Delft University of Technology
Faculty of EEMCS

Causal Factor Investing
with an Application in the Corporate Bond Market

by

D. F. K. Brouwers

to obtain the degree of Master of Science in Applied Mathematics
at the Delft University of Technology,
to be defended publicly on Friday October 18, 2024 at 15:00 PM.

Student number:	4704339		
Project duration:	February 5, 2024 – October 18, 2024		
Thesis committee:	Dr. Ir. F. Yu,	TU Delft,	daily supervisor
	Prof. Dr. Ir. C. Vuik,	TU Delft,	responsible professor
	B. den Boon,	MN,	external supervisor

Abstract

The rise of quantitative investment strategies has been driven by increased data availability and advancements in financial modeling. This thesis introduces Causal Factor Investing (CFI), a novel approach that integrates causality and machine learning to enhance the performance and explainability of factor investing strategies. Traditional factor investing often suffers from specification errors due to its reliance on correlations rather than causal relationships, and machine learning methods are frequently criticized for their 'black box' nature. CFI addresses these issues by using causal discovery methods, which are based on the mathematical properties of graph theory, to identify factors that have a cause-effect relationship with asset returns. These causal factors are then utilized as features in machine learning models to predict future returns, serving as investment signals for portfolio construction.

Our empirical analysis in the European corporate bond market utilized causal discovery algorithms including Fast Causal Inference (FCI) and Greedy Equivalence Search (GES). The use of GES in CFI improves portfolio performance compared to traditional factor investing, while FCI led to insufficient causal graphs. For the portfolios constructed with neural networks in CFI, the use of causal factors resulted in the best-performing portfolio.

Altogether, CFI contributes to the field of quantitative finance by offering an explainable and profitable approach to factor investing. For further research, we suggest exploring alternative causal discovery algorithms, including time-series causal discovery methods and other algorithms that account for hidden confounders to increase the accuracy of the causal graphs. Additionally, a practical improvement would be including transaction costs and adjust the model for risk constraints through optimization approaches.

Keywords: Causal Factor Investing, Quantitative Investing, Credit Market, Causal Discovery, Fast Causal Inference, Greedy Equivalence Search, Machine Learning, Empirical Asset Pricing, Neural Networks

Acknowledgements

This thesis is a mutual collaboration between TU Delft and MN, which is a fiduciary manager and asset manager for multiple Dutch pension funds. I am grateful that, as part of the Academic Excellence Program, I had the opportunity to apply the knowledge gained during my Bachelor's and Master's in Applied Mathematics to a complex, applied problem. This thesis marks the final part of my seven-year journey at TU Delft, a period that I enjoyed.

First and foremost, I would like to thank Dr. Fenghui Yu, with whom I could comfortably share thoughts and ideas on a regular basis. Her valuable feedback greatly contributed to this work. I am grateful to Prof. Dr. Kees Vuik for his support and for serving as the chairman of my thesis committee. I would also like to thank Dr. Fang Fang for her useful insights. From the company side, I am thankful to Bart den Boon for his guidance, business knowledge, and for sparking my interest in corporate bonds, which he believes are far more intriguing than stocks.

Lastly, I would like to thank my friends and family, who were always by my side.

Contents

1 | Introduction

Over the last few decades, the field of quantitative investment strategies has grown significantly, driven largely by the increase in data availability. One of the most famous quantitative investment strategies is factor investing, which aims to realize profits by investing in financial instruments exposed to one or multiple factors. A factor is a quantifiable characteristic or attribute that explains the difference in returns across a set of securities.

Factor investing traces back to the Fama and French 3-factor model in 1992 [13]. Subsequently, factor investing research grew significantly, researching new factors and different factor definitions, mainly focusing on equity. However, in recent years, credit markets and its factor investing research has grown. As of May 2020, the International Capital Market Association (ICMA) approximates the combined equivalent nominal value in the European investment grade [1] (IG) corporate bond market to be €5.65 trillion. A large part of factor investing literature is dedicated to researching which factors best explain returns across a set of assets. This thesis does not focus on factor identification or comparative analysis, but instead on utilizing known factors for an investment strategy.

More recently, machine learning techniques have been integrated into the field of financial mathematics, and thus into the factor investing industry. With abundant data, machine learning can uncover non-linear relationships that may outperform traditional investment strategies. In equity markets, several papers have been published using machine learning methods in factor investment strategies. In credit market, this number drops to a few papers.

In November 2023, an article by de Prado et al. [29] was published with the title: ‘Why Factor Investing Has Failed: The Role of Specification Errors’. The publication argues that *specification errors* in factor strategies contribute to underperformance through a specific causal mechanism grounded in causal theory. A misspecified factor model does not use the ‘correct’ factors based on a cause-effect relationship but instead relies on factors derived from correlations. Another argument in favor of using causal knowledge is that, in combination with machine learning methods, it could lead to improvement in explainability of the model. A significant drawback of machine learning methods is their ‘black box’ nature, which often leads to models that are difficult to interpret due to their reliance on correlation. While quantitative teams in investment firms can be familiar with machine learning, these models must be transparent and understandable to their clients. Hence, adding to the explainability of a machine learning factor model is of high importance. The authors advocate for rebuilding the factor investing literature in a more scientific manner, emphasizing the theory of causality.

This has lead to the main research question of this thesis:

> “Can we increase performance and explainability of factor investing strategies via machine learning and causal theory”

Through this research question, we have developed a novel factor investing strategy: *Causal Factor Investing (CFI)*. Our main contribution to existing literature is the integration of causal theory in an investment strategy using factors. The concept and motivation for CFI are presented in Section 1.1 to provide a high-level overview. In Section 1.2, we discuss the related literature

[1] Bonds with lower risk and therefore higher credit ratings

on machine learning and causality in the field of factor investing. We conclude this chapter with a complete outline of the thesis in Section 1.4.

1.1 Model concept

Our main goal is to integrate causal theory in an investment strategy to enhance profitability and explainability. More specifically, we aim to find the causal structure of factors and returns. A scientific theory can explain why an observed certain event takes place and is falsifiable. This has to be consistent with empirical evidence or ideally proven with experiments including interventions. In experimental sciences (physics, chemistry etc.) it could be possible to identify causal relations by interventional methods which has is fundamentals in the mathematical theory of causal inference.

In finance however, and therefore in factor investing context, direct interventions are not feasible. Financial markets are complex systems, and it is impossible to randomly manipulate variables such as interest rates, economic growth, or factors to observe their effects on asset returns. Instead, we must work with observational data, which complicates the identification of causal relationships. In this setting, *causal discovery* methods, become essential for uncovering the underlying causal structures.

Definition 1.1 (Causal Discovery)**.** *The process of learning the causal structure from observational data is called causal discovery.*

The output of causal discovery methods is a *graph*. The innovation of CFI lies into using the causal structure to identify the factors that have a causal relation to the returns. These factors are defined as *causal factors*. Then, the causal factors are to be used in the investment strategy to enhance profitability and explainability.

The motivation behind the use of machine learning methods is to combine different factors of interest into one composite machine learning signal. Factor investing strategies aim to generate profits by maintaining a portfolio exposed to a certain factor. Some multi-factor strategies use different factor portfolios simultaneously, dividing the invested money over the portfolios. Instead of simultaneously holding multiple factor portfolios, one portfolio could be managed containing the information of all factors. The factor investing strategy remains the same, using the machine learning signal as criterion to construct the factor portfolio. This has two main advantages. First, the profitability of these investment strategies can be improved because machine learning methods can uncover non-linear relations of the factors and the returns. Second, transactions cost can be lowered due to managing only one portfolio. It is important to mention that most researchers do not necessarily resolve to information of factors to create the machine learning signal (see the related literature in Section 1.2). Instead, they use all information available, which are all available characteristics of financial securities.

Logically, we question ourselves: ‘How can we determine the most suitable choice for this signal?’. To answer this question one has to go back to the core goal of the factor investing strategy: harvest returns trough factor exposure. The investor believes that investing in assets exposed to a factor, will enhance the return of the strategy. Therefore, an appropriate choice for a signal is the **predicted** future return. If the prediction of future returns with factors is accurate, this can be used as a criterion for the factor investment strategy by investing in assets with high predicted returns. In literature, the problem of predicting future returns (or asset prices) is often referred to as *empirical asset pricing*. The predicted future returns are modelled via machine learning methods.

Taking these elements into consideration we constructed CFI which can be summarized into three key steps:

1. **Causal Analysis:** Perform causal discovery on factors and returns to obtain causal factors.

2. **Empirical asset pricing via machine learning:** Predict future returns with machine learning methods using the causal factors as features. The predicted future returns will serve as composite signal.

3. **Investment strategy:** Use the signal, the predicted future returns, to invest and construct the portfolio.

CFI is visualized in Figure 1.1, where each arrow indicates the output of the preceding step, which serves as the input for the subsequent step.

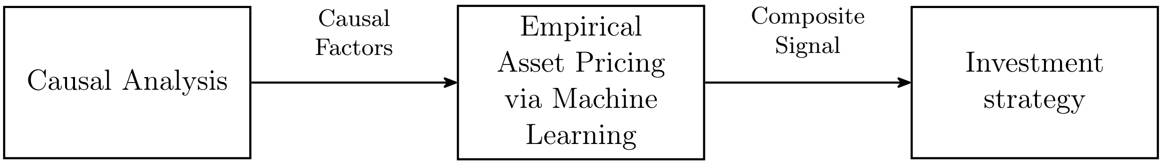


Figure 1.1: Simplified visual representation of Causal Factor Investing (CFI)

CFI is constructed generically, such that it can be used for different type of asset classes. The application of this thesis is in the European corporate bond market where CFI is tailored to fit this context. Via backtesting CFI we obtain empirical results.

1.2 Related literature

Related literature can be categorized into three parts: factor investing, empirical asset pricing via machine learning, and causal discovery. In this section, these categories are discussed in this order. It is possible that some papers could be relevant to multiple categories. For related literature on factor investing, this section provides a high-level overview of important developments within in factor investing with a focus in the credit market. In the next chapter (see Section 2.2.2), we will delve into more detail, including additional articles, regarding the factor definitions themselves. Here we will motivate the choices for the factor definitions used in this thesis as there are multiple possible factor definitions for each factor.

Factor Investing

In 1993, Fama and French [14] pioneered the factor investing literature by publishing their famous 3-factor model. This model sparked new financial economic research aimed at identifying new factors and constructing factor models within both equity and credit markets. In 2015, Fama and French [15] extended the 3-factor model to a 5-factor model including the investment factor and the profitability factor. One of the most influential articles for the credit markets was written by Houweling et al. [22], who defined a multi-factor model in the credit market considering the value, momentum, size, and low-risk factors. These factors were based on bond characteristics, whereas previous literature in the corporate bond market mostly used equity definitions. Promising evidence for the carry factor in credit was presented by Israel et al. [25], who combined equity and credit data to define factors. Slimane et al. [3] extended this work to a six-factor model, incorporating three traditional risk measures as additional factors.

Empirical Asset Pricing

The article by Gu et al. [18], *Empirical Asset Pricing via Machine Learning*, lays the foundation for using machine learning methods for factor investing in this research. They conducted a comparative analysis of machine learning methods for empirical asset pricing in the equity market. Trees and neural networks were the best-performing methods for predicting month-ahead returns. Over 900 predictors were used, combining stock characteristics and macroeconomic signals. Additionally, they demonstrated economic gains by implementing a factor investing strategy on these predictions, taking a long position on the top 10% and a short position on the bottom 10%. The book by Coqueret et al. [9] also reviews numerous machine learning methods for factor investing strategies, focusing on practical implementation.

In subsequent years, several articles were published using machine learning methods to predict returns in markets other than the equity market. In line with the work of Gu et al. [18], Mansouri et al. [30] used machine learning, excluding neural networks, to predict corporate bond returns in emerging credit markets. They used a large set of bond characteristics and macroeconomic variables in their predictor set, assigning their bond characteristics nine factor categories; each factor category contains multiple bond characteristics. An important addition was their examination of profitability after transaction costs and constraints for factor investment strategies. Using linear optimization, they concluded that investment strategies based on regression trees were profitable in a real-life scenario. Bali et al. [2] explored machine learning to price United States corporate bond returns, arguing for performance improvement by adapting the theoretical structure, a stock-bond connection, of the Merton model.

In contrast to the articles in the previous two paragraphs, Cherief et al. [6] used only few factor definitions as features to predict corporate bond returns. Their model builds on the previously mentioned six-factor model of Slimane et al. [3]. The focus of their article was on predictive performance and the importance of the factors, without considering factor investing strategies. Using only eight characteristics, Zhu et al. [50] investigated factor investing strategies in the Chinese commodity market using linear models and tree-based methods.

Causality

As mentioned at the beginning of this chapter, Lopez de Prado et al. [29] argue for using the mathematical theory of causality in factor models. Extensive factor research has created the so-called ‘factor zoo’, containing many factors and factor definitions. Incorrectly specifying a model, by including or excluding certain factors, leads to underperformance due to causal effect relations grounded in causal inference theory. Using the correct causes (factors) of the effect (returns) can solve this problem. However, finding the true causal structure from observational data, as is often the case in financial contexts, is challenging. Causal discovery algorithms can be used to estimate the causal structure of observational data in the form of a causal graph.

To our knowledge, no papers apply causal discovery specifically to factors and returns for an investment strategy, though some explore the relationships between them using causal discovery algorithms. Gu et al. [17] utilized causal discovery to uncover causal relations between the five Fama and French [15] factors and stock returns. Causal discovery was performed separately for each stock and and the return of factor portfolios and used the results to create an aggregated graph. They assumed that the factors and returns were independent and identically distributed i.i.d and there did exist hidden confounders. Therefore, they used three popular i.i.d. causal discovery algorithms: *Peter-Clark* (PC), *Greedy Equivalence Search* (GES), and *Linear Non-Gaussian Acyclic Model* (LiNGAM) belonging to three different causal discovery categories constraint-based, score-based, and functional models, respectively. Sadeghi et al. [41] also researched causal discovery on the same five equity factors and Apple stock returns, using a more complex novel CD method called CDNOTS. This algorithm is a constraint-based algorithm for time series data that allows for unobserved factors in the model. D’Acunto et al. [12] used a time-series functional causal discovery algorithm VAR-LiNGAM on 11 risk equity factors to gain insights into how these factors influence each other. VAR-LiNGAM is a time-series extension of the LiNGAM, which also does not allow for hidden confounders. Howard et al. [23] defined a new factor based on a causal network.

Besides causal discovery on factors, research has been conducted in different financial contexts. One can attempt to find the causal structure of different stock returns on each other. Pamfill et al. [36] propose a score-based algorithm DYNOTEARS that does not allow for hidden confounders, but is robust for higher dimensions. Namely, the researched this algorithm for 97 stocks. Also, several articles, such as articles of Hossain et al. [19] and Zaher et al. [46] have been published containing comparative analyses of various causal discovery methods for temporal and/or non-temporal data. This is useful for arguing for certain causal discovery algorithms in the context of this thesis.

During the same time period as this research, Tang [44] published an article using causal discovery for an investment/trading strategy on stock returns among themselves. Tang used three time-series causal discovery methods ts-FCI, VAR-LiNGAM, and TiMINo, to uncover the causal structure of stock returns over time. From the causal graph, they identified the set of parent vertices of stock X, denoted as $Pa(X)$, which represents the direct causes of stock X. They then fitted a predictive linear regression model on each stock X to forecast future returns using these parent stockss and stock X itself as predictors. Notably, only VAR-LiNGAM could efficiently handle large datasets. The trading portfolio incorporating parent stocks via VAR-LiNGAM outperformed the trading portfolio that used only past values of the stocks themselves.

1.3 Data

Our dataset is the *Markit IBoxx Euro Corporate Senior Index*, provided by S&P. This index consists of monthly data exclusively including investment-grade and senior corporate bonds[1]. Portfolio managers, managing an active corporate bond portfolio, often construct a credit portfolio by selecting bonds from an index and assigning specific weights (underweight or overweight) to each bond based on their investment strategy. Often, the goal of these portfolio managers is to beat the benchmark. We define our dataset as this corporate bond index. It is important to note that the index itself is also used as the *benchmark* and is used to compare different investment strategies in this research.

