%	1.8%	1.98	12.4%	6.3%	1.93	7.6%	3.9%	1.71	8.2%	4.8%
	8	1.90	3.1%	1.6%	1.95	10.1%	5.2%	2.06	7.9%	3.8%	-	-	-
	10	-	-	-	1.71	8.2%	4.8%	1.93	7.6%	3.9%	-	-	-
	15	-	-	-	1.14	4.8%	4.2%	-	-	-	2.06	7.9%	3.8%
OU + Thresh	0	-0.18	-2.4%	13.3%	-0.18	-2.4%	13.3%	-0.18	-2.4%	13.3%	0.60	3.0%	5.1%
	1	0.16	0.6%	3.8%	0.21	2.1%	10.4%	0.88	3.8%	4.3%	0.77	5.2%	6.8%
	3	0.54	1.6%	3.0%	0.77	5.2%	6.8%	0.97	3.8%	4.0%	0.38	0.9%	2.3%
	5	0.38	0.9%	2.3%	0.73	4.4%	6.1%	0.91	3.5%	3.8%	1.16	2.8%	2.4%
	8	1.16	2.8%	2.4%	0.87	4.4%	5.1%	0.86	3.1%	3.6%	-	-	-
	10	-	-	-	0.63	2.9%	4.6%	0.93	3.2%	3.5%	-	-	-
	15	-	-	-	0.62	2.4%	3.8%	-	-	-	0.62	2.4%	3.5%

This table shows the out-of-sample annualized Sharpe ratio (SR), mean return (μ), and volatility (σ) of our three statistical arbitrage models for different numbers of risk factors K , that we use to obtain the residuals. We use the daily out-of-sample residuals from January 1998 to December 2016 and evaluate the out-of-sample arbitrage trading from January 2002 to December 2016. CNN+Trans denotes the convolutional network with transformer model, Fourier+FFN estimates the signal with a FFT and the policy with a feedforward neural network and lastly, OU+Thres is the parametric Ornstein-Uhlenbeck model with thresholding trading policy. The two deep learning models are calibrated on a rolling window of four years and use the Sharpe ratio objective function. The signals are extracted from a rolling window of $L = 30$ days. The $K = 0$ factor model corresponds to directly using stock returns instead of residuals for the signal and trading policy.

model. Using IPCA residuals, the CNN+Transformer achieves the impressive out-of-sample Sharpe ratio of around 4, in spite of trading only the most liquid large cap stocks and the time period after 2002. The mean returns of the CNN+Transformer are similar to the Fourier+FFN model, but have substantially smaller volatilities, which results in the higher Sharpe ratios. The parametric mean-reversion model achieves positive mean returns with Sharpe ratios close to one for the IPCA residuals, but as expected is too restrictive relative to the flexible models. The Fourier+FFN has the same flexibility as the CNN+Transformer in its allocation function, but is restricted to a pre-specified signal structure. The difference in performance quantifies the importance of extracting the complex time-series signals.

Third, the average return of the arbitrage strategies is large in spite of the leverage constraints.

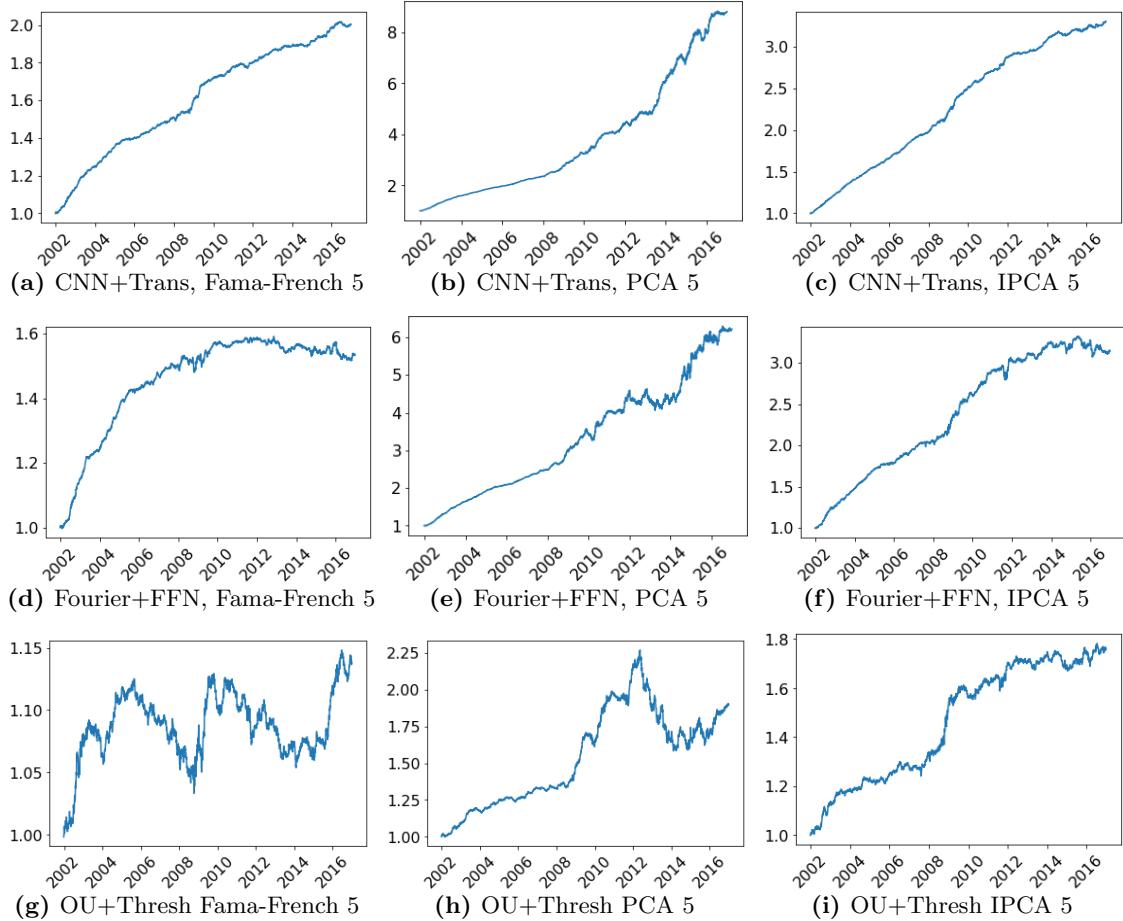
Table 3.2: Significance of Arbitrage Alphas based on Sharpe Ratio Objective

CNN+Trans model															
K	Fama-French					PCA					IPCA				
	α	t_α	R^2	μ	t_μ	α	t_α	R^2	μ	t_μ	α	t_α	R^2	μ	t_μ
0	11.6%	6.4***	30.3%	13.7%	6.3***	11.6%	6.4***	30.3%	13.7%	6.3***	11.6%	6.4***	30.3%	13.7%	6.3***
1	7.0%	14***	2.4%	7.2%	14***	14.9%	10***	0.6%	15.2%	11***	8.1%	12***	9.5%	8.7%	12***
3	5.5%	12***	1.2%	5.5%	12***	15.8%	14***	1.7%	16.0%	14***	8.2%	15***	6.0%	8.6%	15***
5	4.5%	12***	2.3%	4.6%	12***	14.1%	13***	1.3%	14.3%	13***	8.3%	16***	3.9%	8.7%	16***
8	3.3%	9.4***	2.1%	3.4%	9.6***	12.0%	12***	0.9%	12.2%	12***	7.8%	15***	5.0%	8.2%	15***
10	-	-	-	-	-	10.5%	11***	0.7%	10.7%	11***	7.7%	15***	4.0%	8.0%	15***
15	-	-	-	-	-	7.5%	8.8***	0.5%	7.6%	8.9***	8.1%	16***	4.2%	8.4%	16***
Fourier+FFN model															
K	Fama-French					PCA					IPCA				
	α	t_α	R^2	μ	t_μ	α	t_α	R^2	μ	t_μ	α	t_α	R^2	μ	t_μ
0	2.7%	0.8	8.6%	4.9%	1.4	2.7%	0.8	8.6%	4.9%	1.4	2.7%	0.8	8.6%	4.9%	1.4
1	3.0%	3.3**	3.3%	3.2%	3.5***	7.4%	2.7**	3.3%	8.4%	3.1**	4.8%	4.0***	16.4%	6.3%	4.8***
3	3.2%	4.7***	4.2%	3.5%	5.1***	10.9%	6.3***	2.2%	11.2%	6.4***	6.8%	6.4***	13.0%	7.8%	6.9***
5	2.9%	6.1***	3.5%	3.1%	6.4***	12.1%	7.5***	1.5%	12.4%	7.6***	6.7%	6.9***	13.3%	7.7%	7.4***
8	3.0%	7.2***	3.2%	3.1%	7.4***	10.0%	7.5***	0.9%	10.1%	7.6***	6.8%	7.0***	13.3%	7.8%	7.5***
10	-	-	-	-	-	8.0%	6.5***	1.0%	8.2%	6.6***	6.8%	7.1***	12.7%	7.6%	7.5***
15	-	-	-	-	-	4.7%	4.3***	0.4%	4.8%	4.4***	7.1%	7.6***	12.2%	7.9%	8.0***
OU+Thresh model															
K	Fama-French					PCA					IPCA				
	α	t_α	R^2	μ	t_μ	α	t_α	R^2	μ	t_μ	α	t_α	R^2	μ	t_μ
0	-4.5%	-1.4	13.4%	-2.4%	-0.7	-4.5%	-1.4	13.4%	-2.4%	-0.7	-4.5%	-1.4	13.4%	-2.4%	-0.7
1	-0.2%	-0.2	13.5%	0.6%	0.6	0.7%	0.3	6.3%	2.1%	0.8	1.7%	1.4	18.9%	3.0%	2.3*
3	0.9%	1.2	10.4%	1.6%	2.1*	4.3%	2.5*	4.3%	5.2%	3.0**	2.6%	2.6**	18.8%	3.8%	3.4***
5	0.5%	0.9	6.8%	0.9%	1.5	3.7%	2.4*	3.2%	4.4%	2.8**	2.8%	3.0**	17.7%	3.8%	3.8***
8	0.6%	1.2	5.5%	1.0%	1.9	3.9%	3.0**	1.9%	4.4%	3.4***	2.3%	2.6**	17.6%	3.5%	3.6***
10	-	-	-	-	-	2.6%	2.2*	1.4%	2.9%	2.4*	2.1%	2.5*	17.6%	3.1%	3.3***
15	-	-	-	-	-	2.1%	2.1*	0.7%	2.4%	2.4*	2.3%	2.8**	18.1%	3.2%	3.6***

This table shows the out-of-sample pricing errors α of the arbitrage strategies relative of the Fama-French 8 factor model and their mean returns μ for the different arbitrage models and different number of factors K that we use to obtain the residuals. We run a time-series regression of the out-of-sample returns of the arbitrage strategies on the 8-factor model (Fama-French 5 factors + momentum + short-term reversal + long-term reversal) and report the annualized α , accompanying t-statistic value t_α , and the R^2 of the regression. In addition, we report the annualized mean return μ along with its accompanying t-statistic t_μ . The hypothesis test are two-sided and stars indicate p-values of 5% (*), 1% (**), and 0.1% (***)]. All results use the out-of-sample daily returns from January 2002 to December 2016 and the deep learning models are based on a Sharpe ratio objective.

Normalizing the individual stock weights w_{t-1}^R so sum up in absolute value to one limits the short-selling. The CNN+Transformer with a five-factor PCA residual achieves an attractive annual mean return of around 14%. This means that the strategies do not require an infeasible amount of leverage to yield an average return that might be required by investors. In other words, the high Sharpe ratios are not the results of vanishing volatility but a combination of high average returns with moderate volatility.

Fourth, the arbitrage strategies are qualitatively robust to the choice of factor models to obtain residuals. The Fama-French and PCA factor lead to very similar Sharpe ratio results, suggesting that they explain a similar amount of co-movement in the data. However, as the mean returns of PCA factors are usually higher than the mean returns of the Fama-French factors, the risk factors are

Figure 3.6: Cumulative OOS Returns of Different Arbitrage Strategies

These figures show the cumulative daily returns of the arbitrage strategies for our representative models on the out-of-sample trading period between January 2002 and December 2016. We estimate the optimal arbitrage trading strategies for our three benchmark models based on the out-of-sample residuals of the Fama-French, PCA and IPCA 5-factor models. The deep learning models use the Sharpe ratio objective.

different. This confirms the findings of [70] and [51], who show that PCA factors do not coincide with Fama-French type factors and explain different mean returns. The IPCA factors use the additional firm-specific characteristic information. The resulting residuals achieve the highest Sharpe ratios, which illustrates that conditional factor models can capture more information than unconditional models. Including the momentum and reversal factors in the Fama-French 8 factor models to obtain residuals still results in profitable arbitrage strategies. Hence, the arbitrage strategies are not simply capturing a price trend risk premium.

The returns of the CNN+Transformer arbitrage strategies are statistically significant and not subsumed by conventional risk factors. Table 3.2 reports the out-of-sample pricing errors α of the

arbitrage strategies relative of the Fama-French 8 factor model and their mean returns μ . We run a time-series regression of the out-of-sample returns of the arbitrage strategies on the 8-factor model (Fama-French 5 factors + momentum + short-term reversal + long-term reversal) and report the annualized α , accompanying t-statistic value t_α , and the R^2 of the regression. In addition, we report the annualized mean return μ along with its accompanying t-statistic t_μ . The arbitrage strategies for the CNN+Transformer model for $K \geq 1$ are all statistically significant and not explained by the Fama-French 5 factors or price trend factors. Importantly, the pricing errors are essentially as large as the mean returns, which implies that the returns of the CNN+Transformer arbitrage strategies do not carry any risk premium of these eight factors. This is supported by the R^2 values, which are close to zero for the Fama-French or PCA residuals, and hence confirm that these arbitrage portfolios are essentially orthogonal to the Fama-French 8 factors. In contrast, one third of the individual return variation for $K = 0$ is explained by those risk factors. However, even in that case the pricing errors are significant. The residuals of IPCA factors have a higher correlation with the Fama-French 8 factors, suggesting that the conditional IPCA factor model extracts factors that are inherently different from the conventional risk factors. Note that the residuals of a Fama-French 8 factor model are not mechanically orthogonal in the time-series regression on the Fama-French 8 factors, as we construct out-of-sample residuals based on rolling window estimates. The parametric arbitrage strategies are largely explained by conventional risk factors. The third subtable in Table 3.2 shows that the residuals of the OU+Threshold model for Fama-French of PCA residuals do not have statistically significant pricing errors on a 1% level.

The CNN+Transformer has a consistent out-of-sample performance and is not affected by various negative events. Figure 3.6 shows the cumulative out-of-sample returns of the arbitrage strategies for our representative models. We select the residuals of the various five-factor models as including additional factors has only minor effects on the performance. Note that the CNN+Transformer model has consistently almost always positive returns, while maintaining a low volatility and avoiding any large losses. Importantly, the performance of the CNN+Transformer is nearly completely immune to both the “quant quake” which affected quantitative trading groups and funds engaging in statistical arbitrage in August 2007 ([3]), and the period of poor performance in quant funds during 2011–2012 ([23]). The Fourier+FFN model also performs similarly well until the financial crisis, but its risk increases afterwards as displayed by the larger volatility and larger drawdowns. The performance of the parametric model is visibly inferior. This illustrates that although all strategies trade the same residuals, which should be orthogonal to common market risk, profitable arbitrage trading requires an appropriate signal and allocation policy.