Different from a large part of equity modelling, we do not necessarily follow single bonds over time. Instead, factor investing strategies focus on the cross sections of bonds and on portfolios returns over time. Each month it is possible for bonds to appear and disappear in the index, based on new bond entries and the expiration of some of the bonds. This means that the composition of the index changes. Bluntly stating, one can see monthly bond data as ‘new’ monthly sample. In similar way, we consider a single bond as a bond observation. In the application of this thesis, bonds are indexed by i and months by t.

The dataset comprises a total of 336,671 monthly bond observations from 01/01/2009 to 31/12/2023. The number of bonds in the custom benchmark per month varies from approximately 500 to 3500. As time progresses, the number of bonds per month in the index increases, as shown in Figure 1.2.

Table 1.1 presents descriptive statistics of the dataset for all bond observations. In this thesis, all calculations for factor portfolios are performed bottom-up, based on all monthly bond observations. To ensure a fair comparison, the index calculations are done similarly. We fix the index weights at the start of the month, whereas in reality, the weights change minimally on a day basis. This effect is negligible when comparing our bottom-up construction of the index to the published top-level index performance. However, in the S&P published top-level index performance, spreads and yields are calculated in a slightly different way. Therefore, our Z-spread[2] values may slightly differ from the published official numbers. Moreover, all metrics

[1] Investment grade bonds have high credit ratings ranging from AAA-BBB and senior bonds are higher claim on a company’s assets in the event of bankruptcy

[2] Spread measure that captures the yield compensation of a corporate bond, compared to a similar sovereign bond, by using the LIBOR risk free rate (see Definition 2.4)

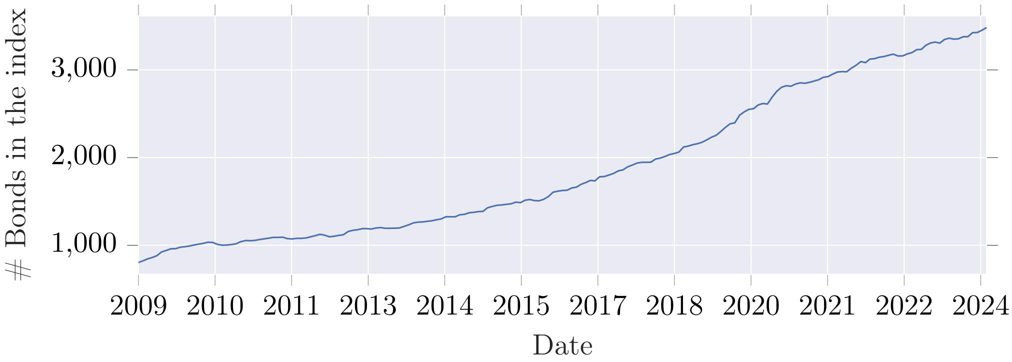


Figure 1.2: Total number of bonds per month the index

of the index portfolio are calculated using index weights.

Table 1.2 shows the bottom up yearly performance and some metrics of the dataset, which, as previously mentioned, is the index itself. The Z-spread was highest during the years 2009-2012, attributed to the financial crisis in 2008 resulting in higher credit spread compensation.

Table 1.1: Descriptive statistics dataset *Iboxx Eur Corporate Senior Index.* For every characteristic the mean and five percentiles (5%, 25%, 50%, 75% and 95%) are stated.

Mean & Percentiles	mean	5%	25%	50%	75%	95%
Duration[2]	4.95	1.30	2.75	4.40	6.49	10.27
Z-Spread	82.84	11.11	34.36	60.10	101.60	223.41
Notional (EUR M)	828	500	500	750	1000	1500
Ex. Return Lib[1] (%)	0.10	-1.48	-0.19	0.08	0.44	1.77

Table 1.2: Yearly performance Iboxx Eur Corporate Senior Index

Date	Duration[2]	Z-spread	Ex. Return Lib[3] (%)
2009	3.98	186.72	0.71
2010	4.09	112.92	-0.01
2011	3.93	145.30	-0.24
2012	4.03	140.30	0.43
2013	4.34	90.62	0.18
2014	4.62	66.63	0.10
2015	4.98	74.44	-0.09
2016	5.20	72.73	0.20
2017	5.26	46.73	0.15
2018	5.16	60.41	-0.20
2019	5.14	70.04	0.24
2020	5.31	99.61	0.08
2021	5.39	57.02	0.12
2022	4.92	99.06	-0.10
2023	4.57	91.76	0.15

[1] Monthly excess return over LIBOR

[2] Modified duration (see Section 2.2.3)

[3] Annual average of monthly excess return over LIBOR

1.4 Outline

This thesis is structured as follows. In Chapter 2, we introduce the reader to the general background of factor investing. Here, the factor definitions used in this thesis are also stated and motivated, given the variety of possible definitions. Chapter 3 provides the mathematical preliminaries necessary for causal discovery. Causal Factor Investing (CFI) is discussed in Chapter 4, covering the integration of the three key steps: causal discovery, empirical asset pricing via machine learning, and the investment strategy. The application of CFI is presented in Chapter 5, where CFI is fitted in the context of the credit market. Chapter 6 presents the numerical results of applying CFI in the corporate bond market, with these results obtained through backtesting. We conclude this thesis with a discussion of our findings in Chapter 7.

2 | Factor Investing

2.1 Factor Models

Factor investing is a strategy that aims to enhance portfolio returns, manage risk and increase diversification. The principle behind factor investing is that the difference of returns of an asset class depends on drivers which we call *factors*.

Definition 2.1 (Factor). *A factor is a quantifiable characteristic or attribute the explains the difference in returns across a set of securities.*

The core aim of factor research is to explain the drivers of *cross-sectional* asset returns.

Definition 2.2 (Cross-section). *In financial analysis, a cross section refers to the examination of a set of securities at a specific point in time without considering differences in time.*

There are two type of factors: Macroeconomic factors, based on macroeconomic trends, and style factors, based on characteristics of an asset. Often, the term 'factor' is also used when speaking about the performance of a portfolio exposed to this factor: a factor portfolio. The construction of a factor portfolio is explained later in this section. In this case a factor F_t can be seen as a time series object. For a certain factor k, we adhere to the notation of F_t^k for a *factor portfolio* and we will use the $f_{i,t}^k$ to denote the *factor score* which is the value of the factor characteristic k of asset i at time t.

A cross-sectional factor models expresses the return of any asset (or portfolio) i at time t as a linear combination of K factor time series. This can be mathematically formulated as

$$r_{t,i} = \alpha_t + \sum_{k=1}^{K} \beta_{i,k} F_t^k + \epsilon_{t,i} \tag{2.1}$$

where

- $r_{t,i}$ is the return of asset i at time t,
- α_t is the return that is not driven by a factor; can be the risk free rate
- F_t^k is the value of the factor portfolio at time t of factor k,
- $\beta_{i,k}$ is called factor loading which is the sensitivity of asset i to the factor k,
- $\epsilon_{t,i}$ is the error at time t for asset i.

Factor loadings $\beta_{i,k}$ are often calculated via cross-sectional regression, where the return of an individual assets is regressed on the returns of factor portfolios. Within factor model research, the aim is to find factors that best explain the cross section of returns by minimizing the residuals of these regressions. This thesis does not focus on factor loadings and factor identification, but merely on using factors for an investment strategy.

One of the most famous examples of a one factor model is the Capital Asset Pricing Model (CAPM) from Markowitz [31]. Here, the return of an asset or portfolio is described by the market return:

$$r_{t,i} = r_f + \beta(r^{market} - r_f) + \epsilon_{t,i}. \tag{2.2}$$

Here r_f is the return from the risk-free rate and r^{market} is the market return. When one takes expectations and removes the subscript we see the famous CAPM in the form

$$\mathbb{E}[r] = r_f + \beta(\mathbb{E}[r^{market}] - r_f). \tag{2.3}$$

In this setting, β can is expressed as

$$\beta = \frac{\mathrm{Cov}(r, r^{market})}{\mathrm{Var}(r^{market})}. \tag{2.4}$$

Subsequent research expanded on this foundation by identifying additional factors that could explain returns better than the market factor alone. In 1993, Fama and French [14], proposed a model including two more factors in the equity market. They observed that two type of stocks outperformed: stocks with small market capitalization outperformed stocks with high market capitalization and stocks with high book-to-market ratio outperformed stocks with low book-to-market ration. The former is referred to as the size factor and the latter as the value factor (assets with lower price relative to their fundamental value). The Fama and French 3 factor model has the following form:

$$r = r_f + \beta(r^{market} - r_f) + \beta_2 \cdot SMB + \beta_3 \cdot HML + \alpha. \tag{2.5}$$

SMB is the size factor and stands for 'Small Minus Big' whereas *HML* is the value factor and stands for 'High Minus Low'. Note that SMB and HML are factor portfolios.

Factor Investing: Portfolio Construction

Based on the article of Fama and French [14], a generic factor portfolio can be defined as follows:

1. Sort the assets on factor score f at t
2. Long the top x % of assets and short the bottom x % of asset in your portfolio P
3. Repeat the process at $t + 1$ through rebalancing portfolio P

Here, the investor has freedom in choosing factors, their investment horizon, percentage of longing/shorting, and rebalancing time. Note that the numerical factor score is calculated per bond, such that we can sort set of bonds.

Instead of ranking assets according to one factor, one can also perform the same operations simultaneously on other factors and create multiple portfolios, which is called a *multi-factor strategy*. In this way, investors can divide their total wealth among different factor portfolios of interest.

2.2 Application in the Corporate Bond Market

2.2.1 Corporate Bonds

A stock, the most commonly known financial derivative, represents ownership in a corporation. When you buy a stock in a company, you purchase a small piece of that company. Stocks are traded on the stock market, and the value of a stock can fluctuate based on various factors such as the company's performance, market conditions, and investor sentiment. A bond is another popular financial instrument, and similarly, bonds are traded in financial markets, with their value also changing over time.

A bond is a financial instrument that functions as a debt obligation. It represents a loan issued by the *issuer*, which often are corporate, sovereign, or sovereign-related entities. These loans are bought by an investor. Often, the loan is of a substantial amount, allowing the issuer to raise capital for their ongoing operations or, for example, fund and finance significant projects or take-overs. In return for holding a bond, the investor receives compensation for being exposed to risk. This compensation comes in the form of interest payments, based on a fixed or variable interest rate, which are called *coupons*. When the bond reaches *maturity*, the issuer repays the original value of the loan, known as the *face value* of the bond, through the principal payment.

The most important forms of credit risk, from an investor's perspective, are listed below:

- **Credit risk:** The risk that the bond issuer will not be able to make the the interest payments and the face value repayment of the bond,
- **Market risk:** The risk that the market price of the bond changes,
- **Interest rate risk:** The risk that changes in interest rates will effect the value of the bond,
- **Liquidity risk:** The risk that the investor will not be able to sell the bond easily on the market due to illiquidity on the bond market
- **Inflation risk:** The risk that inflation will devalue the bond's future cash flows.

The mathematical definition of a bond can be found in Definition 2.3.

Definition 2.3 (Bond)**.** *[35] A zero-coupon bond has value $B_{zero}(t, T)$ at time t and does not yield any coupon payments to the investor. At maturity T, the investor is paid the face value of the bond which, without loss of generality, can be expressed as 1 unit of currency: $B_{zero}(T, T) = 1$.*

A coupon bond $B(t, T)$ provides the bond holder n periodic coupon payments, at pre-specified times $\{t_1, \ldots, t_n\} \in [0, T]$, as well as the face value of the bond at maturity. The coupon payments are specified by a fixed or floating rate of the face value of the bond.

Rating and Seniority

One can imagine that not all bonds have the same credit risk; some companies are more reliable or financially healthy and thus are more likely to pay back the loan. Consequently, all bonds have a *rating* that reflects their creditworthiness. Rating agencies such as Standard & Poor's or Moody's assign ratings to bonds, with the highest rating being AAA and the lowest rating CCC. The collection of bonds with ratings from AAA to BBB are *investment grade* (IG), indicating low credit risk. Bonds with ratings from BBB to C are *high yield* (HY) grade bonds, also known as junk bonds, and indicate high credit risk.

In addition to ratings, bonds can also differ in their seniority within the capital structure. *Senior bonds* have a higher claim on a company's assets in the event of bankruptcy or liquidation, meaning they are repaid before other (junior) bonds. As a result, senior bonds are considered to have lower credit risk because of their higher claim. The scope of this thesis is limited to IG senior bonds.

Yield

The *yield* of a bond is a general term used to express the return on a bond based on the initial investment. A simple way of looking at the return of a bond is the *current yield*, expressed as:

$$\text{Current Yield} = \frac{\text{Annual Coupon Payment}}{\text{Price of the Bond}}. \tag{2.6}$$

and it changes as the bond price changes. Both these calculations do not incorporate other elements such as the time value of money and maturity. The *Yield to Maturity* (YTM) is the interest rate for a bond bought at market price and held until maturity. Mathematically, this is the discount rate that makes the sum of the bond's future cash flows equal to its price. That is,

$$\text{Price} = \sum_{t=1}^{T} \frac{\text{Cash Flows}_t}{(1 + \text{YTM})^t} \tag{2.7}$$

where YTM is the yield to maturity.

The yield curve plots the yields (YTM) of bonds and their maturities. The change in the slope of yield curves can possibly provide information of economic sentiment. A rising positive yield curve slope can suggest economic growth expectations because long-term bonds have higher yields to match future inflation. Conversely, a decreasing slope (possibly an inverted yield curve), can signal an economic recession; yields of bonds with longer maturities decrease because of expectations of an economic downturn. Figure 2.1 shows a plot of different yield curves. At any time, the number of bonds with different maturities is finite, so we do not know the exact yield for all maturities. The yield curve over all maturities is modeled to fit the known points.

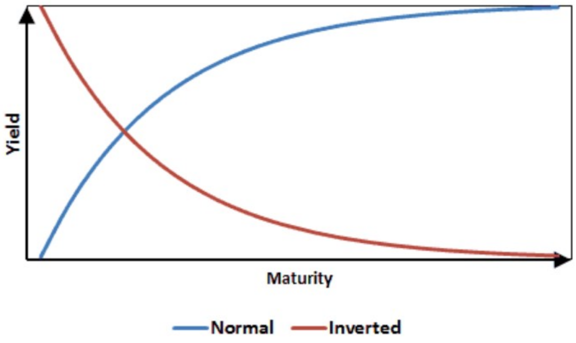


Figure 2.1: Yield curves

Credit spread

Now that the concept of yield curves is discussed, we introduce a fundamental aspect of fixed-income analysis: *credit spread*. Credit spreads represent the difference in yield between a risk-bearing bond and a considered risk-free bond. Often, the comparison is between credit spreads of corporate bonds, which have higher credit risk, and government bonds, which have lower credit risk. This spread essentially measures the extra yield that an investor gains for taking the additional risk associated with corporate debt over 'risk-free' government debt. Credit spread is a crucial measure for investors and analysts, as it offers insights into the riskiness of a bond and is a great tool for comparing different bonds.

The width of a credit spread, for a fixed maturity, can vary greatly depending on the creditworthiness of the bond. High-yield bonds, which have lower credit ratings, normally have higher credit spreads than investment grade bonds of the same maturity. One can also use credit spreads to compare bonds of similar quality. Even though the bonds appear similar, some may have higher yields than others, which could imply that these bonds are underpriced. The *credit* *spread curve* is a graphical representation that plots the credit spread of similar bonds against different maturities. When the actual credit spread is higher than the credit spread curve for a certain maturity, it could also indicate that this bond is undervalued.

There are different ways to measure credit spread. In this thesis, we will use the Z-spread, which is a commonly used credit spread measure.

Definition 2.4 (Z-spread)**.** *The Z-spread Z is the spread, often expressed in basis points, that should be added to the LIBOR curve such that all the discounted cash flows of a bond equal the current market price. The LIBOR curve is considered a risk-free curve, which stands for the London Interbank Offered Rate curve, is a graphical representation of the interest rates at which major global banks lend to one another for various maturities. This can be mathematically expressed as*

$$P = \sum_{i=1}^{n} \frac{C_i}{(1 + L_i + Z)^{t_i}} + \frac{F}{(1 + L_n + Z)^T} \tag{2.8}$$

where

P : *Market price of the bond,*
C_i : *Coupon payment at time t_i,*
L_i : *Spot rate for the corresponding LIBOR zero curve,*
Z : *Z-spread,*
F : *Face value of the bond,*
t_i : *Time to the i-th cash flow,*
T : *Time to maturity of the bond.*

Excess Return

Excess return for corporate bonds refers to the return of a bond above the corresponding risk-free curve. This measure isolates the performance of the bond relative to a risk-free return, providing a clearer view of the bond's actual performance and risk-adjusted return. It is convenient to look at excess returns instead of absolute returns for a corporate bond investor who intends to beat a benchmark without taking active interest rate risk. In this thesis, we will use excess returns over LIBOR, which we refer to as excess return, to evaluate the performance of corporate bonds.

2.2.2 Factor Definitions

The credit factor investing strategy changes minimally form equity strategy. A monthly credit factor investing strategy is defined as follows:

1. Sort the assets on factor f at the start of month t
2. **Long** the top x % of assets in your portfolio P
3. Repeat the process at month $t + 1$ through rebalancing portfolio P

In the corporate bond market, regular shorting is often difficult due to illiquidity or restricted by clients. Therefore, in the construction of factor portfolios, long-only portfolios on the best-ranked subset of bonds are often considered. Furthermore, factor investing strategies allow for flexibility in the percentage of assets in the portfolio as well as choosing between equal or market weights.

Most factors are based on a certain ideology which can be supported by economic or empirical arguments. Researchers constantly try to optimize the factor definition that correspond to a certain factor. This has lead to numerous papers and definitions which, all together, are called the 'factor zoo' as in the article of Cochrane et al. [8]. Therefore, these factors do not have world-wide consensus of the 'correct' definition in the corporate bond market yet. The factors considered in this thesis, along with the rationale behind their selection, are listed below.

- **Size:** The size factor targets bonds with low market debt. The rationale behind this factor is that smaller companies, measured by the market value of their debt, might offer higher returns due to factors such as higher risk or less information availability.
- **Value:** The value factor targets bonds that appear to be undervalued compared to their fundamental value. The rationale behind this factor is that, over time, undervalued bonds will revert to their 'true' value which generates returns.
- **Low-Risk:** The low risk factor targets bonds that are less risky. The rationale behind this factor is that less risky bonds offer better risk-adjusted returns over time.
- **Momentum:** The momentum factor targets bonds that have shown strong past returns. The rationale behind this factor is that past winners tend to be future winners. On the contrary, past losers tend to be future losers.
- **Carry:** The carry factor targets bonds that have a higher yield. This is based on idea that carry measures expected return if market conditions, the risk free and carry structure, stay the same.

In the following sections, we elaborate on the related literature for these factors and state the factor definitions used in this thesis. We not only provide the factor definitions but also detail the calculation of the factor score for each individual bond. The factor scores are denoted as $f_{i,t}$ for bond i at time t.

Size

The size factor, one of the oldest factors, is based on the rationale that smaller firms tend to outperform larger firms in the equity market. Translating this concept to the corporate bond market has led to various definitions. Houweling et al. [22] were among the first to provide evidence on the size factor in the corporate bond market. For each bond i, they define the size factor score using the total index weight of the issuer of that bond, which represents the total debt of the issuer:

$$\mathrm{siz}_{i,t} := \text{Total outstanding debt}^{1} \text{ of the issuer of bond } i \text{ at time } t \tag{2.9}$$

This definition of total debt is the most commonly used in existing literature. In contrast, Israel et al. [25] use equity data to assess the size factor, finding no significant size effect in investment-grade bonds when using equity market capitalization as a metric. J.P. Morgan adopts an even more extensive definition, *otal enterprise value*, as detailed in the paper by Saul [11]. Despite these variations, we adhere to the definition given in Equation (2.9) due to its simplicity and prevalent use in existing research.

Value

To construct the value factor, one needs to invest in 'cheap' securities. Securities that are undervalued tend to outperform those that are overvalued. To determine which bonds are undervalued, fundamental measures like earnings or the equity book value are commonly used. Correia et al. [10] were among the first to introduce the value factor for corporate bonds by explicitly modeling

[1] Total outstanding debt of the issuer in the dataset

the default probability of an issuer. Houweling et al. [22] compare the required market compensation ('fair value') for the credit spread to the actual credit spread. If the required market compensation is higher than the actual credit spread, the bond is considered to be undervalued. The required market compensation is calculated by regressing the credit spread on some risk predictors in the cross section of bonds. Houweling et al. [22] perform a cross-sectional which can be expressed as

$$S_{i,t} = \alpha + \sum_{r=1}^{4} \beta_r \mathbb{1}_{i,t}^{r} + \gamma M_{i,t} + \delta \Delta S_{i,t} + \epsilon_i. \tag{2.10}$$

where $S_{i,t}$ is the credit spread, $\mathbb{1}_{i,t}^{r}$ is the indicator function equal to 1 if bond i has rating r, $M_{i,t}$ is the maturity, and $\Delta S_{i,t}$ is the 3-month spread change in the credit spread. For each bond, the percentage difference between the actual credit spread and the fitted credit spread is used as the value factor score:

$$\text{val}_{i,t} = \frac{S_{i,t} - \hat{S}_{i,t}}{S_{i,t}}. \tag{2.11}$$

A variety of different credit spread measures and regression predictors can be used. In the research paper by FTSE Russell [40], the option-adjusted credit spread (OAS) is used as the credit spread, and they include the time to maturity **squared** in their regression model. They also perform the regressions per industry rather than over the entire cross-section. Henke et al. [21] utilize measures such as stock volatility, leverage ratio, profitability, rating, modified duration, and 3-month OAS change. Instead of using the percentage change between the fitted credit spread and the actual credit spread, they employ a log transformation in the form:

$$\text{val}_{i,t} = \ln\left(\frac{\widehat{\text{OAS}}_{i,t}}{\text{OAS}_{i,t}}\right) - 1. \tag{2.12}$$

Israel et al. [25] use two value factor definitions: the value factor as defined by Correia et al., and a combination of credit rating, bond duration, and volatility of bond excess returns over the last 12 months.

In this thesis, due to limited data availability on company fundamentals, we use the value definition as specified in Equation (2.11) with the regression approach outlined in Equation (2.10).

Low-Risk

The low-risk factor, also known as the defensive factor, is designed to capture premiums through higher risk-adjusted returns from lower-risk companies. A crucial question in this context is the selection of appropriate risk measures to construct the low-risk factor. The creation of the low-risk factor for corporate bonds was introduced by Ilmanen [24], where both credit and interest rate risk were minimized. This was achieved by selecting credit rating and time to maturity as the two factor scores, respectively. A high credit rating and a low maturity would suggest higher low-risk premiums.

Most researchers adopted this definition, including Houweling et al. [22] and Cherief et al. [6] in constructing their low-risk factor. They use the combination of these factor scores in factor construction (see Section 2.1), implemented as a double sort on rating, followed by maturity. Alternatively, some researchers integrate equity and bond data to refine their low-risk assessments. For instance, Israel et al. [25] employed three specific low-risk measures: market leverage, gross profitability, and low duration. Doctor [11] relies even more on company fundamentals in his low-risk factor definitions. In this thesis, we adhere to the two factor scores of Ilmanen [24]. Instead of using a double sort, we aim to create a single factor score and define the low-risk factor as follows:

$$\mathrm{lr}_{i,t} = \begin{cases} 1 + \dfrac{1}{\mathrm{Maturity}_{i,t}}, & \text{if the rating of bond } i \in \{\mathrm{AAA}, \ldots, \mathrm{A}\}, \\ \epsilon + \dfrac{1}{\mathrm{Maturity}_{i,t}}, & \text{if the rating of bond } i = \mathrm{BBB}, \end{cases} \tag{2.13}$$

where ϵ is a very small value. In this definition, a higher low-risk factor score indicates higher low-risk premiums.

Momentum

The momentum factor principle is based on the notion that financial instruments that perform well in the present are likely to continue their performance in the future, while those that perform poorly will continue on the same trajectory. This concept was first explored in the equity market in 1993, but it was not until 2013 that Jostova et al. [26] extended this analysis to the corporate bond market. They defined the momentum factor score based on the past six-month return, incorporating a one-month implementation lag, as follows:

$$\mathrm{mom}_{i,t} = \left(\prod_{j=1}^{6} (1 + r_{i,t-j}) \right) - 1 \tag{2.14}$$

where $r_{i,t}$ is the excess return.

A higher momentum score suggests the potential for higher future returns. Recent articles from Houweling et al. [22], Cherief et al. [6], and Israel et al. [25] have adopted the definition of Jostova et al. [26] in their work. There are elements in this definition that allow some flexibility. For instance, one could use returns based on absolute or excess returns relative to duration-matched treasuries which is proposed by Houweling et al. [22]. Additionally, the time period considered for momentum calculation can vary, with options extending up to twelve months or reducing to three months, although six months remains the most commonly utilized time period. Another variation includes using average returns instead of cumulative returns.

A significant adaptation, known as the *amended momentum* factor, was proposed by Slimane et al. [3]:

$$\text{Amended momentum} = \frac{\text{Compounded average weighted excess returns} - 1}{\text{Average weighted spread duration}}. \tag{2.15}$$

This variation addresses the anomaly where bonds with the lowest momentum scores sometimes perform unexpectedly well. For further details of this variation, the reader is directed to the work of Slimane et al [3].

Similar to other factor scores, it is possible to use equity data linked to bonds, as it could be argued that news trends are better reflected in equity data than in bond data. Furthermore, equity data could better predict future downgrades which is suggested by Henke et al. [21]. In this thesis, we use bond data only and adhere to the most general momentum factor score presented in Equation (2.14).

Carry

Isreal et al. [25] are one of the first to use the carry factor when explaining the cross-sectional return of corporate bonds. Stated simply, carry is return of a security if time passes and nothing changes; a current higher yield bond outperforms a lower yield bond the conditions in the future remain the same. They define carry as the credit spread measure OAS, which we introduced in the previous subsection on the value factor. The carry factor is one of the simplest factors to implement. In the scope of this research we consider Z-spread (see Definition 2.4), which is similar to OAS as we consider few bonds that have optionality. Therefore, the carry factor score can be defined as follows:

$$\text{car}_{i,t} := \text{Z-spread}_{i,t} \tag{2.16}$$

2.2.3 Credit Portfolio Metrics

Besides the profitability of factor investing strategies, it is important to manage the risk of the portfolio. There are multiple metrics available to evaluate portfolios, which we have narrowed down to a specific selection.

As discussed earlier, our primary metric for evaluating portfolio performance in terms of profitability is the excess return over LIBOR (see Section 2.2.1), referred to as excess return. At first glance, calculating the cumulative total return may seem appealing from a client perspective, as it directly shows the growth of invested wealth. However, for portfolio managers, excess return is a more relevant measure as it reflects the performance beyond the risk-free component of the bond. Another common profitability measure is the information ratio (IR). The information ratio is a risk-adjusted return measure of a financial asset or portfolio compared to a benchmark. It is calculated by dividing the expected value of active return $\mathbb{E}[R^p - R^b]$ of the portfolio by the tracking error σ:

$$\text{IR} = \frac{\mathbb{E}[R^p - R^b]}{\sigma} \tag{2.17}$$

where σ is the standard deviation of the active return, R^p the return of the portfolio, and R^b the return of the benchmark. The IR is particularly useful for portfolio managers as it quantifies the ability to generate excess returns relative to the benchmark, adjusted for the risk taken. A higher IR indicates a more efficient use of risk resulting in better risk-adjusted performance.

In addition to return measures, there are several risk-related metrics used to evaluate a credit portfolio. Asset managers cannot arbitrarily construct portfolios but strive to build portfolios that are optimal within the risk constraints. Therefore, we introduce the following additional key portfolio metrics, some of which are discussed in Section 2.2.1:

- **Duration**
  In this thesis we use *modified duration* for duration which is not be confused with Macaulay duration. The duration measures the sensitivity of the bond's price to changes in interest rates. For portfolio managers, duration is crucial as it helps manage interest rate risk. A longer duration implies greater sensitivity to interest rate changes, making it an essential metric for constructing a portfolio that aligns with the fund's risk tolerance.

- **Z-spread**
  A spread measure represents the compensation over the risk-free swapcurve with the same maturity. To complement the measure for excess return, we selected Z-spread as our spread measure. The Z-spread is the constant spread that, when added to the risk-free rate at each point on the yield curve, equates the present value of the bond's cash flows to its market price. It is particularly interesting for portfolio managers as it captures the additional yield a bond offers over the risk-free rate, accounting for credit risk, liquidity, and other factors.

- **Time to Maturity (TTM)**
  Time to maturity contains information on the expiry of the bonds. Bonds with longer maturity often have higher yield, reflecting the increased risk of holding a bond for a longer period. For a portfolio manager, understanding the time to maturity distribution within the portfolio is vital as it impacts the portfolios yield and risk profile.

- **Rating**
  The rating of a bond reflects the creditworthiness of the issuer, as assessed by rating agencies. Bonds with higher ratings (e.g., AAA) are considered safer but offer, in general, lower yields, while lower-rated bonds (e.g., BBB) offer higher yields due to higher default risk. For portfolio managers, the credit rating is a critical metric to ensure that the portfolios overall credit quality aligns with the fund's risk management policies. Maintaining an appropriate average rating helps in achieving a desired risk-return balance. To calculate an average rating, a numerical mapping is required. In this thesis we assigned the numerical values as follows: AAA $\mapsto$ 1, AA $\mapsto$ 2, A $\mapsto$ 3, BBB $\mapsto$ 4.

3 | Mathematical Preliminaries for Causal Discovery

In the application of this thesis, we focus on causal discovery using i.i.d. data rather than time-series data. Consequently, this chapter presents the mathematical preliminaries of causal discovery on independent and identically distributed (i.i.d.) data, which can be extended to time-series data. We primarily make use, regarding i.i.d. causal discovery, of the lecture notes of Peters [38], the paper by de Prado et al. [29], and two recent surveys on causal discovery methods of Hossain et al. [19] and Zanga et al. [47].

In Section 3.1, we introduce the reader to the concept of causality with an intuitive example. Then, in Section 3.2, we define the fundamental mathematical concepts for causal discovery. Section 3.3 elaborates on causal discovery methods, which are subdivided into constraint-based methods (Section 3.3.1) and score-based methods (Section 3.3.2). For each of these two categories, we consider one algorithm in this thesis. The complete causal discovery algorithms, Fast Causal Inference (FCI) (constraint-based) and Greedy Equivalence Search (GES) (score-based), are presented in Sections 3.4 and 3.5, respectively.

3.1 Cause and Effect

'Correlation does not imply causation' is a well-known phrase that highlights the difference between association and cause-and-effect relationships between two events. An observed correlation between events A and B does not necessarily imply that A causes B. A famous example, given in Peters [38], is the incorrect causal statement that 'Eating chocolate produces Nobel prizes.' This statement was popularized by some news platforms based on Messerli's [32] research, which found a correlation between a country's chocolate consumption and the number of Nobel prizes awarded (see Figure 3.1). The statement suggests that if every citizen were randomly forced by the government to eat a certain amount of chocolate, there would be a dependence between chocolate consumption and Nobel prizes. However, we know this is not true; the correlation could instead be explained by hidden variables such as economic strength.

As mentioned in the introduction, de Prado et al. [29] argues that model misspecification can lead to the underperformance of factor or forecasting strategies. We introduce the reader to model misspecification through two intuitive examples adopted from the article of de Prado et al. [29]: undercontrolling for a confounder and overcontrolling for a collider.