3.3.5 Mean-Variance Objective

The deep learning statistical arbitrage strategies can achieve high average returns in spite of leverage constraints. Our main deep learning models are estimated with a Sharpe ratio objective. As the sum

Table 3.3: OOS Annualized Performance Based on Mean-Variance Objective

CNN+Trans strategy, mean-variance objective function											
K	Fama-French			PCA			IPCA			μ	σ
	SR	μ	σ	SR	μ	σ	SR	μ	σ		
0	0.83	9.5%	11.4%	0.83	9.5%	11.4%	0.83	9.5%	11.4%		
1	3.15	10.5%	3.3%	2.21	27.3%	12.3%	2.83	15.9%	5.6%		
3	2.95	7.8%	2.6%	2.38	22.6%	9.5%	3.13	17.9%	5.7%		
5	3.03	5.9%	2.0%	2.75	19.6%	7.1%	3.21	18.2%	5.7%		
8	2.96	4.2%	1.4%	2.68	16.6%	6.2%	3.18	17.0%	5.4%		
10	-	-	-	2.67	15.3%	5.7%	3.21	16.6%	5.2%		
15	-	-	-	2.20	8.7%	4.0%	3.34	16.3%	4.9%		

Fourier+FFN strategy, mean-variance objective function											
K	Fama-French			PCA			IPCA			μ	σ
	SR	μ	σ	SR	μ	σ	SR	μ	σ		
0	0.28	5.5%	19.3%	0.28	5.5%	19.3%	0.28	5.5%	19.3%		
1	0.38	2.5%	6.7%	0.48	16.6%	34.8%	0.56	9.7%	17.2%		
3	1.16	4.3%	3.7%	0.34	32.1%	93.1%	1.06	17.6%	16.7%		
5	1.30	3.1%	2.4%	0.37	22.5%	61.2%	1.17	17.0%	14.5%		
8	1.73	3.6%	2.0%	0.67	17.4%	25.9%	1.21	14.4%	11.9%		
10	-	-	-	0.45	7.4%	16.4%	1.06	12.6%	11.9%		
15	-	-	-	0.56	5.7%	10.2%	1.17	12.1%	10.4%		

This table shows the out-of-sample annualized Sharpe ratio (SR), mean return (μ), and volatility (σ) of our CNN+Transformer and Fourier+FFN models for different numbers of risk factors K , that we use to obtain the residuals. We use a mean-variance objective function with risk aversion $\gamma = 1$. We use the daily out-of-sample residuals from January 1998 to December 2016 and evaluate the out-of-sample arbitrage trading from January 2002 to December 2016. The two deep learning models are calibrated on a rolling window of four years. The signals are extracted from a rolling window of $L = 30$ days. The $K = 0$ factor model corresponds to directly using stock returns instead of residuals for the signal and trading policy.

of absolute stock weights is normalized to one, the arbitrage strategies impose an implicit leverage constraint. We show that the average return can be increased while maintaining this leverage constraint. For this purpose we change the objective for the deep learning model to a mean-variance objective. In order to illustrate the effect of the different objective function, we set the risk aversion parameter to $\gamma = 1$.

Tables 3.3 and 3.4 collect the results for the Sharpe ratio, mean, volatility and significance tests. As expected the Sharpe ratios are slightly lower compared to the corresponding model with Sharpe ratio objective, but the mean returns are substantially increased. The CNN+Transformer model achieves average annual returns around 20% with PCA and IPCA residuals while the volatility is only around half as large as the one of a market portfolio. The mean returns are statistically highly significant and not spanned by conventional risk factors or a price trend risk premium. The Fourier+FFN model can also obtain high average returns, but those come at the cost of a substantial volatility. Overall, we confirm that the more flexible signal extraction function of the CNN+Transformer is crucial for the superior performance.

Table 3.4: Significance of Arbitrage Alphas based on Mean-Variance Objective

CNN+Trans model															
K	Fama-French					PCA					IPCA				
	α	t_α	R^2	μ	t_μ	α	t_α	R^2	μ	t_μ	α	t_α	R^2	μ	t_μ
0	5.8%	2.2*	19.6%	9.5%	3.2**	5.8%	2.2*	19.6%	9.5%	3.2**	5.8%	2.2*	19.6%	9.5%	3.2**
1	9.9%	12***	7.1%	10.5%	12***	26.3%	8.3***	1.6%	27.3%	8.6***	14.0%	11***	23.5%	15.9%	11***
3	7.5%	11***	5.3%	7.8%	11***	22.1%	9.1***	2.2%	22.6%	9.2***	16.6%	12***	17.6%	17.9%	12***
5	5.7%	11***	5.3%	5.9%	12***	19.0%	10***	3.2%	19.6%	11***	16.7%	12***	16.0%	18.2%	12***
8	4.4%	9.8***	3.6%	4.6%	10***	16.3%	10***	1.6%	16.6%	10***	15.5%	12***	18.3%	17.0%	12***
10	-	-	-	-	-	14.8%	10***	1.7%	15.3%	10***	15.2%	13***	20.6%	16.6%	12***
15	-	-	-	-	-	8.5%	8.4***	0.9%	8.7%	8.5***	14.8%	13***	21.6%	16.3%	13***

Fourier+FFN model															
K	Fama-French					PCA					IPCA				
	α	t_α	R^2	μ	t_μ	α	t_α	R^2	μ	t_μ	α	t_α	R^2	μ	t_μ
0	3.2%	0.7	8.4%	5.5%	1.1	3.2%	0.7	8.4%	5.5%	1.1	3.2%	0.7	8.4%	5.5%	1.1
1	2.8%	1.6	1.8%	2.5%	1.5	15.4%	1.7	1.3%	16.6%	1.9	7.9%	1.8	2.6%	9.7%	2.2*
3	4.1%	4.4***	3.4%	4.3%	4.5***	30.3%	1.3	0.1%	32.1%	1.3	17.4%	4.1***	1.9%	17.6%	4.1***
5	2.9%	4.8***	3.1%	3.1%	5.0***	21.0%	1.3	0.1%	22.5%	1.4	15.9%	4.3***	2.6%	17.0%	4.5***
8	3.5%	6.8***	2.3%	3.6%	7.0***	17.4%	2.6**	0.3%	17.2%	2.6**	12.9%	4.3***	4.4%	14.4%	4.7***
10	-	-	-	-	-	7.1%	1.7	0.3%	7.4%	1.8	11.7%	3.9***	3.5%	12.6%	4.1***
15	-	-	-	-	-	5.5%	2.1*	0.1%	5.7%	2.2*	11.3%	4.3***	4.0%	12.1%	4.5***

This table shows the out-of-sample pricing errors α of the arbitrage strategies relative of the Fama-French 8 factor model and their mean returns μ for the different arbitrage models and different number of factors K that we use to obtain the residuals. We use a mean-variance objective function with risk aversion $\gamma = 1$. We run a time-series regression of the out-of-sample returns of the arbitrage strategies on the 8-factor model (Fama-French 5 factors + momentum + short-term reversal + long-term reversal) and report the annualized α , accompanying t-statistic value t_α , and the R^2 of the regression. In addition, we report the annualized mean return μ along with its accompanying t-statistic t_μ . The hypothesis test are two-sided and stars indicate p-values of 5% (*), 1% (**), and 0.1% (***)�. All results use the out-of-sample daily returns from January 2002 to December 2016.

3.3.6 Unconditional Residual Means

The unconditional average of residuals is not a profitable strategy and does not provide information about the potential arbitrage profitability contained in the residuals. A natural question to ask is if the residuals themselves have a risk premium component and if trading an equally weighted portfolio of residuals could be profitable. Table 3.5 shows the performance of this simple strategy. If we do not project out any factors ($K = 0$), this strategy essentially trades an equally weighted market portfolio. Table 3.6 reports the test statistics relative to the Fama-French 8 factor model, which completely subsumes the market risk premium. Once we regress out at least 3 factors, the equally weighted residuals have a mean return of around 1% or lower. The low volatility confirms that the residuals are only weakly cross-sectionally dependent and are largely diversified away. The moderately large Sharpe ratios for PCA residuals is a consequence of the near zero volatility. Scaling up the mean returns to a meaningful magnitude would potentially require an unreasonable amount of leverage. Overall, we confirm that residuals need to be combined with a signal and trading policy that takes advantage of the time series patterns in order to achieve a profitable strategy.

Table 3.5: OOS Annualized Performance of Unconditional Average Residuals

Equally Weighted Residuals											
K	Fama-French			PCA			IPCA			μ	σ
	SR	μ	σ	SR	μ	σ	SR	μ	σ		
0	0.52	11.2%	21.4%	0.52	11.2%	21.4%	0.52	11.2%	21.4%		
1	0.39	1.9%	4.8%	-0.23	-0.4%	1.5%	0.76	3.2%	4.2%		
3	0.18	0.7%	3.7%	0.34	0.3%	0.9%	0.76	2.0%	2.7%		
5	0.22	0.8%	3.5%	0.93	0.7%	0.7%	0.63	1.4%	2.3%		
8	-0.17	-0.5%	2.9%	1.04	0.6%	0.5%	0.66	1.4%	2.2%		
10	-	-	-	0.90	0.4%	0.5%	0.65	1.3%	2.1%		
15	-	-	-	1.08	0.4%	0.4%	0.62	1.3%	2.0%		

This table shows the out-of-sample annualized Sharpe ratio (SR), mean return (μ), and volatility (σ) of equally weighted residuals. We evaluate the out-of-sample arbitrage trading from January 2002 to December 2016. The $K = 0$ factor model corresponds to directly using stock returns instead of residuals for the signal and trading policy.

IPCA factors are close to uncorrelated with conventional risk factors. The R^2 values in Table 3.6 are as expected for the Fama-French factors and, not surprisingly, after regressing out all of those factors, the cross-sectional average of the residuals is essentially orthogonal to those factors. The PCA residuals show a very similar behavior. However, the conditional IPCA model leaves a component in the residuals that it is highly correlated with conventional risk factors. In this sense, the IPCA factors extract a factor model that is quite different from the Fama-French factors.

Importantly, unconditional means and alphas of asset pricing residuals are a poor measure of arbitrage opportunities. The mean and alphas of residuals that are optimally traded based on their time series patterns have mean returns that can be larger by a factor of 50. This implies more generally, that the unconditional perspective of evaluating asset pricing models could potentially overstate the efficiency of markets and the pricing ability of asset pricing models.

Table 3.6: Significance of Arbitrage Alphas Based on Unconditional Average Residuals

K	Equally Weighted Residuals														
	Fama-French					PCA					IPCA				
	α	t_α	R^2	μ	t_μ	α	t_α	R^2	μ	t_μ	α	t_α	R^2	μ	t_μ
0	1.4%	1.4	97.0%	11.2%	2.0*	1.4%	1.4	97.0%	11.2%	2.0*	1.4%	1.4	97.0%	11.2%	2.0*
1	0.4%	0.4	36.6%	1.9%	1.5	0.0%	0.0	25.8%	-0.4%	-0.9	0.4%	1.1	85.0%	3.2%	2.9**
3	0.4%	0.4	9.6%	0.7%	0.7	0.4%	1.9	13.1%	0.3%	1.3	0.9%	3.3**	84.1%	2.0%	2.9**
5	0.2%	0.2	7.0%	0.8%	0.9	0.7%	4.2***	5.9%	0.7%	3.6***	0.4%	2.0*	89.4%	1.4%	2.4*
8	-0.6%	-0.8	0.7%	-0.5%	-0.7	0.6%	4.5***	4.1%	0.6%	4.0***	0.4%	2.1*	89.3%	1.4%	2.5*
10	-	-	-	-	-	0.5%	3.8***	3.0%	0.4%	3.5***	0.3%	1.9	89.4%	1.3%	2.5*
15	-	-	-	-	-	0.4%	4.3***	2.0%	0.4%	4.2***	0.3%	1.6	89.0%	1.3%	2.4*

This table shows the out-of-sample pricing errors α of cross-sectionally equally weighted residuals relative of the Fama-French 8 factor model and their mean returns μ for the different arbitrage models and different number of factors K that we use to obtain the residuals. We run a time-series regression of the out-of-sample returns of the arbitrage strategies on the 8-factor model (Fama-French 5 factors + momentum + short-term reversal + long-term reversal) and report the annualized α , accompanying t-statistic value t_α , and the R^2 of the regression. In addition, we report the annualized mean return μ along with its accompanying t-statistic t_μ . The hypothesis test are two-sided and stars indicate p-values of 5% (*), 1% (**), and 0.1% (***)). All results use the out-of-sample daily returns from January 2002 to December 2016.

3.3.7 Importance of Time-Series Signal

How important is the flexibility in the signal extraction function relative to the allocation function? So far, we have considered a rigid parametric model for the signal and allocation function and a flexible allocation function but either a pre-specified time-series filter or a data-driven flexible filter. In A.7 in Appendix we also report the results for two additional model variations, which serve as ablation tests emphasizing the central importance of applying a time-series model to extract a signal extraction function from the data.

The first model, OU+FFN, uses the same 4-dimensional OU signal as the OU+Threshold policy, but replaces the threshold allocation function with an FFN allocation function. This FFN allocation function has the same architecture as that of the Fourier+FFN policy, except the input is 4-dimensional instead of 30-dimensional. The results show that even despite using a very flexible allocation function, the results are similar or even worse than the simple parametric thresholding rule. This points to the weakness of the OU signal representation: although the allocation function is a powerful universal approximator, it cannot accomplish much with an information-poor input. If the optimal allocation function given the simple OU signal is well approximated by the parametric thresholding rule, then the nonparametric FFN offers too much flexibility without comparable efficiency, which leads to a noisier estimate of a simple function and hence worse out-of-sample performance.

The second model does not extract a time-series signal from the residuals, but uses the residuals themselves as signal to a flexible FFN allocation function. As the allocation function uses the same type of network as for the CNN+Transformer, Fourier+FFN or OU+FFN, this setup directly

assesses the relevance of using a time-series model for the signal. The FFN model also performs worse than the deep learning models that apply a time-series filter to the residuals. This is a good example to emphasize the importance of a time-series model. While FFNs are flexible in learning low dimensional functional relationships, they are limited in learning a complex dependency model if the training data is limited. For example, the FFN is not sufficiently efficient to learn an FFT-like transformation and hence has a substantially worse performance on the original time-series compared to frequency-transformed time-series.

In summary, the flexible data-driven signal extraction function of the CNN+Transformer model seems to be the critical element for statistical arbitrage. A flexible allocation function is not sufficient to compensate for an uninformative signal.

3.3.8 Dependency between Arbitrage Strategies

The trading strategies for different factor models are only weakly correlated. In Table A.6, we report the correlations of the returns of our CNN+Transformer strategies across factor models with 3, 5, and 10 factors, based on the Sharpe ratio objective function strategy.¹³ Notably, the correlations between strategies from different factor model families range from roughly 0.2 to 0.45, indicating that strategies for different factor model families can be used as part of a diversification strategy. While the performance of the arbitrage trading for the residuals obtained with different families of factor models is comparable, the factors themselves are different. Hence, even if the arbitrage signal and allocation functions are similar, the resulting strategies can be weakly correlated. The within-family correlations range from 0.4 to 0.85, indicating that the residuals from the same class of factor model capture similar patterns.

3.3.9 Stability over Time

Our results are robust to length of the local window to extract the trading signal. We re-estimate the CNN+Transformer model on an extended rolling lookback window of $L = 60$ days, while keeping the rest of the model structure the same. Tables 3.7 and 3.8 show that the results are robust to the choice of lookback window. Extending the local window to 60 trading days, which is close to three months, leads to essentially the same performance as using only the most recent $L = 30$ trading days to infer the signal. This is further evidence that the arbitrage signal is different from conventional momentum or reversal strategies that incorporate information from longer time periods. As the signal can be inferred from the most recent past, it implies that either the arbitrage signal depends only on the most recent days or that those days are sufficient to infer the relevant time-series structure. In the next section, we provide evidence that the arbitrage trading signals put strong emphasis on most recent two weeks before the trading.

¹³The correlations for the mean-variance objective function are similar.

Table 3.7: OOS Annualized Performance of CNN+Trans for 60 Days Lookback Window

K	Fama-French			PCA			IPCA		
	SR	μ	σ	SR	μ	σ	SR	μ	σ
0	1.50	13.5%	9.0%	1.50	13.5%	9.0%	1.50	13.5%	9.0%
1	2.95	9.6%	3.2%	2.68	15.8%	5.9%	3.14	8.8%	2.8%
3	3.21	8.7%	2.7%	3.49	16.8%	4.8%	3.84	9.6%	2.5%
5	3.23	6.8%	2.1%	3.54	16.0%	4.5%	3.90	9.2%	2.4%
8	2.96	4.2%	1.4%	3.02	12.5%	4.2%	3.93	8.7%	2.2%
10	-	-	-	2.67	9.9%	3.7%	3.98	9.2%	2.3%
15	-	-	-	2.36	8.1%	3.4%	4.24	9.6%	2.3%

This table shows the out-of-sample annualized Sharpe ratio (SR), mean return (μ), and volatility (σ) of the CNN+Transformer model for different numbers of risk factors K , that we use to obtain the residuals. The signals are extracted from a rolling window of $L = 60$ days. We use the daily out-of-sample residuals from January 1998 to December 2016 and evaluate the out-of-sample arbitrage trading from January 2002 to December 2016. The model is calibrated on a rolling window of four years and uses the Sharpe ratio objective function. The $K = 0$ factor model corresponds to directly using stock returns instead of residuals for the signal and trading policy.