Undercontrolling for a confounder

Let us consider the data-generating process as an example:

$$X := Z\delta + v \tag{3.1}$$

$$Y := X\beta + Z\gamma + u \tag{3.2}$$

where $\gamma, \delta \in \mathbb{R}/\{0\}$ and u, v, Z are independent and identically distributed standard normal variables. Here, let X represent the factor (cause) and Y represent the return (effect), with β

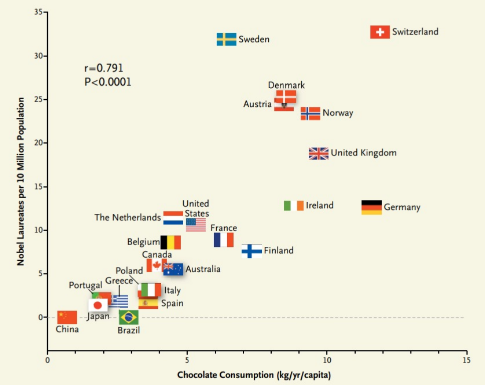


Figure 3.1: Correlation between chocolate consumption (kg/y/capita) and the number of Nobel Prizes (per 10 million inhabitants). Adopted from Messerli [32].

being the risk premium of this factor.[1] In causal language, Z is a *confounder* because it influences both the cause and the effect. This data-generating process can be represented as a graph in Figure 3.2, where the green arrow shows the effect of X on Y.

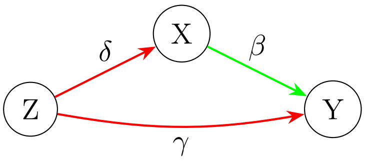


Figure 3.2: Example data-generating process including a confounder. Adopted from de Prado et al. [29].

In practice, the data-generating process is unknown to observers. With knowledge of the causal structure, an investor would specify the model $Y = X\beta + Z\gamma + u$ to estimate the effect of X on Y. Incorrectly specifying the model, in this case *undercontrolling* for a confounder by excluding variable Z, an investor would use $Y = X\beta + \epsilon$ as the model. In Section 3.1 of De Prado et al. [29], they prove that an under-controlled model will underperform in both factor and forecasting strategies.

Overcontrolling for a Collider

Similarly, overcontrolling for a collider can also have negative effects. Let us consider the following data-generating process:

$$Y := X\beta + u, \tag{3.3}$$

$$Z := Y\gamma + X\delta + v, \tag{3.4}$$

where $\gamma, \delta \in \mathbb{R}/\{0\}$ and u, v, X are independent and identically distributed standard normal variables. Again, X is the cause and Y is the effect. In causal terms, Z is a *collider* because it is influenced by both the cause and the effect. This data-generating process is represented as a graph in Figure 3.3, where the green arrow indicates the effect of X on Y.

With accurate knowledge of the causal structure, an investor would specify the model $Y = X\beta + \epsilon$ to estimate the effect of X on Y. Incorrectly specifying the model, referred to as *overcontrolling* by including variable Z, would result in the use of $Y = X\beta + \epsilon$ as the model. For this example, de Prado et al. [29] demonstrate in Section 3.2 that an overcontrolled model will underperform in both factor and forecasting strategies due to the collider effect δ.

[1] This example is adapted from de Prado et al. [29], where X represents $X_{i,t}$, the factor exposure of an asset X. In our example, X represents the factor score.

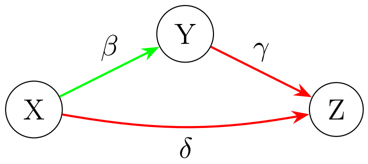


Figure 3.3: Example data-generating process including a collider. Adopted from de Prado et al. [29].

Observational Data and Causal Discovery

As discussed in the introduction of this thesis, interventional methods can be used in experimental sciences to estimate cause-effect relationships. For example, consider a classic medical experiment involving two identical groups of patients to test a drug's effectiveness. Unknown to the doctors, the treatment group receives the drug, while the control group receives a placebo. This double-blind setup ensures that neither the participants nor the researchers know who receives the drug, thereby reducing biases. Notably, being aware of receiving the actual drug or placebo could act as a confounder in this experiment.

However, in finance and factor investing, direct interventions like those in the previous example are not feasible. Therefore, in this case, identifying causal relationships relies on observational data, complicating the process. We resort to *causal discovery* methods, which learn the causal structure from observational data (see Definition 1.1) using mathematical properties from graph theory.

3.2 Definitions and Notation

This section is used to present definitions and theorems required for causal discovery and familiarize the reader with common notation. Causal discovery is heavily based on graph theory, of which we review important definitions first.

Graph Theory

Definition 3.1 (Graph)**.** *A graph* $G = (\boldsymbol{V}, \boldsymbol{E})$ *consists of vertices* $\boldsymbol{V}$ *and edges* $\boldsymbol{E} \subseteq \boldsymbol{V} \times \boldsymbol{V}$ *with* $(X, X) \notin \boldsymbol{E}$ *for any* $(X, X) \in \boldsymbol{V}$

Definition 3.2 (Directed Graph)**.** *A directed graph (DG) is a graph of which edges are directed: every edge* (X, Y) *is distinct from edge* (Y, X)

Definition 3.3 (Mixed Graph)**.** *[48] A mixed graph contains both directed, undirected edges, and bi-directed edges.*

More specifically, undirected edges can be graphically represented as $X - Y$, directed edges as $X \rightarrow Y$, and bi-directed edges as $X \leftrightarrow Y$. Intuitively, for any given vertex X, the set of vertices that have a directed edge going into X are called the *parents* of X and are denoted by $Pa(X)$. The vertices that have an edge into them from X, are called the *children* of X and are denoted by $Ch(X)$. For a bi-directed edge $X \leftrightarrow Y$, X and Y are called a *spouse*, of each other. The spouse set of X is denoted by $Sp(X)$ and contains all spouses of X. For an undirected edge $X - Y$, X and Y are called *neighbors* of each other. Vertices that are connected to X are *adjacent* without specifying the orientation of the edge.

Definition 3.4 (Path)**.** *A path* $\pi = (X - \cdots - Y)$ *is a sequence of distinct vertices where each vertex is connected to the next vertex with an edge. If these edges are directed, this becomes a directed path* $\pi = (X \rightarrow \cdots \rightarrow Y)$*.*

When considering a directed path containing $(\cdots \to X \to \cdots \to Y \to \cdots)$, we call Y a *descendant* of X. Continuing in this language X the *ancestor* of Y.

Definition 3.5 (Cycle). *A cycle is path where the starting vertex is the same as the ending vertex i.e.* $(X \to \cdots \to X)$

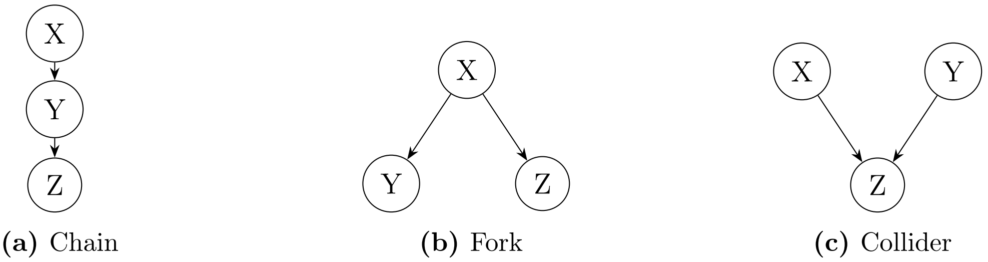
Definition 3.6 (Directed Acyclic Graph). *A directed acyclic graph (DAG) is a directed graph that does not contain cycles.*

Definition 3.7 (Partially Directed Acyclic Graph). *A partially directed acyclic graph (PDAG) is a graph that contains both directed and undirected edges*

We will now introduce the important definition *d-separation* which is used to define graphical independence. The formal definition can be found in Definition 3.9. Simply stating, X and Y are d-separated by set $\boldsymbol{Z}$ if $\boldsymbol{Z}$ *blocks* every path π between X and Y. To understand the d-separation criterion more thoroughly, we discuss three frequently observed fundamental key structures in a causal graph. The three key structures are a *Chain*, *Fork*, and *Collider* (See Figure 3.4).

Definition 3.8 (Chain, Fork and Collider). *Let G be a graph and let π be a path on G. For any three vertices X, Y and Z in π we define the following structures in π:*

- *Chain:* $X \to Y \to Z$
- *Fork:* $X \leftarrow Y \to Z$
- *Collider:* $X \to Y \leftarrow Z$



(a) Chain **(b)** Fork **(c)** Collider

Figure 3.4: Key structures of a causal graph

Now that these three key structures are defined, the definition of d-separation can be presented.

Definition 3.9 (D-separation). *[37] Let G be a directed graph and π a path on G. Let A, B and C be three vertices of G and let $\boldsymbol{Z} \subset \boldsymbol{V}$. π is blocked by $\boldsymbol{Z}$ if and only if π contains*

- *a fork $A \leftarrow B \to C$ or a chain $A \to B \to C$ such that the middle vertex B is in $\boldsymbol{Z}$,*
- *a collider $A \to B \leftarrow C$ such that the middle vertex B, or any descendent of B, is not in $\boldsymbol{Z}$.*

If $\boldsymbol{Z}$ blocks every path between X and Y then $\boldsymbol{Z}$ d-separates X from Y.

To illustrate d-separation we present an example graph included in Figure 3.5. In this graph, A and B are d-separated, without conditioning on any vertex, because they from a collider; middle vertex C and its descendants are not Z which means Z is the empty set. Therefore A and B are always d-separated. However for A and D, we have a chain structure $A \to C \to D$ on the path $(A \to C \to D)^1$ which means the middle vertex C d-separates A and D.

[1] The path itself is a chain structure

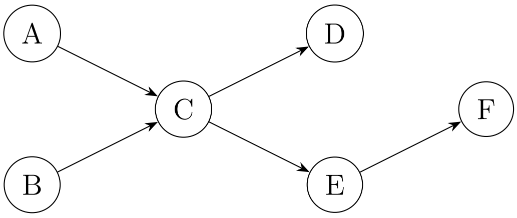


Figure 3.5: Example graph to illustrate d-separation

Causal Graphs

A *Causal Graph* is a graphical representation of a system (of variables) in terms of cause-effect relations. The vertices of the causal graph represent the variables of the system variables $\boldsymbol{V} = (X_1, X_2, \ldots, X_K)$. The edges represent the causal relationship between the variables.

Definition 3.10 (Causal Graph)**.** *[19] [38] A causal graph (CG) is a DAG that represent a joint probability distribution P over a set of random variables $\boldsymbol{V} = (X_1, X_2, \ldots, X_K)$ where P satisfies the* ***Markov property*** *with respect to the graph G.*

The Markov property can be distinguished into three type of definitions: global, local and factorization.

Definition 3.11 (Markov property)**.** *[38] Let G be a graph and a P joint distribution over a set of random variables $\boldsymbol{V} = (X_1, \ldots, X_K)$. This distribution is said to satisfy*

1. *the* ***global Markov property*** *with respect to the DAG G if*

$$\boldsymbol{X_i, X_j} \text{ d-separated by } \boldsymbol{X_k} \;\Rightarrow\; \boldsymbol{X_i} \perp_P \boldsymbol{X_j} \mid \boldsymbol{X_k} \tag{3.5}$$

for all disjoint sets $\boldsymbol{X_i}$, $\boldsymbol{X_j}$, $\boldsymbol{X_k}$. Here, $\perp_P$ represents conditional independence in probability.

2. *the* ***local Markov property*** *with respect to the DAG G if each variable is independent of its non-descendants given its parents.*

3. *the* ***Markov factorization property*** *with respect to the DAG G if*

$$p(\boldsymbol{V}) = p(X_1, \ldots, X_K) = \prod_{j=1}^{K} p(X_j \mid Pa(X_j)) \tag{3.6}$$

(here, we have to assume that P has a density p).

If the joint distribution P has a probability density, the three definitions are equivalent. Thus, if considering a causal graph, all of the three Markov properties hold.

Theorem 3.1. *[38] If P has a density p, then all Markov properties in Definition 3.11 are equivalent.*

Proof. See Theorem 3.27 of Lauritzen [28]. □

Conditional independence is a crucial tool to find causal relations among variables. The global Markov property, statement 1 in Definition 3.11), conveys that d-separation implies conditional independence i.e.

$$X \perp_G Y \mid \boldsymbol{Z} \Rightarrow X \perp_P Y \mid \boldsymbol{Z}. \tag{3.7}$$

Here, we denote d-separation as $\perp_G$ for graphical independence. This direction of the implication is embedded in the definition of a causal graph. The other direction, conditional independence in probability implies d-separation, is an important causal assumption (see Assumption 1) which is known as *faithfulness*:

$$X \perp_G Y \mid \boldsymbol{Z} \Leftarrow X \perp_P Y \mid \boldsymbol{Z}. \tag{3.8}$$

Equivalence Classes

Causal discovery algorithms (Section 3.3) cannot always identify the causal structure in the form of a DAG. Often, these algorithms are limited in finding other graphical representations.

Definition 3.12 (Skeleton)**.** *Let G be a PDAG. The skeleton of G is the undirected graph obtained by chancing all directed edges into undirected edges.*

Definition 3.13 (V-structure)**.** *Let G be a PDAG. A v-structure in G is a structure $X \to Y \leftarrow Z$ where X and Y are not adjacent.*

A v-structure is similar to a collider (see Definition 3.8). V-structures encode the conditional independencies of the conditional probability distribution as stated in the article of Zanga et al. [47]. Any that edge that creates a new v-structure or deletes an existing v-structure by changing orientation is called a *compelled edge*. Every edge that is not compelled is called *reversible*.

In 1990, Verma and Pearl [45] proved the following lemma:

Lemma 3.1 (Markov Equivalent)**.** *[45] Let G and H be two DAGs. G and H are Markov equivalent, denoted by $G \equiv H$, if they have*

- *the same skeleton,*
- *the same v-structures.*

Definition 3.14 (Markov Equivalence Class)**.** *[47] Let G and H be two DAGs. G and H are belong to the same Markov equivalence class (MEC), denoted as $[G]$, if they are Markov equivalent. This set of causal graphs has the same conditional independencies.*

In Figure 3.6 we present an example of two graphs that are Markov equivalent. Note that indeed they have the same skeleton and the same v-structure $Z \to V \leftarrow U$.

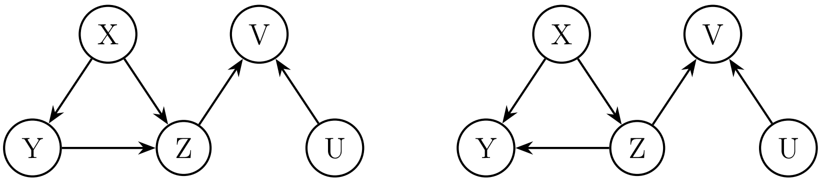


Figure 3.6: Two Markov-equivalent graphs. Adopted from Peters [38].

Equivalence classes can be represented by a CPDAG.

Definition 3.15 (Completed Partially Directed Acyclic Graph)**.** *[19] A Completed Partially Directed Acyclic Graph (CPDAG) is a PDAG that is completed. That is, the CPDAG consists of directed edges that exist in every DAG that have the same conditional independencies and, undirected edges that are reversible in G.*

As an example, using the definition of a CPDAG, we can represent the equivalence class of the two graphs of Figure 3.6 into one CPDAG shown in Figure 3.7.

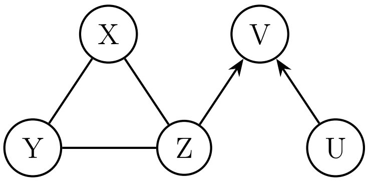


Figure 3.7: Equivalence class of graphs of Figure 3.6 represented by a CPDAG

Ancestral Graphs

Another type of graph is an *ancestral graph* (AG) which is a special case of a mixed graph. Ancestral graphs can be used to represent a data-generating system where there exist hidden confounders (see Definition 3.20).

Definition 3.16 (Ancestral Graph (AG))**.** *[48] A mixed graph G is ancestral if the following conditions holds*

1. *There does not exist a directed cycle in G*
2. *There does not exist a almost directed cycle in G*
3. *For any undirected edge $X - Y$, X and Y have no parents or spouses*

The first and second property induce ancestral hierarchy by removing cycles. The second property contains an *almost directed cycle*, which occurs when $X \leftrightarrow Y$ in G and Y is an ancestor of X. The third property requires that there are no edges going in an edge that has an undirected edge.

Mixed graphs, and thus ancestral graphs, focus on the endpoints of edges instead of the complete edge itself. A mixed graph can have two different endpoints: $-$ or $>$. When the endpoint is generic, which means it can be both $-$ or $>$, an asterisk $*$ is used. In the context of ancestral graphs we wish to define m-separation instead of d-separation. It is necessary to redefine a collider as $(X {*\!\!\rightarrow} Y \leftarrow\!\!* Z)$. A structure that is not a collider is called a *non-collider.*

Definition 3.17 (M-separation)**.** *[47] Let G be a mixed graph, π be a path on G and $\boldsymbol{Z} \subset \boldsymbol{V}$. The path π is blocked by $\boldsymbol{Z}$ if one of the following properties holds:*

- *π contains a non-collider such that the middle vertex of the non-collider is in $\boldsymbol{Z}$*
- *π contains a collider such that the middle vertex, or any descendant of it, is not in $\boldsymbol{Z}$*

$\boldsymbol{Z}$ m-separates X from Y if it blocks every path between X and Y.

Definition 3.18 (Maximal Ancestral Graph (MAG))**.** *[47] [48] A ancestral graph is maximal if any pair of non adjacent vertices are m-separated*

In the most general form, suppose there exists a true causal DAG G over a set of variables $\boldsymbol{V} = \boldsymbol{O} \cup \boldsymbol{L} \cup \boldsymbol{S}$. Here, $\boldsymbol{O}$ represents the set of observed variables, $\boldsymbol{L}$ denotes the set of hidden/unobserved variables (possibly confounders), and $\boldsymbol{S}$ denotes the set of unobserved selection variables to be conditioned upon. A selection variable is a variable that keeps track whether the sample remains in the study/causal discovery or is omitted (see Zhang [48]). Selection variables are often used in medical studies. Given any DAG G over $\boldsymbol{V} = \boldsymbol{O} \cup \boldsymbol{L} \cup \boldsymbol{S}$, there exists a MAG, denoted by, over the observed variables only $\boldsymbol{O}$ that *probabilistically represents* the DAG G. In Theorem 4.18 of Richardson et al. [39] they prove that for any DAG G a MAG can be constructed such that the MAG probabilistically represents G. A MAG that probalistically represents G is denoted by $\mathcal{M}_G$. We refer the interested reader to the work of Zhang [48] for the full construction procedure.

Two different MAGs can have different causal information, but trough equivalence and m-separation, they can represent the same conditional independence constraints. Trough the definition of *discriminating paths*, equivalence of MAGs can be proved[1]. The equivalence class for MAGs can be represented by a *partial ancestral graph.* Here, the endpoint $\circ$ means that the endpoint is not (yet) determined.

Definition 3.19 (Partial Ancestral Graph)**.** *[47] Let G be an ancestral graph. G is a partial ancestral graph if it can have*

[1] For the proof see Sprites et al. [43]

- *directed edges* ($\rightarrow$)*,*
- *bidirected edges* ($\leftrightarrow$)*,*
- *undirected edges* ($-$)*,*
- *partially directed edges* ($\circ\!\!\rightarrow$)*. That is, each edge can have one of three different endpoints:* $-$, $\circ$, *or* $>$

Zanga et al. [47] describe the interpretation of PAGs as follows:

- $X \rightarrow Y$: X causes Y. There could exist a hidden confounder,
- $X \circ\!\!\rightarrow Y$: Y is not an ancestor of X. That is, X causes Y or there is an hidden confounder,
- $X \leftrightarrow Y$. There is a hidden confounder that causes both X and Y. X does not cause Ynor does Y cause X,
- $X \circ\!\!-\!\!\circ Y$ One of the following statements is true: X causes Y or Y causes X; here is a hidden confounder that causes both X and Y; both $X \rightarrow Y$ and $X \circ\!\!\rightarrow Y$; both $X \leftrightarrow Y$ and $X \circ\!\!\rightarrow Y$.

3.3 Causal Discovery

Let $\boldsymbol{G}$ be the set of all possible graphs over the set of variables $\boldsymbol{V}$ in the dataset $\boldsymbol{D}$. Causal discovery methods aim to recover the 'true' graph $G^* \in \boldsymbol{G}$ from the given dataset $\boldsymbol{D}$. For real-life observational data, it cannot be verified if the discovered graph is indeed the true graph. However, using *synthetic* data generated from a known system, causal discovery algorithms can be tested for accuracy. Although causal discovery methods strive to identify the DAG G^* from the data $\boldsymbol{D}$, they are sometimes restricted to finding other graphical representations, such as equivalence classes.

In practice, hidden variables present common challenges. Hidden variables are unobserved variables that are not included in $\boldsymbol{V}$ but influence variables within $\boldsymbol{V}$. These hidden variables are referred to as *hidden confounders.*

Definition 3.20 (Hidden Confounder)**.** *A hidden confounder X, is a variable that is has an effect on any subset of $\boldsymbol{V}$ and is **not** included in $\boldsymbol{V}$ itself.*

Causal discovery algorithms often rely on several causal assumptions. These assumptions are discussed first. Subsequently, we introduce two of the most common approaches to recovering the causal graph from observational data: constraint-based methods and score-based methods.

Causal Assumptions

Causal discovery algorithms often makes use of certain causal assumptions regarding the variables or the data.

Assumption 1 (Faithfulness)**.** *The faithfulness assumption states that all variables that are not d-separated, are dependent; conditional independence in probability implies graphical independence via d-separation. Mathematically,*

$$X \perp_P Y \mid \boldsymbol{Z} \Rightarrow X \perp_G Y \mid \boldsymbol{Z}. \tag{3.9}$$

Assumption 2 (Sufficiency)**.** *The sufficiency assumptions states that there are no hidden confounders, and that all variables that may have common cause are also includes as variables.*

Assumption 3 (Acyclicity)**.** *There are no cycles in a causal graph*

AGs are useful when aiming to relax the causal sufficiency assumption to representing the underlying data-generating processes that may include hidden confounders as a causal structure, even without directly modeling these unobserved variables.

3.3.1 Constraint-based Methods

Constraint-based methods make use of conditional independence test to uncover the causal structure. This relies on the faithfulness assumption (Assumption 1) of the underlying distribution. Trough this assumption, one can translate the conditional independence test into d-separation, and therefore exclude edges between variables that can be d-separated. When the graph also satisfies the Markov property i.e. is a causal graph, then we have a perfect map:

Definition 3.21 (Perfect Map)**.** *[47] A graph G is a perfect map for a probability distribution P if every conditional independence statement from P can be derived from G and vice versa:*

$$X \perp_P Y \mid \boldsymbol{Z} \iff X \perp_G Y \mid \boldsymbol{Z}. \tag{3.10}$$

Testing for conditional independence is well-known statistical procedure which we briefly introduce. Let the null hypotheses H_0 and the alternative hypotheses H_1 be defined as

$$H_0 : X \perp_P Y \mid \boldsymbol{Z}, \tag{3.11}$$

$$H_1 : X \not\perp_P Y \mid \boldsymbol{Z}. \tag{3.12}$$

The conditional independence test statistic is defined as $T(X, Y \mid \boldsymbol{Z})$. In general form, the p-value for a two sided test with test statistic T and observed value of the test statistic $\hat{T}$ is given by

$$\text{p-value} = P(|T| \geq \hat{T} \mid H_0). \tag{3.13}$$

For a two-sided hypothesis test, the zero hypothesis H_0 is rejected if the p-value of the test statistic is smaller than a significance level depending on α:

$$\text{p-value } \leq 2 \cdot \alpha \Rightarrow X \not\perp_P Y \mid \boldsymbol{Z}. \tag{3.14}$$

For instance, if the test statistic is assumed to be standard normal under the H_0 hypothesis, then we can reject the zero hypothesis H_0 if

$$|\hat{T}| > \Phi^{-1}\left(1 - \frac{\alpha}{2}\right). \tag{3.15}$$

3.3.2 Score-based Methods

Score-based methods search over the space of all possible DAGs, denoted by $\mathcal{G}$, to find optimal graph G^* based on a certain scoring criterion $S(G, \boldsymbol{D})$. This can mathematically be expressed as

$$G^* = \underset{G \in \mathcal{G}}{\text{argmax}}\ S(G, \boldsymbol{D}). \tag{3.16}$$

These methods often consists of 2 key components:

1. A search strategy
2. A score function

The search strategy determines how the algorithm will explore possible search states of graphs G. The score function is used to asses the possible graphs G; it maps causal graphs to a numerical score based on how well it fits the data. The graph or equivalence class of graphs that have the highest score function is the output of the algorithm. To understand the score-based algorithms more rigorously, necessary properties and definitions are introduced in the following few paragraphs.

Definition 3.22 (Decomposable Score)**.** *[7] Let A scoring criterion $S(G, \boldsymbol{D})$ is decomposable if it can be defined as a sum of the scores of subgraphs of a vertex and its parents:*

$$S(G, \boldsymbol{D}) = \sum_{X_i \in \boldsymbol{V}} S(X_i, Pa(X_i), \boldsymbol{D}) \tag{3.17}$$

If a score function is decomposable the score computation can be efficiently evaluated through local differences of the causal graph. During causal discovery, previously computed scores of subgraphs can be used perform score calculations efficiently. In particular, the comparison of scores of two DAGs G and H can be handled by taking into account only the vertices that have different parent sets. Frequently used score functions, stated in the article of Barba et al. [1] include the Bayesian Information Criterion (BIC), the Minimum Description Length (MDL), and the Bayesian Dirichlet equivalent (BDe). Two score functions can be equivalent according to the following definition:

Definition 3.23 (Equivalent Score)**.** *[47] A scoring criterion $S(G, \boldsymbol{D})$ is score equivalent if for any pair of graphs G and H that are in the same equivalence class, $S(G, \boldsymbol{D}) = S(H, \boldsymbol{D})$.*

A graph G is said to *contain* a probability distribution P if there exists an (conditional) independence model associated with G that represents P exactly (see Zanga et al. [47]). That is, G is a perfect map of P (see Definition 3.21).

Definition 3.24 (Consistent Score)**.** *[47] Let $\boldsymbol{D}$ be a data set associated with a probability distribution P and random variables $(X_i, \ldots, X_K)$ and let G and H be two graphs. A scoring criterion S is said to be consistent in the limit of the number of samples the following two properties hold:*

- *If H contains P and not G does not contain P, then $S(G, \boldsymbol{D}) < S(H, \boldsymbol{D})$,*
- *If both G and H contain P and the model associated with H has fewer parameters than the one with G, then $S(G, \boldsymbol{D}) < S(H, \boldsymbol{D})$.*

Lemma 3.2. *[7] BIC, MDL and BDe are score equivalent and consistent.*

A score function can also be locally consistent which desired to efficiently compute local scores.

Definition 3.25 (Locally Consistent Score)**.** *[7] Let G be a graph and G' the graph resulting from the addition of the edge $X \to Y$ to G. A scoring criterion $S(G, \boldsymbol{D})$ is said to be locally consistent if the two following conditions hold:*

1. $X \not\perp_P Y \mid Pa(Y) \implies S(G, \boldsymbol{D}) < S(G', \boldsymbol{D})$.
2. $X \perp_P Y \mid Pa(Y) \implies S(G, \boldsymbol{D}) > S(G', \boldsymbol{D})$,

A score that is locally consistent score can be interpreted as follows. If an edge is added that eliminates an independence constraint not contained in the generative distribution P it favors the addition. Adding an 'unnecessary' edge that adds an independence constraint will not be favored by the logical consistent score.

Lemma 3.3. *[7] BIC is locally consistent.*

Proof. See the proof Lemma 7 of Chickering et al. [7] □

Without loss of generality, the BIC is used in this thesis as it is a locally consistent score.

Definition 3.26 (Bayesian Information Criterion (BIC))**.** *[7] Let D be the data set consisting of m i.i.d. samples from distribution P. The Bayesian Information Criterion (BIC) can be defined as*

$$S_{BIC}(G, \boldsymbol{D}) = \log P(\boldsymbol{D} \mid \hat{\boldsymbol{\theta}}, G^h) - \frac{d}{2} \log m \tag{3.18}$$

where

- *$\hat{\boldsymbol{\theta}}$ denotes the maximum likelihood values for the parameter values that specify all conditional probability distributions*
- *d denotes the number of free parameters[1] of graph G*
- *m number of i.i.d. samples from data D*
- *$P(\boldsymbol{D} \mid \hat{\boldsymbol{\theta}}, G^h)$ maximized value of the likelihood function. Here, the use G^h with h for hypothesis denotes the assumption that the observed data contains i.i.d. samples from the distribution P that exactly matches the conditional independence relations i.e. a perfect map.*

The likelihood function often is computed under the assumption that the samples are linear Gaussian.

3.4 Fast Causal Inference (FCI)

In the application of Causal Factor Investing in this thesis, Fast Causal Inference (FCI), introduced by Sprites et al. [42], is the first of two causal discovery algorithms used. It is an adaption of the most-studied constraint-based PC algorithm which is named after its inventors Peter and Clark [42]. The FCI algorithm uses conditional independence tests and extends the PC algorithm trough relaxing the causal sufficiency assumption (Assumption 2). When omitting the causal sufficiency assumption the causal structure may not be representable by a DAG or CPDAG as it is statistically hard problem (see Zhang [48]). Moreover, incorporating the effect of hidden confounders in a model implies that the graph search space is infinite unless the amount of hidden confounders is significantly constrained. If the identification of the hidden confounder themselves is not of main interest, the use of ancestral graphs provides a solution.

It is helpful to understand the concept of the PC algorithm before the FCI algorithm is discussed in more detail. The goal of the PC algorithm is to learn the causal DAG from observational data under the causal sufficiency assumption (Assumption 2). The PC algorithm is based on conditional independence test and therefore it is necessary that causal faithfulness (Assumption 1) and the Markov property hold (Definition 3.11). This leads to a perfect map, Definition 3.21, of graphical independence in the causal graph trough d-separation and conditional independence in probability. However, in general, d-separation does not uniquely determine a single DAG G but a Markov equivalence class $[G]$. A Markov equivalence class $[G]$ can uniquely be represented by a Completed Partially Directed Acyclic Graph (CPDAG). The main idea of the PC algorithm is to test for conditional independence between variables translate this via d-separation into a causal graph.

The FCI algorithm relaxes the causal sufficiency assumption. Suppose we have a true causal DAG G over observed and unobserved variables. It is necessary that the causal faithfulness assumption and the Markov property hold as for the PC algorithm. Through these assumptions, we obtain a perfect map between conditional independence in probability and graphical independence through d-separation of causal DAG G (or its equivalence class). As described in Section 3.2, this DAG G can be probalistically represented by MAG $\mathcal{M}_G$ on the observed variables.

[1] A variable that is not determined by the model itself but is left to be assigned (possible optimally)

This means that there is a correspondence of conditional independence in probability of the observed variables and m-separation of the graph $\mathcal{M}_G$. Similar as the PC algorithm, m-separation relations do not uniquely determine a single MAG, but represent a Markov equivalence class $[\mathcal{M}_G]$. An equivalence class of $\mathcal{M}_G$ can be uniquely represented by a PAG (see Definition 3.19). Therefore, the main idea of FCI is to use conditional independence tests and m-separation to learn the causal structure of observational data. The complete algorithm is given in Algorithm 1 and a visualization of the FCI algorithm is displayed in Figure 3.8.