Table 3.8: Significance of Arbitrage Alphas for 60 Days Lookback Window

CNN+Trans Model , Sharpe objective function, $L = 60$ days lookback window															
K	Fama-French					PCA					IPCA				
	α	t_α	R^2	μ	t_μ	α	t_α	R^2	μ	t_μ	α	t_α	R^2	μ	t_μ
0	11.8%	5.6***	19.5%	13.5%	5.8***	11.8%	5.6***	19.5%	13.5%	5.8***	11.8%	5.6***	19.5%	13.5%	5.8***
1	9.1%	11***	7.2%	9.6%	11***	15.5%	10***	1.2%	15.8%	10***	8.2%	12***	10.1%	8.8%	12***
3	8.3%	12***	7.1%	8.7%	12***	16.5%	13***	2.5%	16.8%	14***	9.2%	15***	9.3%	9.6%	15***
5	6.5%	12***	6.0%	6.8%	13***	15.6%	13***	2.2%	16.0%	14***	8.8%	15***	10.3%	9.2%	15***
8	4.1%	11***	3.2%	4.2%	11***	12.2%	11***	1.6%	12.5%	12***	8.3%	15***	8.9%	8.7%	15***
10	-	-	-	-	-	9.7%	10***	1.0%	9.9%	10***	8.8%	15***	8.3%	9.2%	15***
15	-	-	-	-	-	8.1%	9.1***	0.7%	8.1%	9.1***	9.2%	16***	9.3%	9.6%	16***

This table shows the out-of-sample pricing errors α of the arbitrage strategies relative of the Fama-French 8 factor model and their mean returns μ for the CNN+Transformer model and different number of factors K that we use to obtain the residuals. The signals are extracted from a rolling window of $L = 60$ days. We run a time-series regression of the out-of-sample returns of the arbitrage strategies on the 8-factor model (Fama-French 5 factors + momentum + short-term reversal + long-term reversal) and report the annualized α , accompanying t-statistic value t_α , and the R^2 of the regression. In addition, we report the annualized mean return μ along with its accompanying t-statistic t_μ . The hypothesis test are two-sided and stars indicate p-values of 5% (*), 1% (**), and 0.1% (***)). All results use the out-of-sample daily returns from January 2002 to December 2016 and are based on a Sharpe ratio objective.

A constant-in-time signal and allocation function captures a large fraction of the arbitrage information. We re-estimate the CNN+Transformer model with a constant model instead of the rolling window calibration. Our main models are estimated on a rolling window of four years, which allows the models to adopt to changing economic conditions. Here we use either the first $T_{\text{train}} = 4$ years (1,000 trading days) or $T_{\text{train}} = 8$ years (2,000 trading days) to estimate the signal and allocation function, and then keep those functions constant for the remaining out-of-sample trading period. The results are reported in Tables 3.9 and 3.10. As expected the performance decreases relative to a time-varying model with re-estimation, which suggests that there is some degree of time-variation in the signal and allocation function. The longer training window of 8 years results in slightly higher Sharpe ratios than the 4 year window, as the model has more data and more variety in the market

Table 3.9: OOS Annualized Performance of CNN+Trans for Constant Model

$T_{\text{train}} = 4 \text{ years}$										
K	Fama-French			PCA			IPCA			
	SR	μ	σ	SR	μ	σ	SR	μ	σ	
0	1.10	8.5%	7.8%	1.10	8.5%	7.8%	1.10	8.5%	7.8%	
1	1.90	4.5%	2.3%	0.66	5.2%	7.9%	0.94	3.1%	3.3%	
3	1.60	3.6%	2.2%	1.65	8.7%	5.3%	1.82	5.3%	2.9%	
5	1.81	3.0%	1.7%	1.93	9.8%	5.1%	2.09	5.4%	2.6%	
8	1.70	2.5%	1.5%	2.04	9.6%	4.7%	1.89	5.0%	2.6%	
10	-	-	-	2.06	9.1%	4.4%	1.77	4.7%	2.7%	
15	-	-	-	1.82	7.0%	3.9%	2.09	5.5%	2.7%	

$T_{\text{train}} = 8 \text{ years}$										
K	Fama-French			PCA			IPCA			
	SR	μ	σ	SR	μ	σ	SR	μ	σ	
0	1.33	12.0%	9.0%	1.33	12.0%	9.0%	1.33	12.0%	9.0%	
1	2.06	5.0%	2.4%	1.81	15.2%	8.4%	2.02	8.5%	4.2%	
3	2.46	5.3%	2.2%	2.04	13.1%	6.4%	2.47	7.5%	3.0%	
5	1.82	3.2%	1.8%	1.91	11.9%	6.2%	2.64	7.6%	2.9%	
8	1.48	2.5%	1.7%	1.89	10.8%	5.7%	2.71	8.3%	3.1%	
10	-	-	-	1.82	10.0%	5.5%	2.68	8.2%	3.1%	
15	-	-	-	1.38	6.2%	4.5%	2.70	7.8%	2.9%	

This table shows the out-of-sample annualized Sharpe ratio (SR), mean return (μ), and volatility (σ) of the CNN+Transformer model for different numbers of risk factors K . We estimate the model on only once on the first T_{train} days and keep it constant on the remaining test set. We use the daily out-of-sample residuals from January 1998 to December 2016 and evaluate the out-of-sample arbitrage trading from January 1998 + T_{train} to December 2016. The signals are extracted from a rolling window of $L = 30$ days and we use the Sharpe ratio objective function.

environment to learn the arbitrage information. Importantly, the constant CNN+Transformer still substantially outperforms the other benchmark models, Fourier+FFN and OU+Threshold, even if those are estimated on a rolling window. We conclude that the constant signal and allocation function for the CNN+Transformer model already capture a substantial amount of arbitrage information. Therefore, the constant functions serve as a meaningful model to analyze in more detail in the next section.

3.3.10 Estimated Structure

What are the patterns that our CNN+Transformer model can learn and exploit? In order to answer this question, we analyze the different building blocks of our benchmark model and show their structure on representative and informative residuals inputs. Our goal is to ascertain, characterize, and explain the role that the convolutional features and attention heads play in the determination of the final allocation weight and recognition of salient time series patterns. The benchmark model for this section is the CNN+Transformer based on IPCA 5-factor residuals and a Sharpe ratio objective.

Table 3.10: Significance of Arbitrage Alphas for Constant Model

CNN+Trans model, Sharpe objective function, $T_{\text{train}} = 4$ years															
K	Fama-French					PCA					IPCA				
	α	t_α	R^2	μ	t_μ	α	t_α	R^2	μ	t_μ	α	t_α	R^2	μ	t_μ
0	8.4%	4.2***	3.0%	8.5%	4.3***	8.4%	4.2***	3.0%	8.5%	4.3***	8.4%	4.2***	3.0%	8.5%	4.3***
1	4.0%	6.8***	5.9%	4.5%	7.3***	4.1%	2.0*	4.5%	5.2%	2.5*	3.1%	3.7***	1.6%	3.1%	3.6***
3	3.2%	5.7***	4.9%	3.6%	6.2***	8.2%	6.1***	2.7%	8.7%	6.4***	5.3%	7.4***	11.7%	5.3%	7.0***
5	2.8%	6.6***	4.3%	3.0%	7.0***	9.3%	7.1***	1.8%	9.8%	7.5***	5.5%	8.6***	8.3%	5.4%	8.1***
8	2.3%	6.1***	5.1%	2.5%	6.6***	9.0%	7.5***	2.2%	9.6%	7.9***	5.0%	7.7***	8.2%	5.0%	7.3***
10	-	-	-	-	-	8.6%	7.5***	1.9%	9.1%	8.0***	5.1%	8.0***	16.6%	4.7%	6.9***
15	-	-	-	-	-	6.8%	6.8***	1.0%	7.0%	7.1***	5.8%	9.3***	17.6%	5.5%	8.1***

CNN+Trans model, Sharpe objective function, $T_{\text{train}} = 8$ years															
K	Fama-French					PCA					IPCA				
	α	t_α	R^2	μ	t_μ	α	t_α	R^2	μ	t_μ	α	t_α	R^2	μ	t_μ
0	10.1%	4.1***	18.1%	12.0%	4.4***	10.1%	4.1***	18.1%	12.0%	4.4***	10.1%	4.1***	18.1%	12.0%	4.4***
1	4.4%	6.5***	14.3%	5.0%	6.8***	14.5%	5.8***	2.5%	15.2%	6.0***	7.0%	6.6***	30.6%	8.5%	6.7***
3	4.9%	7.9***	11.6%	5.3%	8.2***	12.8%	6.7***	2.7%	13.1%	6.8***	7.0%	7.9***	8.2%	7.5%	8.2***
5	2.9%	5.8***	12.3%	3.2%	6.0***	11.6%	6.2***	1.6%	11.9%	6.3***	7.1%	8.7***	12.1%	7.6%	8.7***
8	2.3%	4.7***	5.4%	2.5%	4.9***	10.2%	6.0***	3.1%	10.8%	6.3***	7.7%	9.0***	14.6%	8.3%	9.0***
10	-	-	-	-	-	9.4%	5.7***	2.6%	10.0%	6.0***	7.7%	8.9***	11.3%	8.2%	8.9***
15	-	-	-	-	-	6.0%	4.4***	0.9%	6.2%	4.6***	7.4%	8.9***	11.2%	7.8%	8.9***

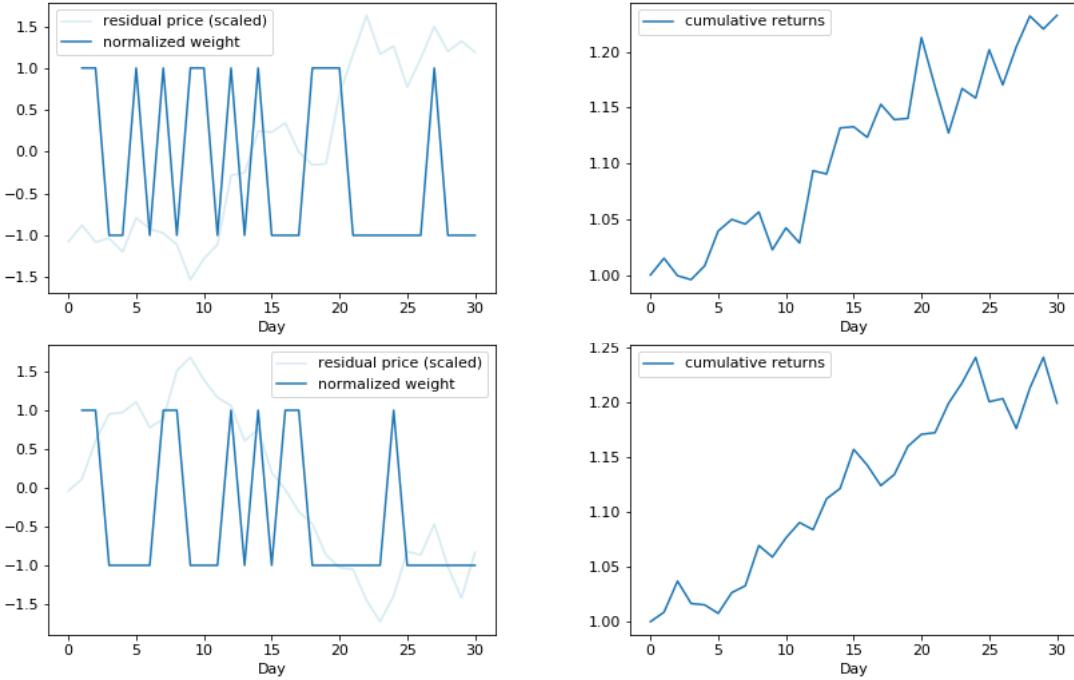
This table shows the out-of-sample pricing errors α of the arbitrage strategies relative of the Fama-French 8 factor model and their mean returns μ for the CNN+Transformer model and different number of factors K . We estimate the model on only once on the first T_{train} days and keep it constant on the remaining test set. We use the daily out-of-sample residuals from January 1998 to December 2016 and evaluate the out-of-sample arbitrage trading from January 1998 + T_{train} to December 2016. The signals are extracted from a rolling window of $L = 30$ days and we use the Sharpe ratio objective function. We run a time-series regression of the out-of-sample returns of the arbitrage strategies on the 8-factor model (Fama-French 5 factors + momentum + short-term reversal + long-term reversal) and report the annualized α , accompanying t-statistic value t_α , and the R^2 of the regression. In addition, we report the annualized mean return μ along with its accompanying t-statistic t_μ . The hypothesis test are two-sided and stars indicate p-values of 5% (*), 1% (**), and 0.1% (***)�.

The model is calibrated on the first 8 years of data and kept constant, which allows us to study the signal and allocation function.

As an illustrative example, Figure 3.7 shows the allocation and return on representative residuals. The left subplot displays an out-of-sample time-series of cumulative returns of a randomly selected residual and its value in the allocation function. We normalize the allocation weight to have an absolute value of one, that is, for this illustrative example we only trade this particular residual. The right subplots depicts the payoff of trading the specific residual with the displayed allocation function. The first residual shows mean-reversion patterns, which are successfully detected and exploited by the function $w^{e|\text{CNN+Trans}}$. The second residual has a downward trend, which is also correctly detected and taken advantage of by the model.¹⁴ These examples suggest that the CNN+Transformer model can learn mean-reversion and trend patterns. Figures 3.2 and A.2 are further examples with the same takeaways.

Next, we “dissect” the CNN+Transformer model to understand what type of functions it can estimate. Our analysis begins with the eight basic convolutional patterns learned by the convolutional

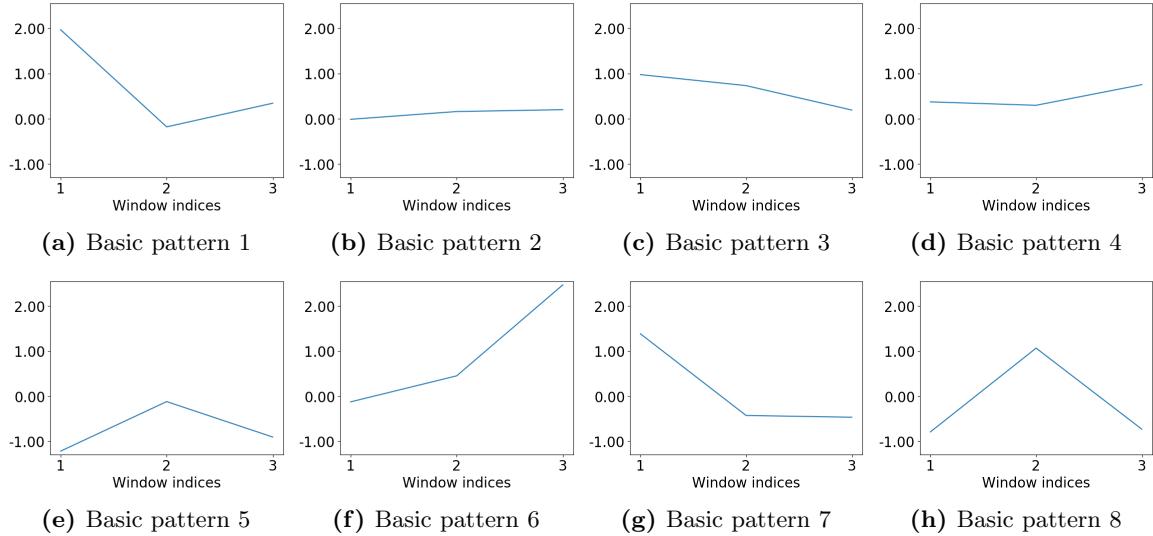
¹⁴Note that our in our empirical study the model trades in all residuals and is not limited to trade only in one residual. Hence, the empirical performance is substantially better as shown in Figure 3.6.