We will describe the procedure of FCI in the following sections. In the algorithm $\texttt{Adjacency}_G(X)$ is the adjacency set of X on the graph G and $\texttt{Sepset}(X, Y)$ is a set that will contain the set of vertices that d-separates X and Y. The FCI algorithm consist of three main steps: (1) Skeleton identification, (2) V-structures determination, (3) and orientation of undirected edges. Following Zhang et al. [48] we use $\mathcal{R}$ as the symbol for a orientation rule. More precisely, the second step (2) can be seen as the first orientation rule $\mathcal{R}0$. Furthermore we give the complete orientation rules $\mathcal{R}1$ and $\mathcal{R}2$ in the third step. The remainder of the orientation rules $\mathcal{R}3 - \mathcal{R}10$ can be found the paper of Zhang [48]. Recall that in a PAG we have 3 type of endpoints for an edge: $-, \circ,$ and $>$. A $\circ$ represents an undetermined mark. When stating orientation rules we make use of an aterisk $*$ meaning it is generic and denotes any of the endpoints.

1. **Skeleton Identification** (lines 1-12, Algorithm 1)
   The algorithm starts with a fully connected graph containing every variable and every edge $\circ\!\!-\!\!\circ$. Then, with conditional independence tests, it tests for conditional independence in probability for any pair, say X and Y, on every set $\boldsymbol{S}$. If, for any $\boldsymbol{S}$, X is conditionally independent of Y given $\boldsymbol{S}$, $\boldsymbol{S}$ is added to $\texttt{Sepset}(X, Y)$ (line 7). By the faithfulness assumption and probabilistic representation by MAGs, conditional independence in probability implies graphical separation (m-separation). Therefore, if $\texttt{Sepset}(X, Y)$ is not empty, one removes the edge $X - Y$ from the original graph (line 8). By repeating this for every pair of vertices, a skeleton remains.

2. **V-structures determination** (lines 13-17, Algorithm 1)
   In the next step, *v-structure identification*, as the name suggests, it aims to find and orient the v-structures (unshielded colliders) which is also referred to as the first orientation rule $\mathcal{R}0$. This is done on the previously found skeleton using the separations sets. For any triplet $X \circ - \circ Z \circ - \circ Y$ with X not adjacent to Y, if Z m-separates X and Y via a v-structure meaning $Z \notin \texttt{Sepset}(X, Y)$, orientate the triplet as a v-structure: $X {*\!\!\rightarrow} Z \leftarrow\!\!* Y$.

3. **Orientation of undirected edges** (lines 18-27, Algorithm 1)
   Now it remains to orient the edges. This is done by iteratively repeating orientation rules $\mathcal{R}1 - \mathcal{R}10$ until none of them applies. Orientation rule $\mathcal{R}1$ uses the assumption that all v-structures have been detected during the previous step. Therefore, orientate triplets in such a way to avoid new v-structures (line 19-21). Orientation rule $\mathcal{R}2$ is based on the acyclicity assumption (Assumption 3). So, orient triplets in such a way to avoid cycles (lines 22-24). Then proceed with the other orientation rules $\mathcal{R}3 - \mathcal{R}10$ as in the work of Zhang [48].

Algorithm 1 Pseudocode FCI Algorithm. Adaption of PC algorithm from Zhu [49].

```
Input: A dataset $\boldsymbol{D}$ of the set of observational variables $\boldsymbol{V}$, and a conditional independence test
    method
Output: The graph $G$
 1: Begin with the fully connected partial ancestral graph $G$ on $\boldsymbol{V}$ with edges $\circ\!-\!\circ$;
 2: $n \leftarrow 0$;
 3: for Each adjacent pair $X \circ\!-\!\circ Y$ with $|\texttt{Adjacency}_G(X)\backslash Y| \geq n$ or $|\text{Adjacency}_G(Y)\backslash X| \geq n$
    do
 4:     for Any $\boldsymbol{S} \subseteq \texttt{Adjacency}_G(X)\backslash Y \cup \texttt{Adjacency}_G(Y)\backslash X$ and $|\boldsymbol{S}| = n$ do
 5:         Test whether $X$ and $Y$ are conditionally independent given $\boldsymbol{S}$;
 6:         if $X \perp\!\!\!\perp Y \mid \boldsymbol{S}$ then
 7:             $\texttt{Sepset}(X, Y) \leftarrow \boldsymbol{S}$;
 8:             Delete the edge $X \circ\!-\!\circ Y$ in $G$;
 9:             $n \leftarrow n + 1$;
10:         end if
11:     end for
12: end for
13: for Each triple of vertices $X, Y, Z$ do
14:     if $X - Z$ and $Y - Z$ are adjacent and $(X, Y)$ are not adjacent in $G$ then
15:         Orient $X \circ\!-\!\circ Z \circ\!-\!\circ Y$ as $X {*\!\!\rightarrow} Z \leftarrow\!\!* Y$ if $Z \notin \texttt{Sepset}(X, Y)$ [$\mathcal{R}0$: V-structures
    Detection];
16:     end if
17: end for
18: for Vertices in $\mathbf{V}$ (until none of them applies) do
19:     if $X, Y$ are not adjacent and $X {*\!\!\rightarrow} Z \circ\!\!-\!* Y$ in $G$ then
20:         Orient $Z \circ\!\!-\!* Y$ as $Z \rightarrow Y$ [$\mathcal{R}1$: Known Non-V-structures Detection];
21:     end if
22:     if $X {*\!\!\rightarrow} Y \rightarrow Z$ or $X \rightarrow Y {*\!\!\rightarrow} Z$, and $X *\!\!-\!\circ Z$ then
23:         Orient $X *\!\!-\!\circ Z$ as $X {*\!\!\rightarrow} Y$ [$\mathcal{R}2$: Cycle Avoidance];
24:     end if
25:     Apply further orientation rules $\mathcal{R}3 - \mathcal{R}10$ [48]
26: end for
```

One can see that the time complexity is exponential, as it depends on the growth of conditioning set S (line 4). To test for conditional independence of any pair (X, Y), the algorithm needs to test $\mid 2^{V/\{X,Y\}} \mid$ conditioning sets shown in the article of Zanga et al. [47].

Furthermore, it is of essence to select a good candidate for the conditional independence test to check for conditional independence (Section 3.3). A frequently used test is the Fisher's Z conditional independence test (see Kalisch [27]). Let X, Y be random variables and $\boldsymbol{Z}$ as set of random variables such that $X, Y \notin \boldsymbol{Z}$. The two-sided Fisher's Z conditional independence test is used to test the null hypothesis H_0 against the alternative hypothesis H_1 given by

$$H_0 : \rho_{XY|\boldsymbol{Z}} = 0, \tag{3.19}$$

$$H_1 : \rho_{XY|\boldsymbol{Z}} \neq 0. \tag{3.20}$$

Here, $\rho_{XY|\boldsymbol{Z}}$ is the partial correlation between X and Y given $\boldsymbol{Z}$. Under the null hypothesis, the test statistic T_{Fisher} follows a standard normal distribution $N(0, 1)$. The T_{Fisher} is the Fisher's Z statistic is given by

$$T_{\text{Fisher}} = \frac{1}{2} \ln\left(\frac{1 + \hat{\rho}_{XY|\boldsymbol{Z}}}{1 - \hat{\rho}_{XY|\boldsymbol{Z}}}\right) \sqrt{n - |\boldsymbol{Z}| - 3} \tag{3.21}$$

where $\hat{\rho}_{XY|\boldsymbol{Z}}$ is the sample partial correlation coefficient, n is the sample size, and $|\boldsymbol{Z}|$ is the number of conditioning variables in $\boldsymbol{Z}$. The test concludes by comparing the absolute value of

T_{Fisher} to the critical value from the standard normal distribution for the desired significance level α. In particular, the zero hypothesis H_0 will be rejected if

$$|T_{\text{Fisher}}| > \Phi^{-1}(1 - \alpha/2). \tag{3.22}$$

where Φ the cumulative density function of a standard normal $N(0,1)$.

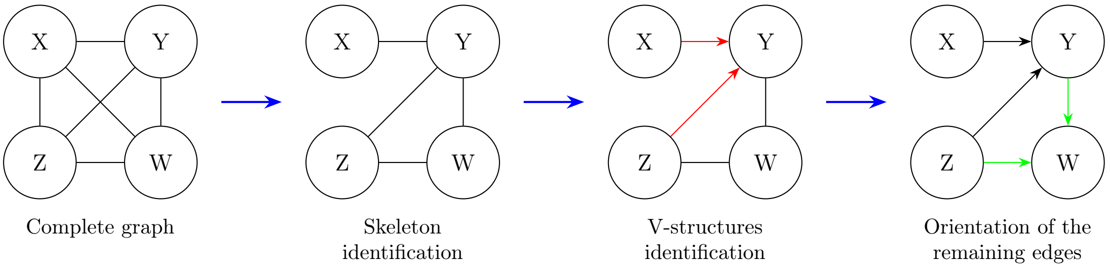

X
Y
Z
W
Complete graph
Skeleton identification
V-structures identification
Orientation of the remaining edges

Figure 3.8: Simplified visualization of the FCI algorithm. Adopted from Hossain [19].

3.5 Greedy Equivalence Search (GES)

Different from constraint-based methods, score-based methods rely on an optimization approach. These methods search over all possible graphs to find the one that best matches the data using a numerical score. For more information on score-based methods, we refer the reader to Section 3.3.2. Greedy Equivalence Search (GES) is the main algorithm in this category. As the name suggests, the search strategy of GES is greedy, where it performs a greedy search over the state space of equivalence classes of DAGs and outputs a CPDAG. A common score function for GES is the Bayesian Information Criterion (BIC), defined in Definition 3.26.

For the formal details of the GES algorithm, including the formal definition of the operators used in the algorithm, we refer the reader to the original work of Chickering et al. [7] and Barba et al. [1]. These formalities include the definitions of the operators `Insert` and `Delete` using neighboring sets. Additionally, by employing Theorem 15 of Chickering et al. [7], they provide a method to test the validity of the `Insert` and `Delete` operators to preserve the equivalence class through so called *consistent extensions*.

GES can be divided into three sub-algorithms. The first is the structure of GES itself (Algorithm 2). The other two sub-algorithms are the two steps of GES: Forward Equivalence Search (FES) (Algorithm 3) and Backward Equivalence Search (BES) (Algorithm 4). The pseudocode of GES is adopted from Barba et al. [1], with the notation for the equivalence class $\mathcal{E}$ changed to $[G]$ to align with the notation used in this thesis.

GES (Algorithm 2) starts with an empty CPDAG G, i.e., a CPDAG without edges, which represents an equivalence class $[G]$ (line 2). It then performs the two search strategies: Forward Equivalence Search (FES) (line 3) and Backward Equivalence Search (BES) (line 4).

1. **Forward Equivalence Search (FES)** (Algorithm 3)
   FES iteratively applies $FS([G])$ which does the following three things: it gives the edge with the highest improvement of the score function (line 3), inserts this edge with `Apply Insert` (line 4), and converts the new created graph in a CPDAG with the function `PDAGtoCPDAG` (line 5). It repeats this procedure until the FS does not return any edge that increase the score function; it has reached a local maximum.
   FS loops greedily over all possible edge additions. It first checks if the `Insert` operation is valid with neighbouring set $\boldsymbol{T}$ of Y and definition of *clique* for which we refer the reader to Theorem 15 of Chickering et al. [7]. If valid, it computes the increment of the scoring metric Δ (line 15) of the newly created subgraph by adding an edge with the `Insert`

function. Because the score function is decomposable, the calculation simplifies (Corollary 16, Chickering et al. [7]). It checks if the increment is better than the best increment in this iteration, and if true it updates the best increment (line 17) and saves the best edge the edge (line 18).

2. **Backward Equivalence Search (BES)** (Algorithm 4)
   After FES, Backward Equivalence Search (BES) is conducted in similar fashion that greedily deletes edges. The `Delete` operator is used instead of the `Insert` operator. Also, `Apply Insert` is changed into `Apply Delete`.

Algorithm 2 GES. Adopted from Barba et al [1].

```
procedure GES
    $[G] \leftarrow$ Empty graph $G$ ($\mathbf{E} = \emptyset$)
    $[G] \leftarrow$ FES($[G]$)
    $[G] \leftarrow$ BES($[G]$)
    return $[G]$
end procedure
```

Algorithm 3 FES. Adopted from Barba et al. [1].

```
 1: procedure FES($[G]$)
 2:     repeat
 3:         $(X \rightarrow Y, \mathbf{T}) = \mathrm{FS}([G])$
 4:         $[G] =$ Apply Insert$(X, Y, \mathbf{T})$ to $[G]$
 5:         $[G] =$ PDAGtoCPDAG($[G]$)
 6:     until $(X \rightarrow Y = \mathrm{null})$
 7:     return $[G]$
 8: end procedure
 9:
10: function FS($[G]$)
11:     edge $\leftarrow$ null; best $\leftarrow 0$
12:     for all $X \in \mathbf{V}$ do
13:         for all $Y \in \mathbf{V} \mid (Y \neq X) \wedge Y$ is not adjacent to $X$ do
14:             for all $\mathbf{T} \subseteq \mathbf{T}_0 \mid \mathrm{Test}(X \rightarrow Y, \mathbf{T}) = $ true do
15:                 Compute $\Delta =$ Insert$(X, Y, \mathbf{T})$
16:                 if $\Delta >$ best then
17:                     best $= \Delta$
18:                     edge $= (X \rightarrow Y)$; subset $= \mathbf{T}$
19:                 end if
20:             end for
21:         end for
22:     end for
23:     return $(X \rightarrow Y, \mathbf{T}) = (\mathrm{edge}, \mathrm{subset})$
24: end function
```

From FES (Algorithm 2), it holds that time complexity is exponential on the size of set $\boldsymbol{T_0}$ as stated in the article of Barba et al. [1]. Simply stating, it is exponential on the number of variables (vertices). More precisely, the total number of search states (graphs) found by GES can never exceed $n \cdot (n-1)$ where n is the number of variables as shown in the article of Chickering [7]. For the interested reader, Chickering [7] shows in his article that GES reaches a local maximum[1].

[1] Chickering [7] shows this by proving a certain conjecture: Meek's Conjecture.

Algorithm 4 BES. Adopted from Barba et al. [1].

```
procedure BES([G])
    repeat
        $(X \to Y, \mathbf{H}) = \mathrm{BS}([G])$
        $[G] =$ Apply Delete$(X, Y, \mathbf{T})$ to $[G]$
        $[G] =$ PDAGtoCPDAG$([G])$
    until $(X \to Y = \mathrm{null})$
    return $[G]$
end procedure

function BS([G])
    edge $\leftarrow$ null; best $\leftarrow 0$
    for all $X \in \mathbf{V}$ do
        for all $Y \in \mathbf{V} \mid (Y \neq X) \wedge Y$ is adjacent to $X$ do
            for all $\mathbf{H} \subseteq \mathbf{H}_0 \mid \mathrm{Test}(X \to Y, \mathbf{H}) = \mathrm{true}$ do
                Compute $\Delta =$ Delete$(X, Y, \mathbf{H})$
                if $\Delta >$ best then
                    best $= \Delta$
                    edge $= (X \to Y)$, subset $= \mathbf{H}$
                end if
            end for
        end for
    end for
    return $(X \to Y, \mathbf{H}) =$ (edge, subset)
end function
```

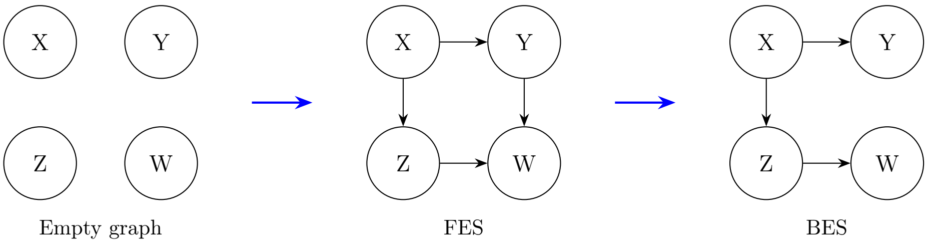


Figure 3.9: Visualization of the GES algorithm

4 | Causal Factor Investing

This section contains the details of Causal Factor Investing (CFI) which is novelly developed in this thesis. The methodology of CFI is constructed generically, such that CFI can be used for different asset classes under different circumstances. Figure 4.1 presents a simplified visual representation of Causal Factor Investing. Let us revisit the three steps of CFI:

1. **Causal Analysis:** Perform causal discovery on factors and returns to obtain causal factors.
2. **Empirical asset pricing via machine learning:** Predict future returns with machine learning methods using the causal factors as features. The predicted future returns will serve as composite signal.
3. **Investment strategy:** Use the signal, the predicted future returns, to invest and construct the portfolio.