Figure 3.7: Examples of Allocation and Returns of CNN+Transformer Strategy

These plots display representative examples of the CNN+Transformer out-of-sample arbitrage trading on a sample of residuals from the IPCA 5-factor model. The left subplots show the normalized cumulative returns of the residuals and the normalized allocation weight, which the specific residual has in the trading strategy. The right subplots illustrate the payoff of trading the specific residual with the displayed allocation function. The model is calibrated on the first 8 years of data and kept constant.

layers of our network, which are displayed in Figure 3.8. The CNN represents a given time-series as a matrix of exposures to local basic patterns. As explained in Section 3.2.4, these local filters are more complicated than simple local linear filters, but we can project our CNN filters into two-dimensional orthogonal linear filters, which are more interpretable. These local patterns are the building blocks to construct global patterns. We see that these basic patterns display a variety of salient price behavior which are considered to be important. Basic patterns 4 and 6 capture local upward trends, basic patterns 3 and 7 track local downward trends and basic patterns 1, 5 and 8 learn reversion patterns. However, the basis patterns do not include very spiked, sharp changes. Overall, the building blocks seem to be sufficient to construct any smooth trend and mean-reversion pattern.

We can understand the global patterns learned by the transformer by studying the attention function. The attention functions $\alpha_i(.,.)$ of each attention head $i = 1, \dots, H = 4$ capture the dependencies between the local patterns. Our arbitrage signal can be interpreted as ‘‘loadings’’ to these ‘‘attention factors’’. We use the same H attention functions for all residuals, but in order to visualize them, we evaluate them for a given residual time-series, which yield the attention weights

Figure 3.8: Local Basic Patterns of Benchmark Model

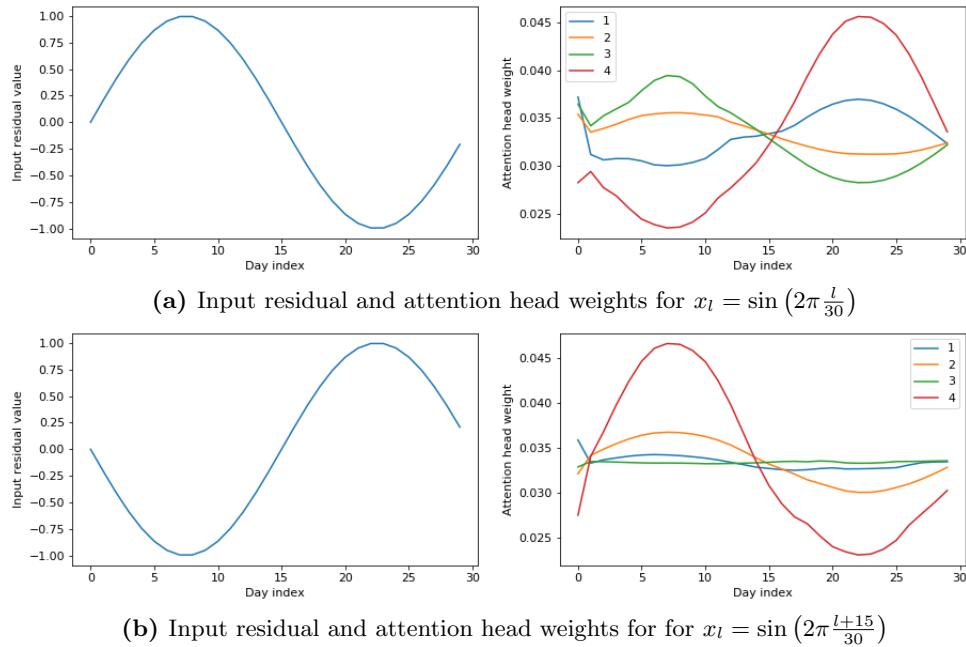
These figures show the $D = 8$ local filters of the CNN estimated for the benchmark model in our empirical analysis. These are projections of our higher dimensional nonlinear filter from a 2-layer CNN into two-dimensional linear filters. These building blocks are labeled “basic patterns”. The benchmark model is the CNN+Transformer model based on IPCA 5-factor residuals. We estimate the model on only once on the first $T_{\text{train}}=8$ years based on the Sharpe ratio objective.

per head:

$$\alpha_{i,l,j} = \boldsymbol{\alpha}_i(\tilde{x}_l, \tilde{x}_j) \quad \text{for } i = 1, \dots, H.$$

As our signal only depends on the final cumulative return projection h_L^{proj} , the attention weights $\alpha_{i,L,j}$ for $i = 1, \dots, H$ and $j = 1, \dots, L$ contain all the “global factor” information. Hence, we will plot the $H \times L$ dimensional attention weights of the attention heads to understand which global patterns are activated by specific time-series.

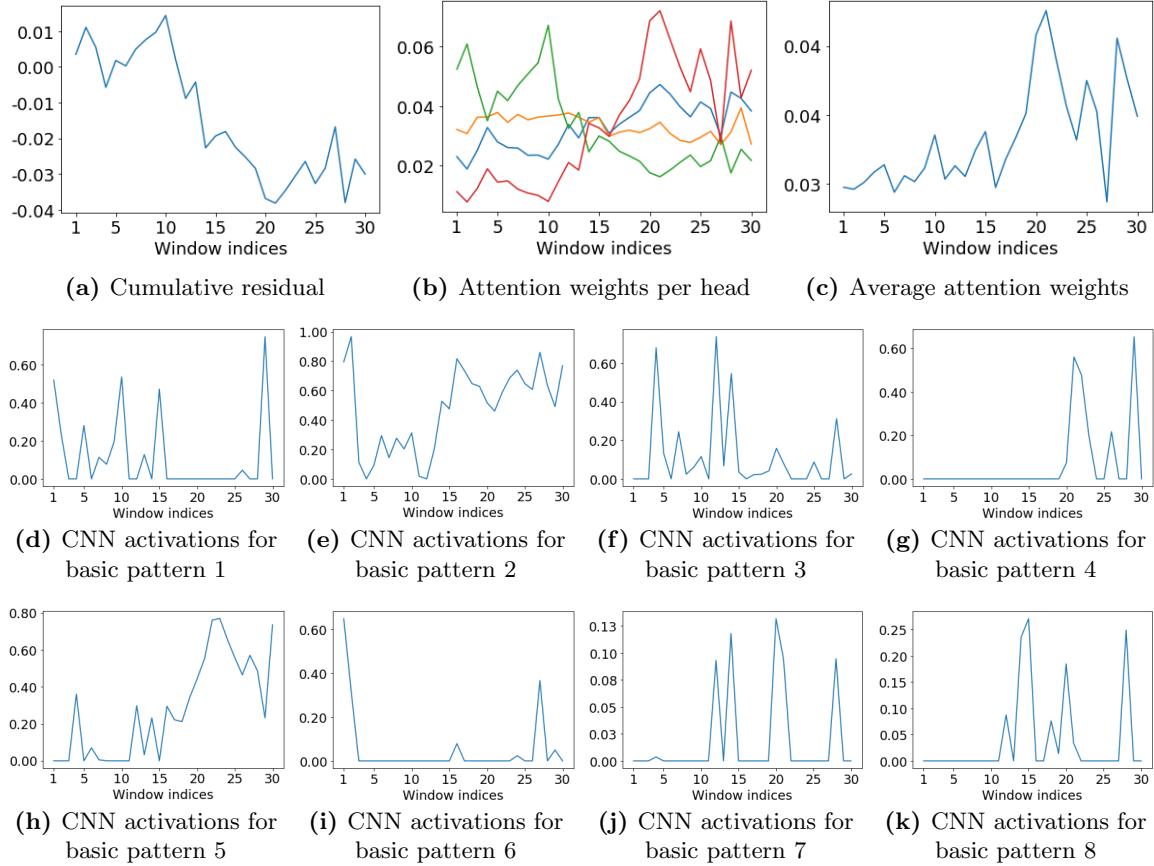
Figure 3.9 plots the attention head weights for simulated sinusoidal residual input time-series. Note that the attention head weights discover the sinusoidal pattern although the model was estimated on the empirical data and not specifically trained for this simulated input. The different attention heads capture different patterns. The fourth attention head displayed in red has the strongest activation and capture high-frequency mean reversion patterns. These attention head weights are positive for negative realizations. We will label these fourth attention head weights a “negative reversal” pattern. The third attention head weights depicted by the green curve co-move with the mean-reversion patterns of the original time-series, that is they are positive for high values, but seem to be only activated if this positive “hilltop” appears at the beginning of the time-series. If the mean-reversion cycle achieves its positive values at the end, the third attention head is not activated. We will label these third attention head weights the “early reversal” pattern. The first

Figure 3.9: Example Attention Weights for Sinusoidal Residual Inputs

These plots show the attention head weights of the CNN+Transformer benchmark model for simulated sinusoidal residual input time series. Both sine functions have a cycle of 30 days and the second is shifted by 15 days. The right subplot shows the attention weights for the $H = 4$ attention heads for the specific residuals. The empirical benchmark model is the CNN+Transformer model based on IPCA 5-factor residuals. We estimate the model on only once on the first $T_{\text{train}}=8$ years based on the Sharpe ratio objective.

attention head in blue seems to be a “dampened” version of the fourth red attention head. Figure A.3 in the Appendix shows additional simulated input time-series that confirm this interpretation. In summary, the different attention head weights can be assigned to specific global patterns.

In Figure 3.10, we plot the different components of the CNN+Transformer model evaluated on a randomly selected, representative 30-day empirical residual. The cumulative residual in subfigure (a) is the input to the CNN. This $L = 30$ dimensional vector is represented by the CNN in terms of its “exposure” to local basic pattern. The subfigures (d)–(k) show this $D \times L$ dimensional representation, which is the output of the CNN. As we have $D = 8$ local filters, we obtain eight time-series that display the activation to each filter. For example, basic pattern 1 is associated with a “reversal kink” in subfigure 3.8(a) and hence has the strongest activation to this basic pattern on day 28, when the residual has a downward spike. This $D \times L$ matrix of exposures to local patterns is the input to the transformer. The attention head weights in subfigure (b) connect the local patterns to a global pattern. The fourth attention head weight in red has its highest values during the “bottom” of the residual movements, confirming our previous intuition that this attention head activates during bad times. The third attention head weight in green spikes during the “top” at the beginning of the residual time-series, which is in line with our interpretation as an early reversal pattern. The first

Figure 3.10: CNN+Transformer Model Structure for Representative Residual

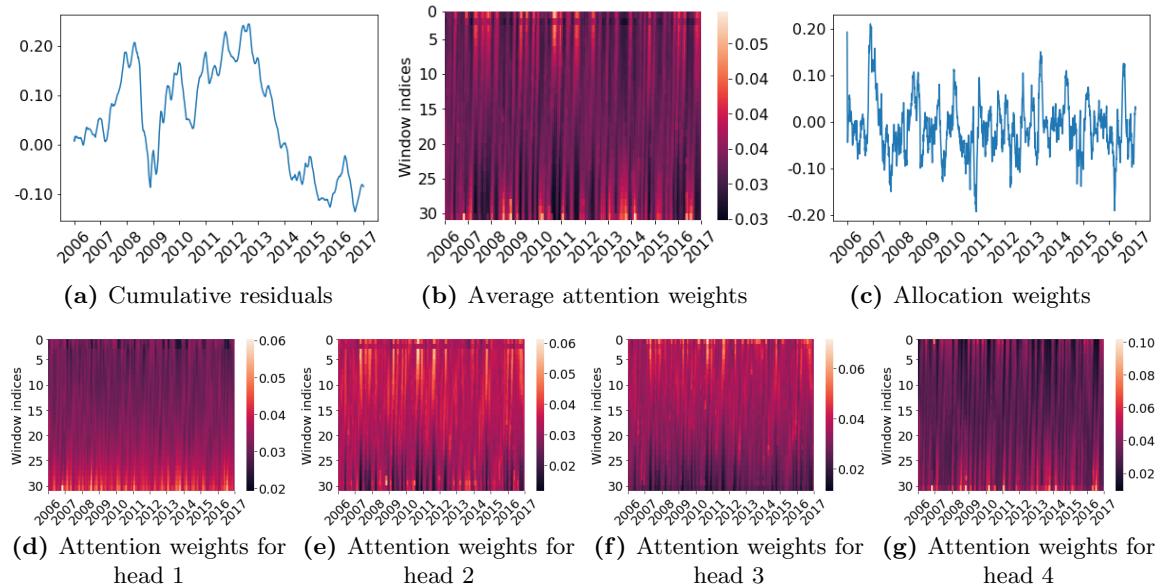
These figures illustrate the different components of the CNN+Transformer benchmark model evaluated for a randomly selected, representative empirical residual. The cumulative residual returns, which are the input to the model, are plotted in (a). The convolutional activations (d)–(k) quantify the exposure of the residual time-series to local basis filters. Subplot (b) displays the attention weights for the $H = 4$ attention heads, which represent global dependency patterns. Subplot (c) shows the average of these four attention head weights. The empirical benchmark model is the CNN+Transformer model based on IPCA 5-factor residuals. We estimate the model on only once on the first $T_{\text{train}}=8$ years based on the Sharpe ratio objective.

attention head in blue is a damped version of the red attention head. The average over the four attention head weights depicted in subplot (c) suggests that the heads attend on average more closely to the latter half of the time series.

In Figure 3.11 we generalize the analysis of Figure 3.10 to study the model structure of the benchmark model over time. While Figure 3.10 represents a “snapshot” for one point in time, we now display the allocation weights and attention head weights of a single representative residual for different times. Subfigure 3.11(a) shows the cumulative residual time-series. For a specific point in time we use the lagged $L = 30$ days as an input to obtain the allocation weights and attention head weights for that time. The out-of-sample allocation weights correctly change the

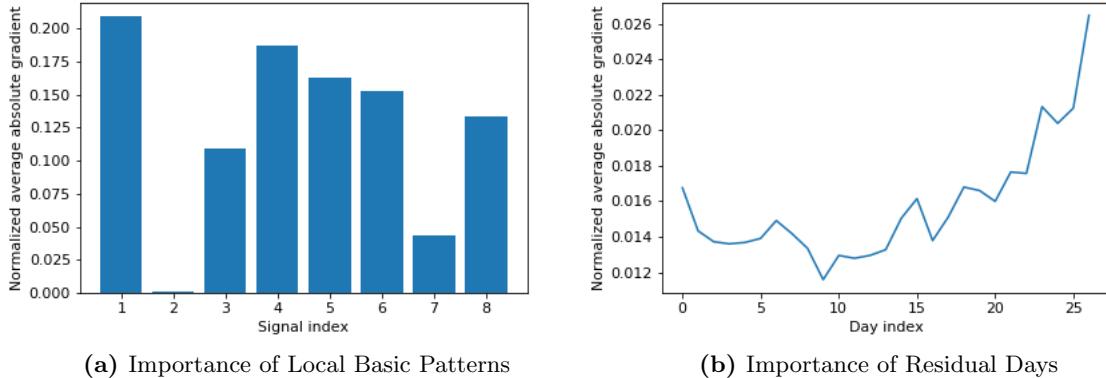
directions to exploit the patterns in the residual time-series. The attention head weights over time offer additional insights into the structure of the global patterns. Each vertical slice from window index 1 to window index 30 displays the normalized attention weights for the time point under the slice. The third attention head, which was displayed in green in Figures 3.9 and 3.10 has the largest values during “up-patterns” of the residual, for example for 2007, 2010 and 2012. Importantly, the attention weights focus on the early days within the 30-day window. This confirms our previous interpretation as an “early reversal” pattern. Attention head four, which was previously represented as a red line, has the highest values during down-times, such as 2009, 2014 and the middle of 2016. In contrast to attention head 3, this head focuses on the immediate past within the local window. Attention head 1, which is a damped down-version, focuses more uniformly on all the values within the local window.

Figure 3.11: CNN+Transformer Model Structure for Representative Residual Over Time



These figures illustrate the out-of-sample behavior from 2006 to 2016 of the CNN+Transformer benchmark model for a single residual time-series. The cumulative residual returns are plotted in (a), and the suggested allocation weights before cross-sectional normalization are plotted in (c). The attention head weights (d)–(g) quantify the activation for each attention head over time. Subplot (b) shows the average of these weights over the four heads for different times. All time-series have been smoothed using a simple moving average with a 30-day window for better presentation. The empirical benchmark model is the CNN+Transformer model based on IPCA 5-factor residuals. We estimate the model on only once on the first $T_{\text{train}}=8$ years based on the Sharpe ratio objective.