The experimental setup for causal discovery, using factors and returns, is discussed in Section 4.1. Through the causal structure, we seek to identify causal factors. In Section 4.2, we define the empirical asset pricing problem via machine learning as an additive prediction error model. Section 4.3 details the investment strategy, which is an adaptation of the traditional factor investing strategy. This section also contains the full algorithm of CFI.

Furthermore, we use the same notation as defined in Chapter 2. A factor portfolio is denoted by F_t^k, representing the value (or return) of the factor portfolio at time t, where $t \in \{t_0, \ldots, T\}$ of factor k with $k \in \{1, \ldots, K\}$. A factor score $f_{i,t}^k$ represents the value of the factor characteristic k of asset i, where $i \in \{1, \ldots, I\}$ at time t. The return of asset i at time t is denoted by $r_{i,t}$.

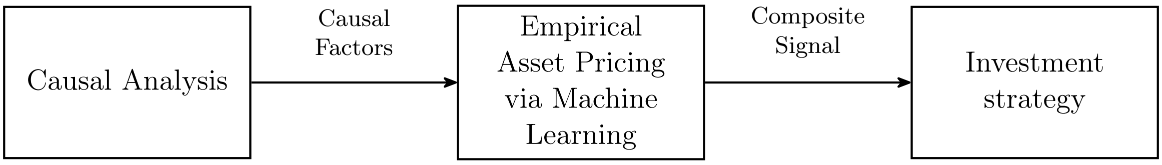


Figure 4.1: Simplified visual representation of CFI

4.1 Causal Discovery on Factors and Returns

The goal of the causal analysis is to uncover the causal structure of factors and returns using causal discovery methods. The resulting causal graph will be used to determine the causal factors used in the second step of CFI: empirical asset pricing via machine learning (see Section 4.2). In this section, terminology related to causal discovery is frequently used. For further details on causal discovery, we refer the reader to Chapter 3, which explains the necessary mathematical preliminaries.

In Section 4.1.1, we will specify the problem mathematically and discuss potential approaches. Numerous causal discovery algorithms exist for various types of problems, each employing different causal assumptions. The following subsection, Section 4.1.2, provides an overview of possible causal discovery algorithms for the different approaches. In the final subsection, Section 4.1.3, we define causal factors and introduce the algorithm used to determine these from the causal graph.

4.1.1 Setting and Approaches

The causal discovery problem can be defined as recovering true graph G^* given dataset $\boldsymbol{D}$. Causal discovery algorithms can be split into two categories based on the type of data: independent and identically distributed (i.i.d.) data and time-series data. We propose the following different mathematical approaches, some of which are based on methods found in literature, to perform causal discovery on factors and returns.

1. **i.i.d. asset samples of factor scores and returns**
   In this case, the data $\boldsymbol{D}$ is assumed to consist of i.i.d. assets samples. An asset sample has the form $(f^1, f^2, \ldots, f^K, r) := (f^1_{i,t}, f^2_{i,t}, \ldots, f^K_{i,t}, r_{i,t})$ for $i \in \{1, \ldots, I\}$ and $t \in \{t_0, \ldots, T\}$ fixed. The set of vertices $\boldsymbol{V}$ can be defined as
   $$\boldsymbol{V} = (f^1, f^2, \ldots, f^K, r). \tag{4.1}$$
   Thus, we consider causal discovery on i.i.d. factor scores and returns of individual assets.

2. **i.i.d. asset samples of factor scores and lagged returns**
   This approach differs from the previous trough one single adaption. Instead of including the return at time t, the lagged return at time $t+1$ is used. The sample is defined as $(f^1, f^2, \ldots, f^K, r^{\text{lagged}}) := (f^1_{i,t}, f^2_{i,t}, \ldots, f^K_{i,t}, r_{i,t+1})$ for $i \in \{1, \ldots, I\}$ and $t \in \{t_0, \ldots, T-1\}$ fixed. The set of vertices $\boldsymbol{V}$ is defined as
   $$\boldsymbol{V} = (f^1, f^2, \ldots, f^K, r^{\text{lagged}}). \tag{4.2}$$
   An argument for this approach is that cause-effect relations are identified between factors and **future** returns.

3. **i.i.d. samples of factor portfolio and returns with aggregation of graphs**
   In this case we consider i.i.d. samples. As proposed in the article of Gu et al. [17], these samples consist of the factor portfolios and asset returns. A single sample from data $\boldsymbol{D}$ is now defined as $(F^1, F^2, \ldots, F^K, r) := (F^1_t, F^2_t, \ldots, F^K_t, r_{i,t})$ for $i \in \{1, \ldots, I\}$ and $t \in \{t_0, \ldots, T\}$ fixed. Here, for i fixed, the amount of samples for a single asset is equal to the amount of time observations T of this asset. Causal discovery is performed on the set of vertices
   $$\boldsymbol{V} = (F^1, F^2, \ldots, F^K, r). \tag{4.3}$$
   Because factor portfolios have equal values a fixed point in time t, causal graphs are derived for each asset individually. Then, the graphs are aggregated into one graph by assigning weights to the number of appearances of each directed edge. Directed edges with weights above a certain threshold are kept in the aggregated graph.

4. **Time series of factor portfolios and market return**
   One can also perform time-series causal discovery on the factor portfolios and market return r^m which is done by D'Acunto et al. [12]. In this case, we have a multi-dimensional time series object $(F^1, F^2, \ldots, F^K, r^m)_t := (F^1_t, F^2_t, \ldots, F^K_t, r^m_t)$. The set of vertices of the time-series causal discovery can be defined as
   $$\boldsymbol{V} = (F^1, F^2, \ldots, F^K, r^m). \tag{4.4}$$

5. **Time series of factor portfolios and asset return**
   Similar as the previous case, time-series causal discovery can be done on factor portfolios and individual asset return as done by Sadeghi et al. [41]. In our context of factors and returns, we have a multi-dimensional time series object $(F^1, F^2, \dots, F^K, r)_t := (F^1_t, F^2_t, \dots, F^K_t, r_{i,t})$ for $i \in \{1, \dots, I\}$ fixed. The set of vertices during time-series causal discovery for a fixed asset i can be defined as

   $$\boldsymbol{V} = (F^1, F^2, \dots, F^K, r). \tag{4.5}$$

   The causal structure of factor portfolios and the return of a single asset can be found. If one is only interested in a single graph, there exists an option combine graphs of all assets into one aggregated graph.

6. **Time series of factor scores and asset return**
   Time-series causal discovery can be done on factor scores and individual asset return. In this case you consider the factor scores as a time series object. In a slightly different context, considering causal relations of assets returns among themselves, Tang [44] also argue for multidimensional time-series causal discovery. In our context of factors and returns, we have a multi-dimensional time series object $(f^1, f^2, \dots, f^K, r)_t := (f^1_{i,t}, f^2_{i,t}, \dots, f^K_{i,t}, r_{i,t})$ for $i \in \{1, \dots, I\}$ fixed. The set of vertices during time-series causal discovery for a fixed asset i can be defined as

   $$\boldsymbol{V} = (f^1, f^2, \dots, f^K, r). \tag{4.6}$$

   The causal structure of factor scores and a return of single asset can be found. If one is only interested in a single graph, there exists an option combine graphs of all assets into one aggregated graph.

One of the goals of CFI is to create a profitable investment strategy. An important argument using factor scores on individual asset level (approach 1, 2, and 6), is that it aligns with the next two steps of CFI: empirical asset pricing via machine learning and the investment strategy. The training of the machine learning models and the investment strategy itself do not consider factor portfolios, but factor scores of individual assets. The machine learning models, discussed in Section 4.2, use factor scores as features during training and prediction. The investment strategy, presented in 4.3, sorts the set of assets on factor scores. Therefore, through approach 1, 2, or 6, we aim to uncover the causal relations of the same variables that are used in the machine learning models and investment strategy, potentially leading to increased profitability. If the goal was merely to identify causal relations between factors and returns, factor portfolios could be of interest as well.

For approach 1 and 2, the use of i.i.d. causal discovery algorithms is necessary and this results in a single graph. The integration machine learning models can be done creating a single model using this single graph. For approach 6, time-series causal discovery is required. There are two options to integrate empirical asset pricing via machine learning. First, a single graph can be created trough aggregation of graphs of individual assets over time. This aggregated graph is used to create a single machine learning model. The second option is to use each individual graph, corresponding to a single asset, to create machine learning model for each specific asset as done by Tang [44]. For more advanced machine learning models, the training time will increase drastically as you will have to train as many models as there are assets.

When choosing between approach 1 and 2, we would argue for approach 2 as this captures the causal relations between the factor scores and the future returns. This aligns well with the other two steps of FCI. Namely, the machine learning models are trained to predict the future (lagged) returns and the investment strategy uses these predicted future (lagged) returns.

4.1.2 Causal Discovery Algorithms

As discussed in the previous section, there are different causal discovery approaches. Besides the approach, different causal discovery algorithms are available. There is no universally optimal causal discovery algorithm for every dataset. Causal discovery algorithms have been tested on synthetic datasets where the underlying data-generating process is known, and different algorithms performed better on different synthetic datasets (see Hossain et al. [19]).

Causal discovery algorithms are also tested on real datasets where the true causal graph remains unknown (see Hossain et al. [19], Nogueira et al. [34] and Morrafah et al. [33]). Depending on the chosen approach, as discussed in the previous subsection, either an i.i.d. or time-series causal discovery algorithm can be considered. For both types of data, a wide range of methods from different categories exists, as shown in Figure 4.2 for i.i.d. data and Figure 4.3 for time-series data. The most researched methods are constraint-based and score-based methods.

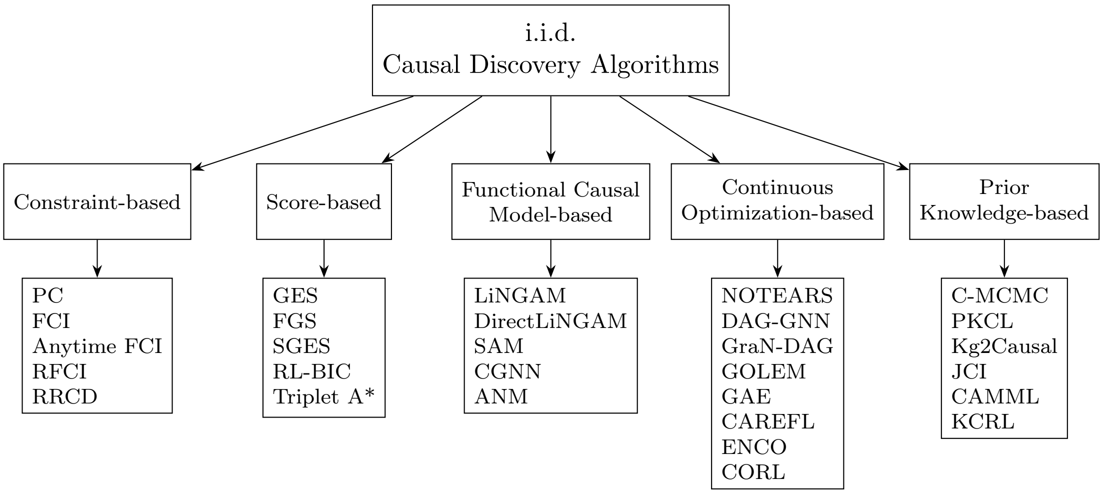


Figure 4.2: A collection of i.i.d. causal discovery methods. Adopted from Hossain et al. [19].

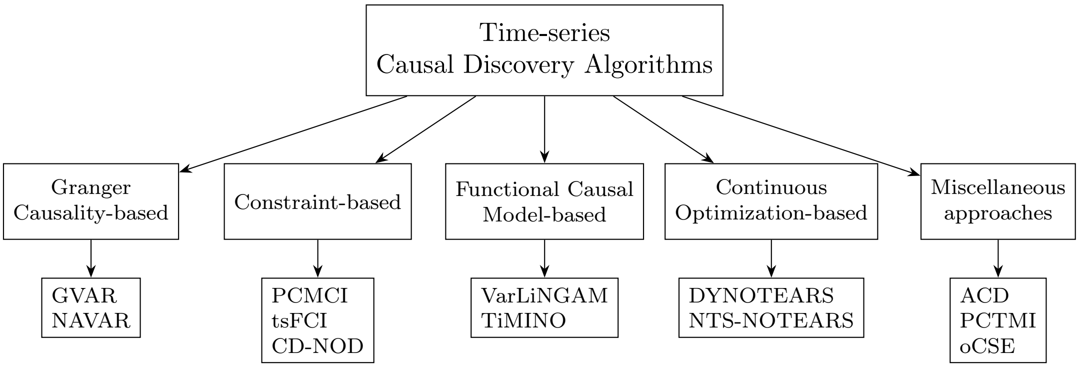


Figure 4.3: A collection of time-series causal discovery methods. Adopted from Hossain et al. [19].

Each causal discovery algorithms use certain assumptions which are discussed in Chapter 3. In the context of financial data, there may other unobserved variables that influence factors and/or returns. Therefore, some researchers aim to relax the causal sufficiency assumption (Assumption 2). For example, Sadeghi et al. [41] use time-series causal discovery method CD-NOTS, an adaption of CDNOD, to obtain the causal structure of factor portfolios and returns (approach 5). Here, they assume causal sufficiency by including a the variable of time in the set of vertices $\boldsymbol{V}$. Tang [44] uses Fast Causal Inference time series (tsFCI), derived from FCI, performing causal discovery on time-series data of stocks, which allows hidden confounders due to the use of partial ancestral graphs (variation of approach 4-5 using only stock data).

Some researchers incorporate the causal sufficiency assumptions. Gu et al. [17] use the an algorithm of three categories: PC for constraint-based, Greedy Equivalence Search (GES) for score-based, and LiNGAM for functional causal model-based. All of these algorithms do not allow for hidden confounders. D'Acunto et al. [12] use a time-series approach using causal sufficient VAR-LiNGAM on equity risk factor portfolios and market return (approach 4).

Depending on the amount size of the variable set or the number of causal discovery procedures, time complexity plays an important role. For high dimensional data, causal discovery algorithm can have high computational time. However during past two decades some of these algorithms also have been improved on computational time. For instance, Really Fast Causal Inference (RFCI) is an adaption of the Fast Causal Inference (FCI) algorithm improving computational time. When the amount of variables of the vertex set remains small, computational complexity is often not a problem.

4.1.3 Causal Factor Selection Algorithm

In this section we discuss the interpretation of the causal graph. With knowledge of the causal graph, colliders and confounders (see Section 3.1) can be identified. Excluding colliders and including observed confounders to your investing strategy could lead to improved performance. The graph, outputted by causal discovery, determines the input variables (features) for the second step of causal factor investing: empirical asset pricing via machine learning.

For this thesis, let us newly define a *causal factor* and *direct causal factor*. Both definition variations include possible confounders (not hidden) and exclude possible colliders.

Definition 4.1 (Causal Factor)**.** *Let $F \in \boldsymbol{V}$ be a factor and $R \in \boldsymbol{V}$ the future return. Let and G be a causal graph recovered by a causal discovery algorithm. Then, F is a causal factor if there exists a possible directed path π from F to R*

Definition 4.2 (Direct Causal Factor)**.** *Let $F \in \boldsymbol{V}$ be a factor and $R \in \boldsymbol{V}$ the future return. Let and G be a causal graph recovered by a causal discovery algorithm. Then, F is a direct causal factor if F is a possible parent of R i.e. $F \in Pa(R)$*

The word *possible* is used because causal graphs can have possible directed paths through undirected edges of which the orientation is not derived through the causal discovery algorithm. This is because several causal discovery algorithms output graphs represented by other graphical representation than a DAG such as an CPDAG or a PAG.