The average attention weights in (b) illustrate the asymmetric response of the transformer network. During uptrends, it focuses on the residual prices which are further in the past part of the window to decide which to position to take; however, during downtrends, it focuses on the most recent cumulative residual prices in the lookback window, which indicates that it is taking into account the latest data in order to decide what position to take. This indicates that our CNN+Transformer

Figure 3.12: Variable Importance for Allocation Weight

These figures show the normalized average absolute gradient (NAAG) of the allocation weight with respect to various inputs to intermediate layers in the CNN+Transformer benchmark network. A higher NAAG indicates a higher importance. Subplot (a) quantifies the importance of the $D = 8$ different convolutional filters, that is, we display the gradient with respect to the output of the convolutional network, which is the input to the self-attention layer. In (b), we report the importance of the first 27 days of the input residual time series. Each average absolute gradient is normalized by dividing each element by the sum of all elements. The empirical benchmark model is the CNN+Transformer model based on IPCA 5-factor residuals. We estimate the model on only once on the first $T_{\text{train}}=8$ years based on the Sharpe ratio objective.

policy network has learned to act swiftly during downtrends, and more slowly during uptrends. This shows that our model learns in particular the commonly repeated wisdom that “markets take escalators up and elevators down”. This asymmetric policy is a key benefit of the attention-based model, which cannot easily be replicated by the parametric Ornstein-Uhlenbeck or fixed basis pattern benchmark models we compare against. The convolutional subnetwork’s patterns provide translation invariant information about what kind of trend is present within each 3-day subwindow of the 30-day cumulative residual price lookback window, which allows the transformer subnetwork to form a stable attention function that results in this unique policy.

Figure 3.12 sheds further light on which days and patterns are important. The figure shows the normalized average absolute gradient (NAAG) of the allocation weight with respect to various inputs to intermediate layers in the CNN+Transformer benchmark network. A higher NAAG indicates a higher importance. Subplot (a) quantifies the importance of the $D = 8$ different basic patterns. We observe that the flat basic pattern 2 has a negligible weight, while basis patterns that are needed for trend or reversal patterns have high importance. In (b), we report the importance of the first 27 days of the input residual time series.¹⁵ Crucially, all previous days matter, which emphasizes that the trading allocation depends on the past dynamics. The most recent 14 days seem to get on average more attention for the trading decisions. However, as indicated in Figure 3.11, the importance of the days seems to be asymmetric for different global patterns.

¹⁵As the attention head weights are determined relative to the last local 3-day window, that subwindow has mechanically a larger weight and is not comparable to the other 27 days.

3.3.11 Market Frictions and Transaction Costs

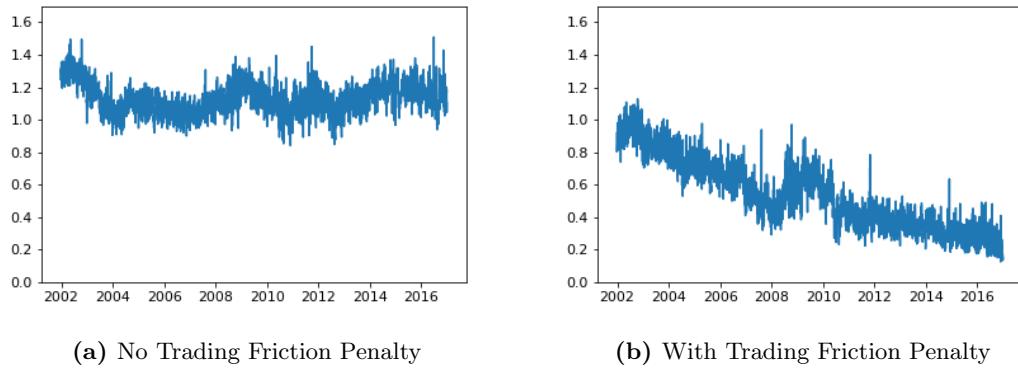
Our deep learning arbitrage strategies remain profitable in the presence of realistic trading frictions. In practice, trading costs associated with high turnover or large short-selling positions can diminish the profitability of arbitrage trading. In order to ensure that our model produces economically meaningful results, we extend it to the setting in which both transaction costs and holding costs are accounted for. We do not model market frictions associated with market impact, as in our empirical analysis we restrict the asset universe to stocks with large market capitalization, which are especially liquid.

Table 3.11: OOS Performance of CNN+Trans with Trading Frictions

K	IPCA factor model					
	Sharpe ratio			Mean-variance		
	SR	μ	σ	SR	μ	σ
0	0.52	8.5%	16.3%	0.22	2.6%	11.9%
1	0.85	5.9%	6.9%	0.86	5.5%	6.4%
3	1.24	6.6%	5.4%	1.16	6.9%	5.9%
5	1.11	5.5%	5.0%	1.02	5.3%	5.3%
10	0.98	5.1%	5.2%	1.04	5.4%	5.2%
15	0.94	4.8%	5.1%	1.02	5.1%	5.0%

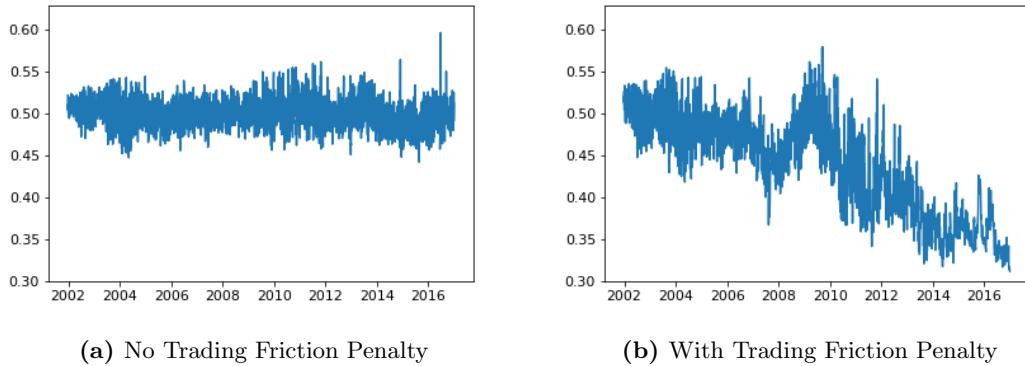
This table shows the out-of-sample annualized Sharpe ratio (SR), mean return (μ), and volatility (σ) for the CNN+Transformer model with trading frictions on IPCA residuals. We use the daily out-of-sample residuals from January 1998 to December 2016 and evaluate the out-of-sample arbitrage trading from January 2002 to December 2016. The models are calibrated on a rolling window of four years and use either the Sharpe ratio or mean-variance objective function with trading costs ($\text{cost}(w_{t-1}^R, w_{t-2}^R) = 0.0005\|w_{t-1}^R - w_{t-2}^R\|_{L^1} + 0.0001\|\min(w_{t-1}^R, 0)\|_{L^1}$). The signals are extracted from a rolling window of $L = 30$ days.

Figure 3.13: Turnover of CNN+Transformer Model with and without Trading Friction Objective



These figures show the daily turnover of CNN+Transformer model with and without trading friction objective on the representative IPCA 5-factor residuals for the out-of-sample trading period between January 2002 and December 2016. The models are calibrated on a rolling window of four years and use the Sharpe ratio objective function with or without trading costs ($\text{cost}(w_{t-1}^R, w_{t-2}^R) = 0.0005\|w_{t-1}^R - w_{t-2}^R\|_{L^1} + 0.0001\|\min(w_{t-1}^R, 0)\|_{L^1}$). We define turnover as the ℓ_1 norm of the difference between allocation weight vectors at consecutive times, i.e. $\|w_{t-1}^R - w_{t-2}^R\|_{L^1}$.

Figure 3.14: Proportion of Short Allocation Weights of CNN+Transformer Model with and without Trading Friction Objective



These figures show the daily fraction of short trades of the CNN+Transformer strategies with and without trading friction objective on the representative IPCA 5-factor residuals for the out-of-sample trading period between January 2002 and December 2016. Each plot shows the absolute value of the sum of negative weights $\|\min(w_{t-1}^R, 0)\|_{L^1}$ relative to the sum of absolute values of all weights, which is normalized to $\|w_{t-1}^R\|_1 = 1$. The models are calibrated on a rolling window of four years and use the Sharpe ratio objective function with or without trading costs ($\text{cost}(w_{t-1}^R, w_{t-2}^R) = 0.0005\|w_{t-1}^R - w_{t-2}^R\|_{L^1} + 0.0001\|\min(w_{t-1}^R, 0)\|_{L^1}$).

In our market-friction extension, the daily returns R_t of the strategy now have constant linear penalties associated with the daily turnover and the proportion of short trades. These penalties quantify proportional transaction costs, which are used to model trading fees, size of the bid-ask spread, etc., and holding costs, which are used to model short borrow rate fees charged by a brokerage. In particular, we incorporate a subset of the market friction models proposed by [8], which are commonly used in the statistical arbitrage literature.¹⁶ Mathematically, we subtract the market-friction costs

$$\text{cost}(w_{t-1}^R, w_{t-2}^R) = 0.0005\|w_{t-1}^R - w_{t-2}^R\|_{L^1} + 0.0001\|\min(w_{t-1}^R, 0)\|_{L^1}$$

from the portfolio returns and use these net portfolio returns in the optimization problem of section 3.2.3, where $w_{t-1}^R \in \mathbb{R}^{N_{t-1}}$ is the strategy's allocation weight vector at time $t-1$. The first penalty term represents a slippage/transaction cost of 5 basis points per transaction, whereas the second one is a holding cost of 1 basis point per short position. Both costs are universal for all times and all stocks. This corresponds to a modification of the objective function in the training and evaluation parts of our algorithm. We use this model for the sake of illustration and simplicity given that in our empirical study we trade a universe of highly liquid US stocks, but more complicated models¹⁷ may be included in the computations without any significant structural changes.

Table 3.11 displays the Sharpe ratios, average returns, and volatility of our CNN+Transformer

¹⁶See for example [2], [83] and [47].

¹⁷For example, those considering time and stock-dependent transaction costs or market impact of the trades on the stock prices.

model under market frictions for IPCA residuals. The results for PCA residuals are collected in the Appendix in Table 3.11 with very similar findings. We exclude the Fama-French factor model from the analysis with market frictions, as we take the traded factors from Kenneth French Data Library as given, which are based on a larger stock universe with different trading costs and, hence, would not be directly comparable to the IPCA and PCA results.¹⁸ As expected the Sharpe ratios are lower and range from 0.94 to 1.24 for a reasonable number of IPCA factors. The Sharpe ratio and mean-variance objective have the desired effects, but lead to overall very similar results. Importantly, the arbitrage strategies retain their economic significance even in the presence of trading costs.

These results present a lower bound on the profitability under trading frictions, as we have made four simplifying assumptions. First, in the current implementation the factor composition cannot be changed due to trading costs. A possible extension could construct the latent risk factors by including the trading friction objective. For example, the sparse representation of latent factors as in Pelger and Xiong (2021) would reduce trading costs. Second, because the policy with frictions is recursive, we are conducting an approximate training process to maintain parallelization given our computational resources and the large volume of data, but this may lead to suboptimal optimization results. However, it would be possible to conduct an exact sequential training process at the cost of more computation. Third, our modified architecture with the market-friction objective is given by the simplest modification to our architecture without frictions, but it is possible that the optimal transaction and holding cost-minimizing strategy has a more complicated functional form or is not Markovian and requires additional previous allocations. Last but not least, we keep the hyperparameters of our main analysis, but we could potentially improve the performance by employing hyperparameter tuning.

The effect of trading frictions is time-varying and our model can exploit particularly profitable arbitrage time periods by increasing trading and short positions. In Figure 3.13 we analyze the daily turnover of a representative CNN+Transformer strategy based on IPCA 5-factor residuals and a Sharpe ratio objective. Broadly, we see that our model with trading friction penalty is able to adapt by decreasing daily turnover. However, our model seems to reduce turnover based on trading opportunities. During the times of high market volatility such as 2007–2009, arbitrage trading could be potentially be more profitable, which our model takes advantage of. On the other hand, during the later years of the calm bull market from 2011–2015, strategies with less turnover could maintain profitability. This pattern is confirmed in Figure 3.14 which shows the daily proportion of allocation weights, which are short stocks in our universe. As expected the holding cost friction model reduces the overall proportion of short trades. Interestingly, our model is able to intelligently choose time periods during which it can maximize performance by taking positions with higher short proportion,

¹⁸Regardless, each factor corresponds to a portfolio of traded assets, and thus the residuals of this model could be traded in a number of a number of ways under suitable extensions. For example, we could include ETFs which try to track a value or size premium, project these latent factors onto our asset universe, or approximate each factor with a number of sparse subset of assets in our asset universe as in [72]. However, these changes constitute differences that would make the results incomparable to the PCA and IPCA results.

such as the market turmoil at the end of 2015 and the financial crisis of 2008. Effectively, this indicates that the CNN+Transformer trading policy has learned to avoid holding and transaction costs by generally modifying the original strategy’s allocations to be less short-biased on average, and to more appropriately enter short-dominant positions during relevant subperiods.

3.4 Conclusion

In this chapter, we introduce a unifying conceptual framework to compare different statistical arbitrage approaches based on the decomposition into (1) arbitrage portfolio generation, (2) signal extraction and (3) allocation decision. We develop a novel deep learning statistical arbitrage approach. It uses conditional latent factors to generate arbitrage portfolios. The signal is estimated with a CNN+Transformer, which combines global dependency patterns with local filters. The allocation is estimated with a nonparametric FFN based on a global trading objective.

We conduct a comprehensive empirical out-of-sample study on U.S. equities and demonstrate the potential of machine learning methods in arbitrage trading. Our CNN+Transformer substantially outperforms all benchmark approaches. The implied trading strategies are not spanned by conventional risk factors, including price trend factors, and survive realistic transaction and holding costs. Our model provides insights into optimal trading policies which are based on asymmetric trend and reversion patterns. In particular, our study shows that the trading signal extraction is the most challenging and separating element among different statistical arbitrage approaches.

Our findings contribute to the debate on efficiency of markets. We quantify the scope of profits that arbitrageurs can achieve in equity markets. Importantly, the substantial profitability of our arbitrage strategies is not inconsistent with equilibrium asset pricing, following similar arguments as in [32]. It could rather be viewed as empirical evidence about how efficiency is maintained in practice. We document non-declining profitability of arbitrage trading over time, which suggests that the profits are compensation for arbitrageurs to enforce the law of one price. Our findings also suggest that unconditional means of asset pricing residuals as a measure of alpha might not correctly reflect the amount of arbitrage left in financial markets.

Appendix A

Appendix for Chapter 3

A.1 Data

A.1.1 List of the Firm-Specific Characteristics

Please see Table A.1 on the next page.

A.2 Implementation of Different Models

A.2.1 Feedforward Neural Network (FFN)

In the Fourier+FFN and FFN models, we utilize a feedforward network with L^{FFN} layers as illustrated in Figure A.1. Each hidden layer takes the output from the previous layer and transforms it into an output as

$$x^{(l)} = \text{ReLU} \left(W^{\text{FFN},(l-1)\top} x^{(l-1)} + w_0^{\text{FFN},(l-1)} \right) = \text{ReLU} \left(w_0^{\text{FFN},(l-1)} + \sum_{k=1}^{K^{(l-1)}} w_k^{\text{FFN},(l-1)} x_k^{(l-1)} \right)$$
$$y = W^{\text{FFN},(L^{\text{FFN}})\top} x^{(L)} + w_0^{\text{FFN},(L^{\text{FFN}})}$$

with hidden layer outputs $x^{(l)} = (x_1^{(l)}, \dots, x_{K^{(l)}}^{(l)}) \in \mathbb{R}^{K^{(l)}}$, parameters $W^{(\text{FFN},l)} = (w_1^{\text{FFN},(l)}, \dots, w_{K^{(l)}}^{\text{FFN},(l)}) \in \mathbb{R}^{K^{(l)} \times K^{(l-1)}}$ for $l = 0, \dots, L^{\text{FFN}} - 1$ and $W^{\text{FFN},(L^{\text{FFN}})} \in \mathbb{R}^{K^{(L)}}$, and where $\text{ReLU}(x_k) = \max(x_k, 0)$.

A.2.2 Ornstein-Uhlenbeck Model

Following [2] and [83] we model X_t as an Ornstein-Uhlenbeck (OU) process

$$dX_t = \kappa (\mu - X_t) dt + \sigma dB_t$$

Table A.1: Firm Characteristics by Category

Past Returns			Value
(1)	r2_1	Short-term momentum	(26) A2ME
(2)	r12_2	Momentum	(27) BEME
(3)	r12_7	Intermediate momentum	(28) C
(4)	r36_13	Long-term momentum	(29) CF
(5)	ST.Rev	Short-term reversal	(30) CF2P
(6)	LT.Rev	Long-term reversal	(31) D2P (32) E2P
Investment			(33) Q
(7)	Investment	Investment	(34) S2P
(8)	NOA	Net operating assets	(35) Lev
(9)	DPI2A	Change in property, plants, and equipment	
(10)	NI	Net Share Issues	
Profitability			Trading Frictions
(11)	PROF	Profitability	(36) AT
(12)	ATO	Net sales over lagged net operating assets	(37) Beta
(13)	CTO	Capital turnover	(38) IdioVol
(14)	FC2Y	Fixed costs to sales	(39) LME
(15)	OP	Operating profitability	(40) LTurnover
(16)	PM	Profit margin	(41) MktBeta
(17)	RNA	Return on net operating assets	(42) Rel2High
(18)	ROA	Return on assets	(43) Resid_Var
(19)	ROE	Return on equity	(44) Spread
(20)	SGA2S	Selling, general and administrative expenses to sales	(45) SUV
(21)	D2A	Capital intensity	(46) Variance
Intangibles			
(22)	AC	Accrual	
(23)	OA	Operating accruals	
(24)	OL	Operating leverage	
(25)	PCM	Price to cost margin	

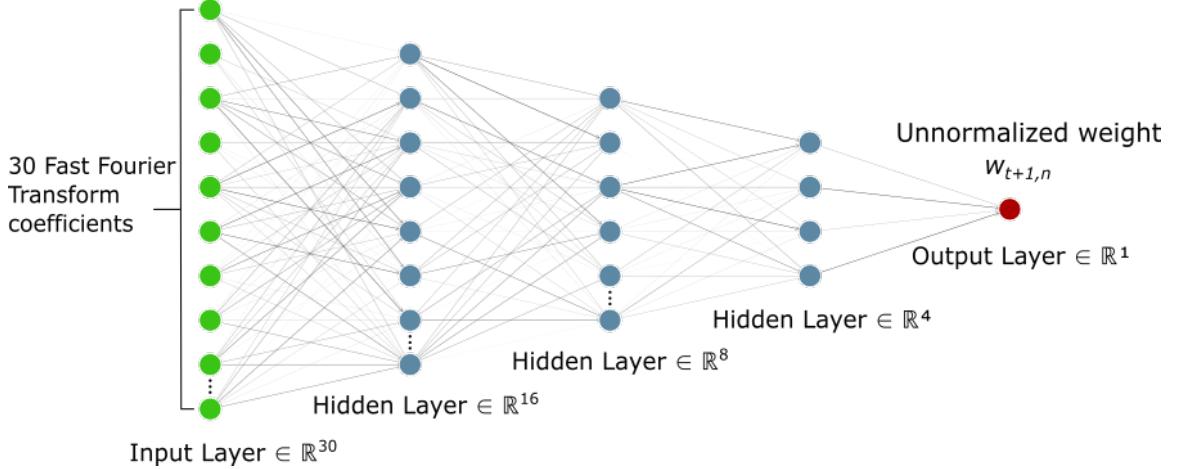
This table shows the 46 firm-specific characteristics sorted into six categories. More details on the construction are in the Internet Appendix of [15].

for a Brownian motion B_t . As the analytical solution of the above stochastic differential equation is

$$X_{t+\Delta t} = (1 - e^{-\kappa \Delta t})\mu + e^{-\kappa \Delta t} X_t + \sigma \int_t^{t+\Delta t} e^{-\kappa(t+\Delta t-s)} dB_s$$

for any Δt , we can without loss of generality set $\Delta t = 1$, and estimate the parameters κ, μ and σ from the AR(1) model

$$X_{t+1} = a + bX_t + e_t,$$

Figure A.1: Feedforward Network Architecture

where each e_t is a normal, independent and identically distributed random variable with mean 0. The parameters are estimated with a standard linear regression, which yields

$$\hat{\kappa} = -\frac{\log(\hat{b})}{\Delta t}, \quad \hat{\mu} = \frac{\hat{a}}{1 - \hat{b}}, \quad \frac{\hat{\sigma}}{\sqrt{2\hat{\kappa}}} = \sqrt{\frac{\hat{\sigma}_e^2}{1 - \hat{b}^2}}.$$

The strategy depends on the ratio $\frac{X_L - \hat{\mu}}{\hat{\sigma}\sqrt{2/\hat{\kappa}}}$. Note that this is only defined for $b < 1$ which is equivalent to parameter restrictions that the OU process is mean-reverting. The trading policy depends on the thresholds c_{thresh} and c_{crit} , which are hyperparameters. These hyperparameters are selected on the validation data from the candidate values $c_{\text{thresh}} \in \{1, 1.25, 1.5\}$ and $c_{\text{crit}} \in \{0.25, 0.5, 0.75\}$. Our benchmark model has the values $c_{\text{thresh}} = 1.25$ and $c_{\text{crit}} = 0.25$, which coincides with the optimal values in [2] and [83].

A.2.3 Convolutional Neural Network with Transformer

Convolutional Neural Network

In our empirical application, we consider a 2-layered convolutional network with some standard technical additions. The network takes as input a window $x^{(0)} = x \in \mathbb{R}^L$ of L consecutive daily cumulative returns or log prices of a residual, and outputs the feature matrix $\tilde{x} \in \mathbb{R}^{L \times F}$ given by computing the following quantities for $l = 1, \dots, L, d = 1, \dots, D$

$$y_{l,d}^{(0)} = b_d^{(0)} + \sum_{m=1}^{D_{\text{size}}} W_{d,m}^{(0)} x_{l-m+1}^{(0)}, \quad x_{l,d}^{(1)} = \text{ReLU} \left(\frac{y_{1,d}^{(0)} - \mu_d^{(0)}}{\sigma_d^{(0)}} \right). \quad (\text{A.1})$$

$$y_{l,d}^{(1)} = b_d^{(1)} + \sum_{m=1}^{D_{\text{size}}} \sum_{j=1}^D W_{d,j,m}^{(1)} x_{l-m+1,j}^{(1)}, \quad x_{l,d}^{(2)} = \text{ReLU} \left(\frac{y_{l,d}^{(1)} - \mu_d^{(1)}}{\sigma_d^{(1)}} \right), \quad (\text{A.2})$$

$$\tilde{x}_{l,d} = x_{l,d}^{(2)} + x_l^{(0)}, \quad (\text{A.3})$$

where

$$\mu_k^{(i)} = \frac{1}{L} \sum_{l=1}^L y_{l,k}^{(i)}, \quad \sigma_k^{(i)} = \sqrt{\frac{1}{L} \sum_{l=1}^L (y_{l,k}^{(i)} - \mu_k^{(i)})^2}.$$

and $b^{(0)}, b^{(1)} \in \mathbb{R}^D$, $W^{(0)} \in \mathbb{R}^{D \times D_{\text{size}}}$ and $W^{(1)} \in \mathbb{R}^{D \times D \times D_{\text{size}}}$ are parameters to be estimated. Compared with the simple convolutional network introduced in the main text, the previous equations incorporate three standard technical improvements commonly used in deep learning practice. First, they include “bias terms” $b^{(i)}$ in the first part of equations A.1 and A.2 to allow for more flexible modeling. Second, they include so-called “instance normalization” before each activation function to speed up the optimization and avoid vanishing gradients caused by the saturation of the ReLU activations. Third, they include a “residual connection” in equation A.3 to facilitate gradient propagation during training.

Transformer Network

The benchmark model in our empirical application is a one-layer transformer, following the implementation of the seminal paper of [79]. First, the sequence of features $\tilde{x} \in \mathbb{R}^{L \times D}$ is projected onto D/H -dimensional subspaces (called the “attention heads”) for an integer H dividing D , obtaining, for $1 \leq i \leq H$,

$$V_i = \tilde{x}W_i^V + b_i^V \in \mathbb{R}^{L \times D/H}, \quad K_i = \tilde{x}W_i^K + b_i^K \in \mathbb{R}^{L \times D/H}, \quad Q_i = \tilde{x}W_i^Q + b_i^Q \in \mathbb{R}^{L \times D/H},$$

where $W_i^V, W_i^K, W_i^Q \in \mathbb{R}^{D \times F/H}$, $b_i^V, b_i^K, b_i^Q \in \mathbb{R}^{D/H}$ are parameters to be estimated. Next, each projection V_i is processed temporally obtaining the hidden states $h_i \in \mathbb{R}^{L \times D/H}$, with

$$h_{i,l} = \sum_{j=1}^L w_{l,j,i} V_{i,j} \in \mathbb{R}^{D/H}, \quad w_{l,j,i} = \frac{\exp(K_{i,l} \cdot Q_{i,j})}{\sum_{m=1}^L \exp(K_{i,l} \cdot Q_{i,m})} \in [0, 1].$$

These states are then concatenated and linearly combined, obtaining the last hidden state

$$h = \text{Concat}(h_1, \dots, h_h)W^O + b^O \in \mathbb{R}^{L \times D},$$

where $W^O \in \mathbb{R}^{F \times F}$, $b^O \in \mathbb{R}^F$ are parameters to be estimated.

Finally, h is normalized and processed time-wise though a 2-layered feedforward network as described in detail in the original paper ([79]). The number of hidden units in the intermediate layer is a technical hyperparameter that we call HDN in section A.3.1. This network also has dropout

regularization with hyperparameter called DRP in section A.3.1.

A.2.4 Network Estimation Details

As explained in Section 3.2.3, we estimate the parameters of the models with neural networks by solving the optimization problems introduced in equation (3.4) or in equation (3.6) of section 3.2.3, depending on the model and the objective function. In all cases, this is done by replacing the mean and variance by their annualized sample counterpart over a training set, and by finding the optimal network parameters with stochastic gradient descent using PyTorch’s Adam optimizer and the optimization hyperparameters learning rate and number of optimization epochs described in detail in section A.3.1.

As mentioned in Section 3.3.3, our main results use rolling windows of 1,000 days as training sets. The networks are reestimated every 125 days to strike a balance between computational efficiency and adaptation to changing economic conditions, and the strategies’ returns are always obtained out-of-sample. Additionally, to be able to train our model over these long windows without running into memory issues, we split each training window into temporal “batches”, as is commonly done in deep learning applications. Each batch contains the returns and residuals for all the stocks in a subwindow of 125 days of the original training window, with the subwindows being consecutive and non-overlapping (i.e., for a training window of 1000 days, we split it into the subwindow containing the first 125 days, the subwindow containing the days between the 126th day and the 250th day, etc.). The optimization process is applied successively to each batch, completing the full sequence of batches before starting a new optimization iteration or epoch.

In the implementation of our optimization procedure under market frictions, we found it useful to include the last allocation as an additional input to the allocation function \mathbf{w}^ϵ , as the inclusion of the cost term makes the objective function depend on it. However, the inclusion of the previous allocations in either the objective function or the architecture of the model complicates the parallelization of the training and evaluation computations, because after this change the model requires the output of the previous lookback window in order to compute the output of the current window. To allow training to remain parallelized, which is desirable for reasonable computational speed given the volume of data of our empirical application, in our implementation of the training function in each epoch, we take the previous allocations from the output of the previous epoch and use them as a pre-computed approximation of the allocations for the current epoch. This approximation converges in our empirical experiments and allows us to maintain parallelization, but may produce suboptimal results. For evaluation purposes, however, everything is computed exactly and with no approximations using a sequential approach.

Throughout section 3.3, all presented results have been computed with PyTorch 1.5 and have been parallelized across 8 NVIDIA GeForce GTX Titan V GPUs, on a server with two Intel Xeon E5-2698v3 32-core CPUs and 1 TB of RAM. The full rationale for the hyperparameter choices are

described in detail in section A.3.1, but for a CNN+Transformer model with a lookback window of 30 days, 8 convolutional filters with a filter size of 2, 4 attention heads, 125-day reestimation using a rolling lookback window of 1000, it takes our deep model approximately 7 hours to be periodically estimated and run in our 19 years of daily out-of-sample data with our universe of on average \sim 550 stocks per month.

A.3 Additional Empirical Results

A.3.1 Robustness to Hyperparameter Selection

In this subsection, we describe our hyperparameter selection procedure and explore additional hyperparameter choices to show that the performance of our strategies is extremely robust to our choices. These results complement the time stability checks we exhibited in Section 3.3.9. To decide which hyperparameters we would select for use in our network, we fixed a validation dataset as follows: we took the first 1000 trading days of our data set of residuals (all trading days from January 1, 1998 through December 31, 2001) of the 5-factor IPCA-based model, which is estimated with a 20-year rolling window. Because it is solely used for training in our rolling train/test procedures used to compute strategy returns, this data is completely in-sample, and thus completely avoids look-ahead bias which would influence any of our out-of-sample trading results in the main text. We started with a reasonable set for our hyperparameters, and tested also additional points adjacent to these sets.¹ For each model represented by a point on the grid, we trained the model using the Sharpe ratio objective on the first 750 days of the 1000 trading days, and evaluated it by its out-of-sample Sharpe ratio on the last 250 days of the 1000 trading days. We tested 16 combinations of hyperparameters, which are illustrated in Table A.2. The results of our test on the last 250 days of our validation data are displayed in Table A.3.

The results in Table A.3 show that all Sharpe ratios fall within a tight range of values, which is roughly [3.5, 4.2]. Means and volatilities concentrate similarly, falling within [13%, 17.8%] and [3.6%, 4.3%]. Computation of 95% bootstrapped confidence intervals for mean return shows that all models' confidence intervals contain the interval [10%, 20%], with volatilities similarly contained. Hence, these models are statistically not distinguishable. Given the statistical insignificance of the differences in performance of these models, we chose the model displayed in Table A.2, which is the most parsimonious one, that is it has the smallest number of parameters, and hence benefits low GPU memory usage and ease of interpretability.

To ensure that our results are stable across several choices of hyperparameters, we study the results of four additional models with perturbed hyperparameters. This complements our robustness results of Section 3.3.9 regarding the size of the lookback window and the retraining frequency. The

¹Note the computation over a large set of hyperparameters is computationally infeasible, which requires us to restrict the set to reasonable values.

Table A.2: Hyperparameter options for the network in the empirical analysis

Notation	Hyperparameters	Candidates	Chosen
D	Number of filters in the convolutional network	8, 16	8
ATT	Number of attention heads	2, 4	4
HDN	Number of hidden units in the transformer's linear layer	2D, 3D	2D
DRP	Dropout rate in the transformer	0.25, 0.5	0.25
D_{size}	Filter size in the convolutional network	2	2
LKB	Number of days in the residual lookback window	30	30
WDW	Number of days in the rolling training window	1000	1000
RTFQ	Number of days of the retraining frequency	125	125
BTCH	Batch size, in days	125	125
LR	Learning rate	0.001	0.001
EPCH	Number of optimization epochs	100	100
OPT	Optimization method	Adam	Adam

This table shows the parameters for our network architecture with respect to the Sharpe ratio on our validation data and the candidates we tried. In DRP, we follow the convention that the dropout rate p is the proportion of units which are removed.

Table A.3: Performance of candidate models on the last year of the validation data set

D	ATT	HDN	DRP	SR	μ	σ
8	2	2	0.25	3.81	16.3%	4.3%
8	2	2	0.50	3.92	16.0%	4.1%
8	2	3	0.25	3.79	16.2%	4.3%
8	2	3	0.50	4.00	16.4%	4.1%
8	4	2	0.25	3.81	15.6%	4.1%
8	4	2	0.50	4.13	17.8%	4.3%
8	4	3	0.25	3.82	15.6%	4.1%
8	4	3	0.50	4.16	17.4%	4.2%
16	2	2	0.25	4.00	14.8%	3.7%
16	2	2	0.50	4.06	16.2%	4.0%
16	2	3	0.25	4.11	14.9%	3.6%
16	2	3	0.50	4.06	16.6%	4.1%
16	4	2	0.25	3.93	15.6%	4.0%
16	4	2	0.50	3.66	13.9%	3.8%
16	4	3	0.25	4.18	16.8%	4.0%
16	4	3	0.50	3.51	13.0%	3.7%

This table shows the model performance with respect to the Sharpe ratio, mean, and volatility on our validation data set for the candidate models implied by Table A.2. The models are trained on the first three years of the validation data set (1998–2000) and tested on the last year (2001). In DRP, we follow the convention that the dropout rate p is the proportion of units which are removed.

four additional networks and their hyperparameter configurations are listed in Table A.4, with Network 1 being the network studied throughout this empirical section. Network 2 corresponds to more filters and commensurately more hidden units to consume them, and higher dropout rates to more strongly regularize the additional parameters. Network 3 halves the number of attention heads from our original specification. Networks 4 and 5 modify the size of the rolling training window from 1000 trading days to 1250 and 750 trading days, respectively, which corresponds closely to three and

five calendar years. These additional hyperparameter configurations constitute local perturbations in hyperparameter space, to which our strategies' performance should be relatively robust.

Table A.4: Alternative best performing models on the data from 2002–2016

Model	FLNB	FLSZ	ATT	HDN	DRP	LKB	WDW
Network 1	[1,8]	2	4	16	0.25	30	1000
Network 2	[1,16]	2	4	32	0.5	30	1000
Network 3	[1,8]	2	2	16	0.25	30	1000
Network 4	[1,8]	2	4	16	0.25	30	1250
Network 5	[1,8]	2	4	16	0.25	30	750

This table reports four of the best performing models for our network architecture with respect to the Sharpe ratio on our data from 2002–2016 and the candidates described in Table A.2. Our original network, which is studied throughout this section, is labeled as Network 1.

Table A.5: Performance of the alternative models on our benchmark residual datasets, 2002–2016

Model	Fama-French 5			PCA 5			IPCA 5		
	SR	μ	σ	SR	μ	σ	SR	μ	σ
Network 1	3.21	4.6%	1.4%	3.36	14.3%	4.2%	4.16	8.7%	2.1%
Network 2	3.16	4.6%	1.4%	3.26	13.9%	4.3%	4.35	8.4%	1.9%
Network 3	3.30	4.8%	1.4%	3.17	13.4%	4.2%	4.00	8.4%	2.1%
Network 4	2.93	4.1%	1.4%	2.74	11.7%	4.3%	3.96	7.9%	2.0%
Network 5	3.13	4.9%	1.6%	3.52	15.0%	4.3%	3.77	8.6%	2.3%

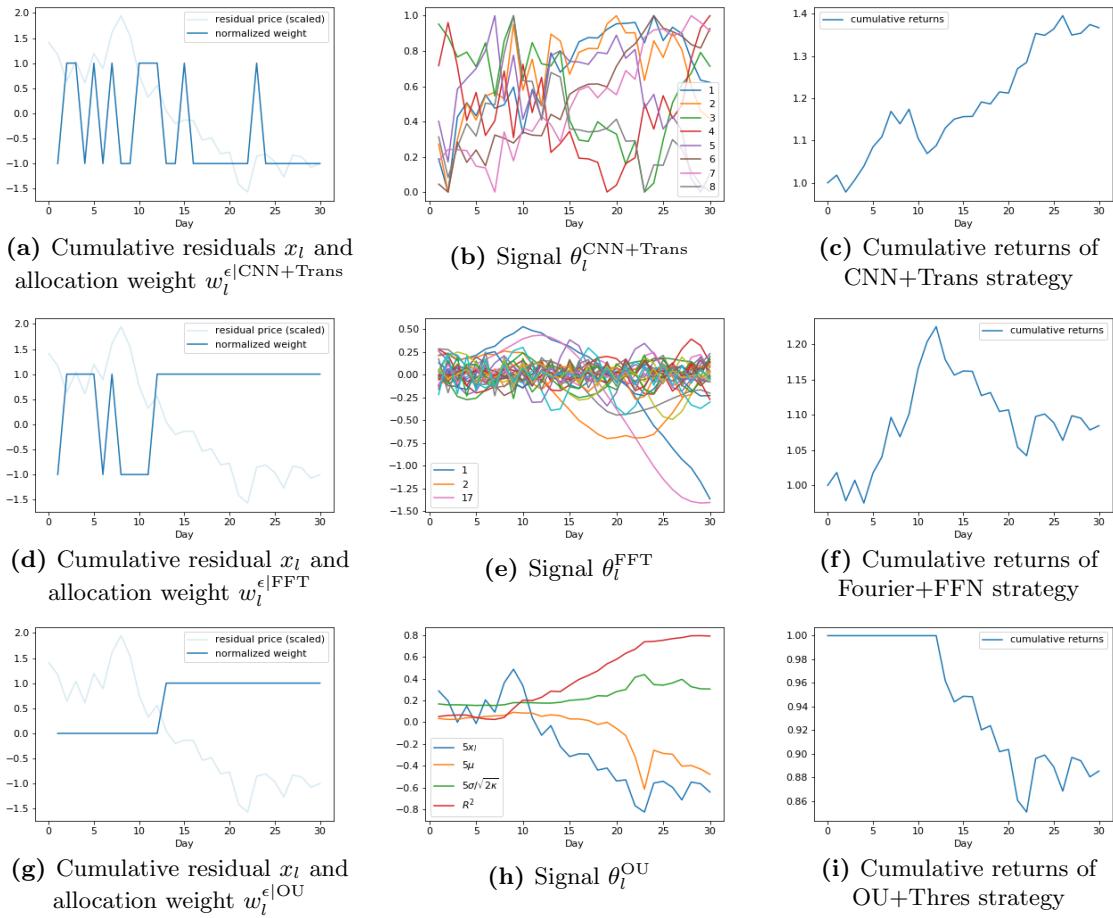
This table shows the average annualized returns, volatilities and Sharpe ratios of our alternative models from Table A.4 on our three benchmark residual datasets, trained with the Sharpe ratio objective function.

In Table A.5, we report the results of these models on a representative subset of 5-factor models, which are now evaluated on the full out-of-sample data. We see that the Sharpe ratios are broadly similar across all three different perturbations of network architecture hyperparameters (i.e., number of filters, number of attention heads, and dropout rate). The small range of values induced by these choices shows that our network performs similarly over a variety of sensible network parameters, and highlights the efficacy of our reasonable choice of convolutional, attentional, and feedforward subnetworks which specialize in finding small temporal patterns, arranging these patterns throughout time, and deciding on allocations based on these arranged patterns. The only hyperparameter choice which can be considered to be slightly lower than the rest is the choice of using a rolling training window of 1250 trading days. This suggests that statistical arbitrage policies are time-varying to adopt to changing economic conditions. Our results show that using the latest 3 or 4 years of data for the estimation of the parameters of this model achieves an adequate balance between capturing changing conditions and using sufficient data.

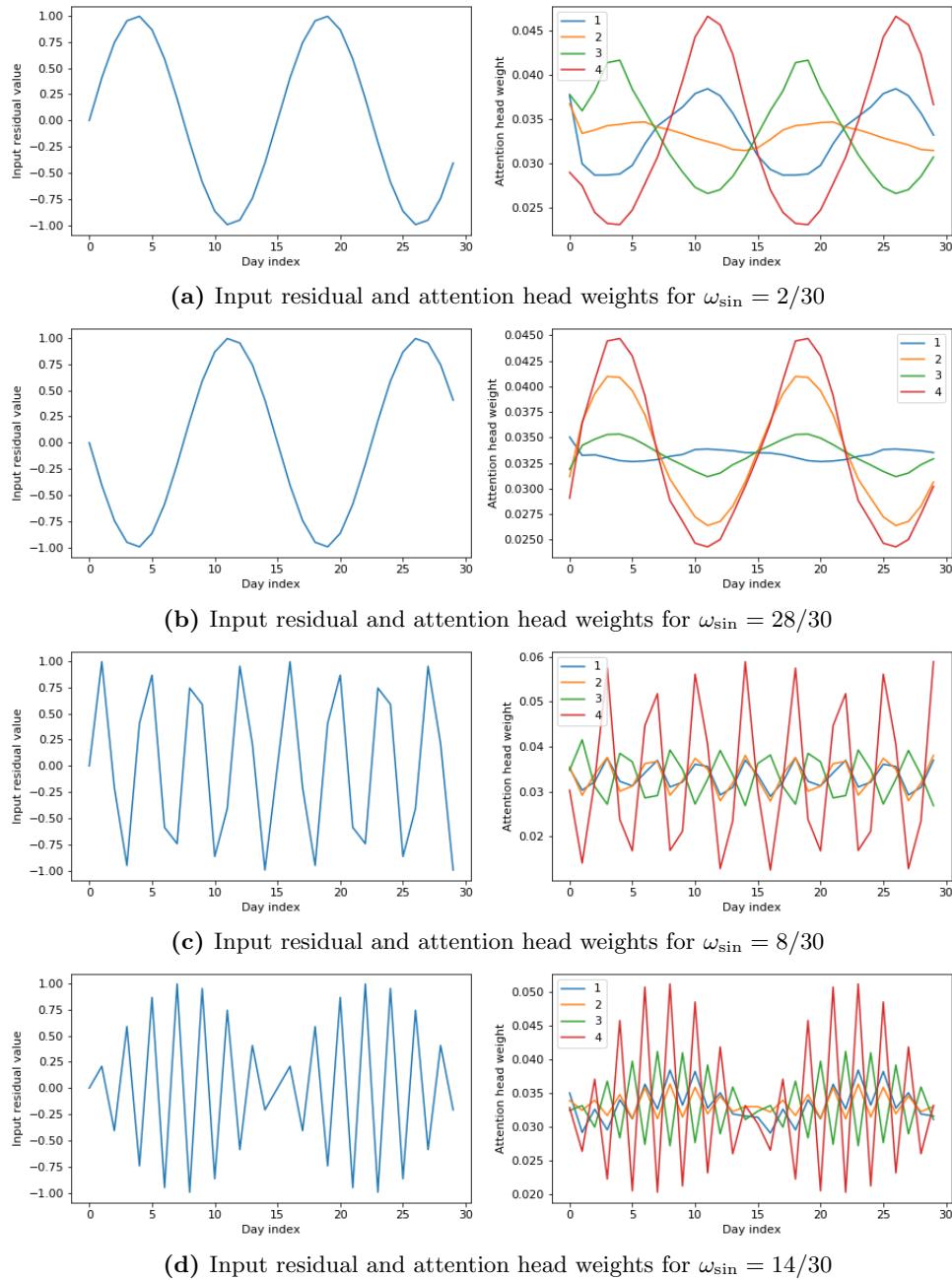
For the allocation function feedforward network (FFN) utilized for the Fourier+FFN model, we choose a reasonable architecture based on deep learning conventions and have verified that the results are robust to this choice. Because the input of the network are the $L = 30$ coefficients of the Fourier decomposition of each residual window $(X_l^{(n,t)})_{1 \leq l \leq L}$ and the output is the corresponding allocation weight $w_{n,t} \in \mathbb{R}$, we follow standard deep learning practices and consider 3 hidden layers with dimensions 16,8,4 regularized with a dropout rate of 0.25. We use the ReLU activation function, and train using the same procedure outlined in A.2.4, with the same batch size, learning rate, number of optimization epochs, and optimization method as in Table A.2.

A.3.2 Interpretation

Figure A.2: Additional Examples of Allocation Weights and Signals



These plots are an illustrative example of the allocation weights and signals of the Ornstein-Uhlenbeck with Threshold (OU+Thres), Fast Fourier Transform (FFT) with Feedforward Neural Network (FFN), and Convolutional Neural Network (CNN) with Transformer models for a specific cumulative residual. The models are estimated on the empirical data, and the residual is a representative empirical example. In more detail, we consider the residuals from five IPCA factors and estimate the benchmark models as explained in Section 3.3.10. The left subplots display the cumulative residual process along with the out-of-sample allocation weights $w_l^{\epsilon| \cdot}$ that each model assigns to this specific residual. In this example, we consider trading only this specific residual and hence normalize the weights to $\{-1, 0, 1\}$. The middle column plots show the time-series of estimated out-of-sample signals for each model, by applying the θ_l arbitrage signal function to the previous L cumulative returns of the residual. The right column plots display the out-of-sample cumulative returns of trading this particular residual based on the corresponding allocation weights. We use a rolling lookback window of $L = 30$ days to estimate the signal and allocation, which we evaluate for the out-of-sample on the next 30 days. The plots only show the out-of-sample period. The evaluation of this illustrative example is a simplification of the general model that we use in our empirical main analysis, where we trade all residuals and map them back into the original stock returns.

Figure A.3: Example Attention Weights for Sinusoidal Residual Inputs

These plots show the attention head weights of the CNN+Transformer benchmark model for simulated sinusoidal residual input time series. The inputs are $x_l = \sin(2\pi\omega_{\sin}l)$, for various ω_{\sin} and $l \in \{0, \dots, 29\}$. The right subplot shows the attention weights for the $H = 4$ attention heads for the specific residuals. The empirical benchmark model is the CNN+Transformer model based on IPCA 5-factor residuals. We estimate the model on only once on the first $T_{\text{train}}=8$ years based on the Sharpe ratio objective.

A.3.3 Dependency between Arbitrage Strategies

Table A.6: Correlations between the Returns of the CNN+Transformer Arbitrage Strategies

	Fama-French 3	PCA 3	IPCA 3	Fama-French 5	PCA 5	IPCA 5	PCA 10	IPCA 10
Fama-French 3	1.00	0.32	0.44	0.62	0.25	0.43	0.21	0.44
PCA 3	0.32	1.00	0.32	0.34	0.62	0.35	0.41	0.36
IPCA 3	0.44	0.32	1.00	0.37	0.28	0.81	0.21	0.75
Fama-French 5	0.62	0.34	0.37	1.00	0.28	0.39	0.23	0.40
PCA 5	0.25	0.62	0.28	0.28	1.00	0.29	0.47	0.31
IPCA 5	0.43	0.35	0.81	0.39	0.29	1.00	0.23	0.84
PCA 10	0.21	0.41	0.21	0.23	0.47	0.23	1.00	0.25
IPCA 10	0.44	0.36	0.75	0.40	0.31	0.84	0.25	1.00

This table reports the correlations of our CNN+Transformer strategies for some representative choices of the factor models. The correlations are calculated with returns of the out-of-sample arbitrage trading from January 2002 to December 2016. The models are calibrated on a rolling window of four years and use the Sharpe ratio objective function. The signals are extracted from a rolling window of $L = 30$ days.

A.3.4 Time-Series Signal

In this appendix, we report the OOS returns of strategies using alternative models for the ablation tests in Section 3.3. For the FFN feedforward network, we use the same architecture, hyperparameters, optimization settings, etc. as in the Fourier+FFN model utilized throughout the empirical results section and described in Appendix A.3.1. For the OU+FFN model, because the input is the low-dimensional OU signal in \mathbb{R}^4 , we consider a 3 hidden layer with dimensions 4,4,4 regularized with a dropout rate of 0.25. We use the sigmoid activation function, and estimate it using the same procedure outlined in section A.2.4, with the same batch size, learning rate, number of optimization epochs, and optimization method as in Table A.2.

Table A.7: OOS Annualized Performance Based on Sharpe Ratio Objective

Model	K	Factors			Fama-French			PCA			IPCA		
		SR	μ	σ	SR	μ	σ	SR	μ	σ	SR	μ	σ
OU	0	0.50	10.6%	21.3%	0.50	10.6%	21.3%	0.50	10.6%	21.3%	0.50	10.6%	21.3%
	1	0.34	0.8%	2.3%	0.05	0.7%	11.9%	0.60	4.8%	8.0%			
	3	0.16	0.2%	1.4%	0.44	3.4%	7.8%	0.70	4.6%	6.6%			
	5	0.17	0.2%	1.2%	0.68	4.7%	7.0%	0.66	4.2%	6.3%			
	8	-0.34	-0.3%	1.0%	0.51	3.1%	6.0%	0.60	3.9%	6.2%			
	10	-	-	-	0.26	1.3%	5.0%	0.56	3.5%	6.2%			
	15	-	-	-	0.31	1.4%	4.3%	0.54	3.3%	6.1%			
FFN	0	0.57	8.8%	15.3%	0.57	8.8%	15.3%	0.57	8.8%	15.3%	0.57	8.8%	15.3%
	1	0.60	2.0%	3.3%	0.53	6.2%	11.7%	1.07	6.5%	6.1%			
	3	1.02	2.6%	2.6%	1.15	8.2%	7.2%	1.50	7.6%	5.0%			
	5	1.32	2.3%	1.7%	1.42	9.8%	6.9%	1.55	7.3%	4.7%			
	8	1.31	2.1%	1.6%	1.05	6.4%	6.2%	1.52	7.2%	4.7%			
	10	-	-	-	0.70	3.5%	5.0%	1.48	7.0%	4.7%			
	15	-	-	-	0.51	2.4%	4.8%	1.68	7.5%	4.5%			

This table shows the out-of-sample annualized Sharpe ratio (SR), mean return (μ), and volatility (σ) of our three statistical arbitrage models for different numbers of risk factors K , that we use to obtain the residuals. We use the daily out-of-sample residuals from January 1998 to December 2016 and evaluate the out-of-sample arbitrage trading from January 2002 to December 2016. OU+FFN denotes a parametric Ornstein-Uhlenbeck model to extract the signal, but a flexible feedforward neural network to estimate the allocation function. RawFFN takes the residuals directly as signals and estimates an allocation function with a feedforward neural network. The deep learning models are calibrated on a rolling window of four years and use the Sharpe ratio objective function. The signals are extracted from a rolling window of $L = 30$ days. The $K = 0$ factor model corresponds to directly using stock returns instead of residuals for the signal and trading policy.

A.3.5 Trading Friction Results for PCA Residuals

Table A.8: OOS Performance of CNN+Trans with Trading Frictions

K	PCA factor model					
	Sharpe ratio			Mean-variance		
	SR	μ	σ	SR	μ	σ
0	0.52	8.5%	16.3%	0.22	2.6%	11.9%
1	0.88	7.3%	8.4%	0.79	9.0%	11.4%
3	0.90	5.7%	6.3%	0.62	4.7%	7.6%
5	0.81	4.5%	5.6%	0.68	4.4%	6.4%
10	-0.08	-0.4%	4.8%	-0.08	-0.4%	4.6%
15	-0.87	-3.7%	4.3%	-0.96	-3.5%	3.7%

This table shows the out-of-sample annualized Sharpe ratio (SR), mean return (μ), and volatility (σ) for the CNN+Transformer model with trading frictions on PCA residuals. We use the daily out-of-sample residuals from January 1998 to December 2016 and evaluate the out-of-sample arbitrage trading from January 2002 to December 2016. The models are calibrated on a rolling window of four years and use either the Sharpe ratio or mean-variance objective function with trading costs $\text{cost}(w_{t-1}^R, w_{t-2}^R) = 0.0005\|w_{t-1}^R - w_{t-2}^R\|_{L^1} + 0.0001\|\min(w_{t-1}^R, 0)\|_{L^1}$. The signals are extracted from a rolling window of $L = 30$ days.

Bibliography

- [1] R. Almgren and N. Chriss. Optimal execution of portfolio transactions. *Journal of Risk*, 3:5–40, 2001.
- [2] Marco Avellaneda and Jeong-Hyun Lee. Statistical arbitrage in the US equities market. *Quantitative Finance*, 10(7):761–782, 2010.
- [3] Alistair Barr. Quant quake shakes hedge-fund giants. *Marketwatch*, Aug 2007.
- [4] F. E Benth and K. H. Karlsen. A note on merton’s portfolio selection problem for the schwartz mean-reversion model. *Stochastic analysis and applications*, 23(4):687–704, 2005.
- [5] D. Bertsimas and A. Lo. Optimal control of execution costs. *Journal of Financial Markets*, 1:1–50, 1998.
- [6] D. Bianchi, M. Büchner, and A. Tamoni. Bond risk premia with machine learning. *Working paper*, 2019.
- [7] Guéant O. Bismuth, A. and J. Pu. Portfolio choice, portfolio liquidation, and portfolio transition under drift uncertainty. *Mathematics and Financial Economics*, 13(4):661–719, 2019.
- [8] Stephen Boyd, Enzo Busseti, Steve Diamond, Ronald N Kahn, Kwangmoo Koh, Peter Nystrup, Jan Speth, et al. Multi-period trading via convex optimization. *Foundations and Trends in Optimization*, 3(1):1–76, 2017.
- [9] S. Brendle. Portfolio selection under incomplete information. *Stochastic processes and their applications*, 116(5):701–723, 2006.
- [10] S. Bryzgalova, M. Pelger, and J. Zhu. Forest through the trees: Building cross-sections of stock returns. *Working paper*, 2019.
- [11] M. M. Carhart. On persistence in mutual fund performance. *Journal of Finance*, 52(1):57–82, 1997.

- [12] Á. Cartea, L. Gan, and S. Jaimungal. Trading cointegrated assets with price impact. *Mathematical Finance*, 19(6):542–567, 2018.
- [13] Á. Cartea, S. Jaimungal, and J. Peñalva. *Algorithmic and high frequency trading*. Cambridge University Press, 2015.
- [14] Álvaro Cartea and Sebastian Jaimungal. Algorithmic trading of co-integrated assets. *International Journal of Theoretical and Applied Finance*, 19(6):165038, 2016.
- [15] Luyang Chen, Markus Pelger, and Jason Zhu. Deep learning in asset pricing. Available at SSRN: <https://ssrn.com/abstract=3350138>, 2019.
- [16] Y. Chen, W. Chen, and S. Huang. Developing arbitrage strategy in high-frequency pairs trading with filterbank cnn algorithm. In *2018 IEEE International Conference on Agents (ICA)*, pages 113–116, 2018.
- [17] M. C. Chiu and H. Y. Wong. Mean-variance portfolio selection of cointegrated assets. *J. Econ. Dyn. Control*, 35:1369–1385, 2011.
- [18] Lin Cong, Ke Tang, Jingyuan Wang, and Yang Zhang. Single-step portfolio construction through reinforcement learning and economically interpretable ai. Available online at: <https://ssrn.com/abstract=3554486>, 2020.
- [19] Alexandre d’Aspremont. Identifying small mean-reverting portfolios. *Quantitative Finance*, 11(3):351–364, 2011.
- [20] D. Duffie. *Dynamic asset pricing theory*. Princeton University Press, 3 edition, 2010.
- [21] Filipovic D. Duffie, D. and W. Schachermayer. Affine processes and applications in finance. *The Annals of Applied Probability*, 13(3):984–1053, 2003.
- [22] C.L. Dunis, J. Laws, and B. Evans. Modelling and trading the soybean-oil crush spread with recurrent and higher order networks: a comparative analysis. *Neural Network World*, 16(3):193–213, 2006.
- [23] Helia Ebrahimi. Hedge fund quants lose money in 2012. *The Telegraph*, Jan 2013.
- [24] R.J. Elliott, J. Van Der Hoek, and W.P. Malcolm. Pairs trading. *Quantitative Finance*, 5(3):513–545, 2005.
- [25] Eugene F Fama and Kenneth R French. Common risk factors in the returns on stocks and bonds. *Journal of*, 1993.
- [26] Eugene F Fama and Kenneth R French. A five-factor asset pricing model. *Journal of financial economics*, 116(1):1–22, 2015.

- [27] Thomas Günter Fischer, Christopher Krauss, and Alexander Deinert. Statistical arbitrage in cryptocurrency markets. *Journal of Risk and Financial Management*, 12(1):31, 2019.
- [28] Papanicolaou A. Fouque, J.-P. and R. Sircar. Filtering and portfolio optimization with stochastic unobserved drift in asset returns. *Communications in Mathematical Sciences*, 13(4):935–953, 2015.
- [29] Joachim Freyberger, Andreas Neuhierl, and Michael Weber. Dissecting characteristics nonparametrically. *Review of Financial Studies, forthcoming*, 33(5):2326–2377, 2020.
- [30] N. Garleanu and L. H. Pedersen. Dynamic trading with predictable returns and transaction costs. *J. Finance*, 68(6):2309–2340, 2013.
- [31] N. Garleanu and L. H. Pedersen. Dynamic portfolio choice with frictions. *J. Econ. Theory*, 164:487–516, 2016.
- [32] E. Gatev, W. N. Goetzmann, and K.G. Rouwenhorst. Pairs trading: performance of a relative-value arbitrage rule. *Review of Financial Studies*, 19(3):797–827, 2006.
- [33] S. Gu, B. Kelly, and D. Xiu. Autoencoder asset pricing models. *Journal of Econometrics*, 222(1):429–450, 2021.
- [34] Shihao Gu, Bryan T Kelly, and Dacheng Xiu. Empirical asset pricing via machine learning. *Review of Financial Studies*, 33(5):2223–2273, 2020.
- [35] Shihao Gu, Bryan T Kelly, and Dacheng Xiu. Empirical asset pricing via machine learning. *Review of Financial Studies*, 33(5):2223–2273, 2020.
- [36] Jorge Guijarro-Ordóñez. High-dimensional statistical arbitrage with factor models and stochastic control. *Applied Mathematical Finance*, 24(4):328–358, 2019.
- [37] Jorge Guijarro-Ordóñez, Markus Pelger, and Greg Zanotti. Deep learning statistical arbitrage. 2021.
- [38] J. Guyon and P. H. Labordère. *Nonlinear option pricing*. Chapman and Hall/CRC, 2013.
- [39] Nicolas Huck. Pairs selection and outranking: An application to the sp 100 index. *European Journal of Operational Research*, 196:819–825, 2009.
- [40] Sebastian Jaimungal, Brian Ning, and Franco Ho Ting Lin. Double deep q-learning for optimal execution, 2020.
- [41] Jakub W Jurek and Halla Yang. Dynamic portfolio selection in arbitrage. In *EFA 2006 Meetings Paper*, 2007.

- [42] B. Kelly, S. Pruitt, and Y. Su. Characteristics are covariances: A unified model of risk and return. *Journal of Financial Economics*, 134(3):501–524, 2019.
- [43] Taewook Kim and Ha Young Kim. Optimizing the pairs-trading strategy using deep reinforcement learning with trading and stop-loss boundaries. *Complexity*, 2019, 2019.
- [44] Yerkin Kitapbayev and Tim Leung. Optimal mean-reverting spread trading: nonlinear integral equation approach. *Annals of Finance*, 13(2):181–203, 2017.
- [45] Serhiy Kozak, Stefan Nagel, and Shrihari Santosh. Shrinking the cross section. *Journal of Financial Economics*, 135(2):271–292, 2020.
- [46] D.-M. P. Kratz. *Optimal liquidation in dark pools in discrete and continuous time*. PhD thesis, Humboldt-Universitat zu Berlin, 2011.
- [47] Christopher Krauss, Xuan Anh Doa, and Nicolas Huck. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the s&p 500. *European Journal of Operational Research*, 259:689–702, 2017.
- [48] Charles-Albert Lehalle, Laura Leal, and Mathieu Laurière. Learning a functional control for high-frequency finance, 2021.
- [49] Y. Lei and J. Xu. Costly arbitrage through pairs trading. *J. Econ. Dyn. Control*, 56:1–19, 2015.
- [50] M Lettau and M. Pelger. Factors that fit the time series and cross-section of stock returns. *Review of Financial Studies*, 33(5):2274–2325, 2020.
- [51] M. Lettau and M. Pelger. Factors that fit the time-series and cross-section of stock returns. *Review of Financial Studies*, 33(5):2274–2325, 2020.
- [52] Martin Lettau and Markus Pelger. Estimating latent asset pricing factors. *Journal of Econometrics*, 218(1):1–31, 2020.
- [53] T. Leung and X. Li. *Optimal Mean Reversion Trading: Mathematical Analysis and Practical Applications*. World Scientific: Singapore, 2016.
- [54] Tim Leung and Xin Li. Optimal mean reversion trading with transaction costs and stop-loss exit. *International Journal of Theoretical and Applied Finance*, 18(03):1550020, 2015.
- [55] Thomas Nanfeng Li and Agnès Tourin. Optimal pairs trading with time-varying volatility. *International Journal of Financial Engineering*, 03(03):1650023, 2016.
- [56] TN Li and A Papanicolaou. Dynamic optimal portfolios for multiple co-integrated assets. *arXiv preprint arXiv:1908.02164*, 2019.

- [57] Yuen K. C. Liang, Z. and J. Guo. Optimal proportional reinsurance and investment in a stock market with ornstein–uhlenbeck process. *Insurance: Mathematics and Economics*, 49(2):207–215, 2011.
- [58] Bryan Lim and Stefan Zohren. Time series forecasting with deep learning: A survey. *arXiv preprint arXiv:2004.13408*, 2020.
- [59] Paul Sopher Lintilhac and Agnes Tourin. Model-based pairs trading in the bitcoin markets. *Quantitative Finance*, 17(5):703–716, 2016.
- [60] J. Liu and A. Timmermann. Optimal convergence trade strategies. *Review of Financial Studies*, 26(4):1048–1086, 2013.
- [61] Robert C. Merton. Lifetime portfolio selection under uncertainty: The continuous-time case. *The Review of Economics and Statistics*, 51(3):247–257, 1969.
- [62] Robert C Merton. Optimum consumption and portfolio rules in a continuous-time model. *Journal of Economic Theory*, 3(4):373–413, 1971.
- [63] Muhle-Karbe J. Moreau, L. and H.M. Soner. Trading with small price impact. *Mathematical Finance*, 27(2):350–400, 2017.
- [64] Sandjo-A. N. Moutari, S. and F. Colin. An explicit solution for a portfolio selection problem with stochastic volatility. *Journal of Mathematical Finance*, 7:199–218, 2017.
- [65] Supakorn Mudchanatongsuk, James A Primbs, and Wilfred Wong. Optimal pairs trading: A stochastic control approach. In *2008 American Control Conference*, pages 1035–1039. IEEE, 2008.
- [66] Liu R. Weber M. Muhle-Karbe, J. Rebalancing with linear and quadratic costs. *SIAM Journal on Control and Optimization*, 55(6):3533–3563, 2017.
- [67] John M. Mulvey, Yifan Sun, Mengdi Wang, and Jing Ye. Optimizing a portfolio of mean-reverting assets with transaction costs via a feedforward neural network. *Quantitative Finance*, forthcoming, 2020.
- [68] M.-M. Ngo and H. Pham. Optimal switching for pairs trading rule: A viscosity solutions approach. *J. Math. Anal. Appl.*, 441(1):403–425, 2016.
- [69] A. Obizhaeva and J. Wang. Optimal trading strategy and supply/demand dynamics. *Journal of Financial Markets*, 16(1):1–32, 2013.
- [70] M. Pelger. Understanding systematic risk: A high-frequency approach. *Journal of Finance*, 74(4):2179–2220, 2020.

- [71] M. Pelger and R. Xiong. Interpretable proximate factors for large dimensions. *To appear in Journal of Business Economics and Statistics*, 2018.
- [72] Markus Pelger and Ruoxuan Xiong. Interpretable sparse proximate factors for large dimensions. Available at SSRN 3175006, 2019.
- [73] Hossein Rad, Rand Kwong Yew Low, and Robert Faff. The profitability of pairs trading strategies: distance, cointegration and copula methods. *Quantitative Finance*, 16(10):1541–1558, 2016.
- [74] G. Ritter. Statistical arbitrage in cryptocurrency markets. Available online at: <https://ssrn.com/abstract=3015609>, 2017.
- [75] L. C. G. Rogers and S. Singh. The cost of illiquidity and its effects on hedging. *Mathematical Finance*, 20(4):597–615, 2010.
- [76] Justin Sirignano and Rama Cont. Universal features of price formation in financial markets: perspectives from deep learning. *Quantitative Finance*, 19(9):1449–1459, 2019.
- [77] Justin Sirignano, Apaar Sadhwani, and Kay Giesecke. Deep learning for mortgage risk. 2018.
- [78] Justin A. Sirignano. Deep learning for limit order books. *Quantitative Finance*, 19(4):549–570, 2019.
- [79] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv: 1706.03762*, 2017.
- [80] G. Vidyamurthy. *Pairs Trading: Quantitative Methods and Analysis*. Wiley, 2004.
- [81] Xiaohong Chen and H. White. Improved rates and asymptotic normality for nonparametric neural network estimators. *IEEE Transactions on Information Theory*, 45(2):682–691, 1999.
- [82] Raphael Yan and Agnes Tourin. Dynamic pairs trading using the stochastic control approach. *Journal of Economic Dynamics and Control*, 37, 03 2013.
- [83] Joongyeub Yeo and George Papanicolaou. Risk control of mean-reversion time in statistical arbitrage. *Risk and Decision Analysis*, 6(4):263–290, 2017.