Let us denote the set of causal factors as $\boldsymbol{F}^{\text{causal}}$. Note that any direct causal factor is automatically a causal factor. Recall that due to the local Markov property (Definition 3.11) each variable is independent in probability of its non-descendants given its parents. That is,

$$F^2 \perp_P R \mid F^1 \tag{4.7}$$

where factor F^1 is a parent of the return R and factor F^2 is a non-descendant. This can be expressed by a visual example as in Figure 4.4.

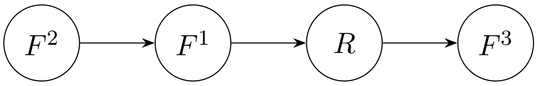


Figure 4.4: Example graph to illustrate local Markov property

In this figure F^2 is causal factor of R and F^1 is a direct causal factor of R. By the local Markov property it holds that F^2 has no causal effect on R if F^1 is known because the effect of F^2 on R is conveyed in F^1. Therefore, in theory, the definition of direct causal factor suffices to obtain all factors that effect the returns. However, in real-world problems, there could exist noise and/or there could be an effect of a causal factor on the return that is unmeasured by the causal discovery algorithm. It is still interesting to use the definition of causal factors as it still includes confounders and excluded colliders. In the application of this thesis, both definitions are tested numerically trough backtests.

As the factor selection from graphs is new, we propose a simple approach *causal factor selection* (Algorithm 5) that relies on (direct) causal factors.

Algorithm 5 (Direct) Causal Factor Selection

```
Input: Causal graph $G = (\boldsymbol{V}, \boldsymbol{E})$
if There are no (direct) causal factors then
    return all factors $\boldsymbol{F} \subset \boldsymbol{V}$
end if
return (Direct) causal factors $\boldsymbol{F}^{\text{causal}} \subset \boldsymbol{V}$
```

Later in this thesis, we will introduce an expanding window and moving window (see Section 4.2.2). Using such a window format, causal discovery is performed for multiple periods and is integrated with machine learning and the investment strategy. Therefore, in absence of causal factors, we return all factors $\boldsymbol{F}$ such that the investment strategy can continue. We are aware that omitting the causal graph is not preferred. If there are no (direct) causal factors, it suggests that the casual discovery algorithm did not find the correct graph. However, in absence of better solutions we resolve to this algorithm.

We demonstrate the Causal Factor Selection algorithm on an example graph in Figure 4.5 where R is the return. In this graph various possible directed paths are observed from factors to future returns: ($F^1 \rightarrow ... \rightarrow R$), ($F^2 \rightarrow ... \rightarrow R$), ($F^3 \rightarrow ... \rightarrow R$) and ($F^4 \rightarrow ... \rightarrow R$). However, there does not exist a directed path from ($F^5 \rightarrow ... \rightarrow R$) such that F^1, F^2, F^3, F^4 are causal factors and F^5 is not. The direct causal factors are the parents of R which are F^3 and F^4.

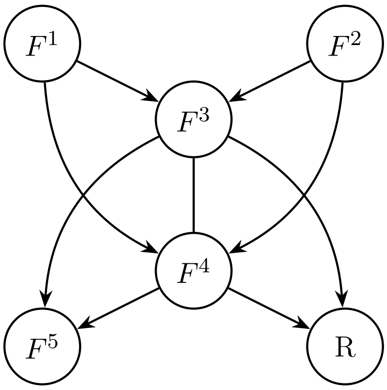


Figure 4.5: Example graph to illustrate (Direct) Causal Selection Algorithm

4.2 Empirical Asset Pricing via Machine Learning

The second component of the Causal Factor Investing (CFI) consists of the use of machine learning methods to price future asset returns in the cross section. In literature this is often referred to as *empirical asset pricing via machine learning*. For CFI to be causal, the features used in the machine learning are causal factors obtained via causal discovery algorithms. The output of empirical asset pricing, the predicted future return, will serve as a composite signal for the investment strategy of CFI.

In Section 4.2.1 the setup of the empirical asset pricing is explained and formulated mathematically. In the following section (Section 4.2.2) we elaborate on hyperparameter tuning. Section 4.2.3 is the final subsection in which we elaborate on the performance evaluation of machine learning methods.

4.2.1 Setting and Mathematical Formulation

Simply stating, the goal is to accurately predict future returns using a causal factors as features. In literature, empirical asset pricing is often done on a feature set of numerous characteristics instead of just factors as done by Gu et al. [18], Coqueret et al. [9], and Mansouri et al. [30]. Sometimes, available characteristics are assigned to a certain factor category as done by Mansouri et al. [30]. It could be possible to extend causal factor investing to causal characteristic investing. However, factors are seen as drivers, based on economic arguments, of cross-sectional returns. Causal relations seem more intuitive and explainable on such a set than on a large set of all available characteristics. In addition, due to the high dimensionality, the computational time of machine learning methods will be longer.

Let us define r as the return which we refer to as the *target variable*, z the set of *features*, and ϵ the error. The set of features z consists causal factor scores of an asset. That is, $\forall f$

$$z_{i,t} = (f^1, f^2, \dots, f^L)_{i,t} \quad \text{where } f \in \boldsymbol{F}^{\text{causal}} \tag{4.8}$$

It is important to note that the features $z_{i,t}$ are (direct) causal factor **scores** even if causal discovery with factor portfolios is used. Otherwise, using factor portfolios as features, every asset at a certain time point has the same features which is undesirable.

We can describe the future return of an asset as additive prediction error model as proposed by Gu et al. [18]. We have

$$r_{i,t+1} = \mathbb{E}_t[r_{i,t+1}] + \epsilon_{i,t+1} \tag{4.9}$$

where

$$\mathbb{E}_t[r_{i,t+1}] = g(z_{i,t}). \tag{4.10}$$

Here, assets are indexed with i and time with t. The goal is to find a representation of $\mathbb{E}_t[r_{i,t+1}]$ as a function of features $z_{i,t}$ to maximize the out-of-sample performance for predicted future returns $\hat{r}_{i,t+1}$. That is, we try to find $g(z_{i,t})$, a function of features, that minimizes the error ϵ. Note that the function g is defined in such a way that it neither dependent on i or t such that it has the same form over time and for all assets. This means that assets observations are considered in this form. In addition, the function g uses features $z_{i,t}$ which implies that the function g does not rely on features prior to time t nor on other assets j in the cross section.

Through different machine learning methods, we can approximate the expression g. Machine learning methods are suitable tools to address this problem as they have high predictive power, capture non-linear relations, and perform well out-of-sample if constructed thoughtfully taking overfitting into consideration.

4.2.2 Hyperparameter Tuning

It is important think about how choose subsamples of your dataset for training and testing. Furthermore, the selection of *hyperparameters* for the machine learning model is critical. Hyperparameters are parameters used for tuning the model; they allow to control the learning process and overfitting.

By regularly refitting a machine learning model it can incorporate most recent developments and have higher predictive power for future predictions. We propose a common approach, as done by Gu et al. [18]. For each time window, split the the data into three distinct time periods. The first period T_1 is the training phase where we train the model with different hyperparameter combinations. The subsequent period T_2 is used to validate the for tuning the hyperparameters. That is, we search for the optimal hyperparameters such that objective function, based on prediction forecast, has the lowest loss on the validation set. A frequently used objective function is the Mean Squared Error (MSE) which is also used the evaluate the performance of machine learning methods and is discussed in the next section. Then, the model is trained again on the combined set $T_1 \cup T_2$ with the optimal hyperparameters found in the previous step. The remaining segment is used as the testing period T_3 where we evaluate the out-of-sample performance with the optimal hyperparameters.

After this procedure, we move the window forward in time by Δt and repeat the process. The entire training period T_1 can be moved forward which preserves the length of the window. This approach is referred to as *moving window*. It is also possible to move only the end of the training period by Δt, allowing the training period to gradually expand as the window moves forward. This approach is referred to as the *expanding window*.

During backtesting, T_3 is used as the testing period. However, portfolio managers can adopt this strategy real-time, utilizing the machine learning model to forecast returns in the future period T_3 for their investments.

4.2.3 Performance Evaluation of Machine Learning Methods

The performance of the machine learning methods is evaluated by out-of-sample test. That is, we analyse the performance of the methods in T_3. One of the most intuitive evaluation methods is to inspect the error. This is done by computing the *Mean Squared Error* (MSE).

Definition 4.3 (Mean Squared Error)**.**

$$MSE = \frac{1}{N} \sum_{(i,t) \in T_3} \left(r_{i,t+1} - \hat{r}_{i,t+1}\right)^2 \tag{4.11}$$

In this case, N is the number of total asset observations (and predictions) in all test sets T_3. Besides the MSE, following Gu et al. [18], the *out-of-sample R-squared* (R^2_{OOS}) is used and is defined below.

Definition 4.4 (R^2 Out-of-sample)**.** *[18]*

$$R^2_{OOS} = 1 - \frac{\sum_{(i,t) \in T_3} \left(r_{i,t+1} - \hat{r}_{i,t+1}\right)^2}{\sum_{(i,t) \in T_3} r^2_{i,t+1}}. \tag{4.12}$$

The R^2_{OOS} statistic is a crucial measure in evaluating the performance of predictive models. It represents the proportion of the variance in the dependent variable that is predictable from the independent variables, assessed using data of test set T_3. This statistic provides a clear indication of how well the model generalizes to new, unseen data. A higher, close to one, R^2_{OOS} value signifies a model that more accurately captures the underlying relationships in the data. A subtle change of this R^2_{OOS} metric, compared to the original definition, is that the denominator is the sum of the squared excess return without demeaning. That is, we compare the predictive performance against the naive zero forecast value as done by Gu et al. [18] and Mansouri et al. [30].

4.3 Investment strategy

In the previous two steps we have identified (direct) causal factors and these (direct) causal factors are used to generate a machine learning signal. The last step of CFI contains the investment strategy itself. The traditional factor investing strategy, as discussed in Chapter 2, is slightly changed. Instead of sorting the set of financial instruments on a factor F, the set of assets is sorted on the predicted future returns $\hat{r}_{t+1}$. The investment strategy is now given as follows:

1. Sort the assets on **predicted future return** $\hat{r}_{t+1}$ at t
2. Long the top x % of assets and short the bottom x % of asset in your portfolio P
3. Repeat the process at $t+1$ through rebalancing portfolio P

Combining the three components, we present the full procedure of CFI in pseudocode given by Algorithm 6 .

Algorithm 6 Causal Factor Investing

```
Input: Set of assets, Machine learning method, Causal discovery algorithm, Factors of interest
    $\boldsymbol{F}$, Period $\Delta t$, End time $T$, Initial window $W_0 = [T_1, T_2, T_3]$
Output: Portfolio $P_T$
W ← W_0
while $T_3 \leq T$ do
    Use causal discovery algorithm on $\boldsymbol{F}$ and returns $r$ for combined training and validation period $T_1 \cup T_2$ to obtain graph $G$
    Obtain (direct) causal factors $\boldsymbol{F}^{\text{causal}}$ from graph $G$ using Algorithm 5 (Direct) Causal Factor Selection
    Train machine learning model on $T_1$ with features $\boldsymbol{F}^{\text{causal}}$ and the future returns of the assets as target variable
    Hyperparameter tuning on validation period $T_2$
    Train machine learning model on $T_1 \cup T_2$ with optimal hyperparameters
    for $t \in T_3$ do
        Predict future returns $\hat{r}_{t+1}$ at time $t$ with fitted machine learning method
        Sort set of assets on $\hat{r}_{t+1}$ at time $t$
        Rebalance portfolio $P$ by taking long position in top assets and optionally a short position in bottom assets
        $t \leftarrow t + 1$
    end for
    $W \leftarrow W + \Delta t$
end while
```

5 | Application of Causal Factor Investing in the Corporate Bond Market

In this chapter fit CFI to the application of this thesis: the European credit market. We will resort to five credit factors: size, value, momentum, low-risk, and carry. The motivation and specification of these factors can be found in Section 2.2.2. The numerical results of CFI, obtained via backtesting, of are stated in the next Chapter.

5.1 Causal Discovery on Credit Factors and Bond Returns

Approach

In Section 4.1 we discussed several approahes to perform causal discovery. We favoured causal discovery approach 1,2, or 6:

- 1. i.i.d. asset samples of factor scores and returns,
- 2. i.i.d. asset samples of factor scores and lagged returns,
- 6. Time series of factor scores and asset returns.

In the application of this thesis we will not use a time-series approach (approach 6) because of the following two reasons:

1. This thesis focuses on the corporate bond market. Bonds can reach expiry over time, meaning they disappear from the market, while new bonds can be issued at any time. This dynamic makes it difficult to perform time-series causal discovery for CFI. In CFI, causal discovery is performed for a fixed period. Time-series causal discovery would require omitting bonds that are issued or expire during this period, leading to incomplete data. Bonds with shorter lifetimes than the fixed period would never be included.
2. In the application of this thesis, we use monthly data. Using time-series causal discovery would result in fewer samples per bond. For instance, running causal discovery algorithms over a 5-year period yields only 60 observations, which is undesirable for many algorithms.

Because of the two arguments above we do not use time-series causal discovery and we resort to i.i.d causal discovery, approach 1 or 2, in the application of this thesis. As argued in Section 4.1 we prefer approach 2 over approach 1 as it aligns with the next steps of CFI. Therefore, we will use approach 2 in our numerical experiments. More specifically, we use bond samples $(siz_{i,t}, val_{i,t}, lr_{i,t}, mom_{i,t}, car_{i,t}, r^{\text{lagged}}_{i,t+1})$ where $i \in \{1, \ldots, I\}$ and $t \in \{t_0, \ldots, T-1\}$ fixed. Our set of vertices for causal discovery contains factor scores and lagged returns:

$$\boldsymbol{V} = (siz, val, lr, mom, car, r^{\text{lagged}}). \tag{5.1}$$

The numerical results of causal discovery can be found in Section 6.2.

Causal Discovery Algorithms in the Corporate Bond Market

To the best of our knowledge, this thesis is the first[1] to apply causal discovery on factors and returns within an investment strategy. We consider the two most researched categories in i.i.d. causal discovery: constraint-based and score-based methods.

As explained in Section 4.1, we aim to relax the causal sufficiency assumption due to the use of financial data. For constraint-based methods, we utilize Fast Causal Inference (FCI), which is an adaptation of the PC method. FCI relaxes the causal sufficiency assumption and performs well with large samples, as highlighted in the survey by Hossain et al. [19].

For score-based methods, the most popular method is Greedy Equivalence Search (GES). In contrast to FCI, GES requires the causal sufficiency assumption. To the best of our knowledge, there are no score-based methods that relax the causal sufficiency assumption. Score-based methods search over the equivalence space of all possible Directed Acyclic Graphs (DAGs) to find the graph that best explains the data using a score function. Due to the optimization-based nature of score-based methods, we believe GES provides valuable information in the form of a causal graph even though it does not allow hidden confounders.

Recall from Chapter 4 that causal discovery and fitting machine learning models are performed iteratively on the union of the training and validation set $T_1 \cup T_2$. During causal discovery, the (direct) causal factor scores are normalized by uniformization to the $[0, 1]$ interval. This normalization is necessary because the same set $T_1 \cup T_2$ is also normalized in the next step of CFI. This is done because fitting machine learning methods require normalized features, as done by Gu et al. [18]. Moreover, this normalization reduces the effect of outliers during causal discovery.

The results of the causal analysis can be found in Section 6.2. Here we state the (direct) causal factors that obtained trough causal discovery for each period. The graphs itself are given in Appendix B.

5.2 Empirical Pricing of Bonds Returns via Machine Learning

Recall from Section 4.2 that our goal is to accurately predict future returns using causal factors. In the application of this thesis, the target variable $r_{i,t+1}$ is defined as the month-ahead excess bond return:

$$r_{i,t+1} := r_{i,t+1}^{\text{excess return}}. \tag{5.2}$$

In the corporate bond market, the search for factors that explain the cross-section of returns has resulted in a variety of candidates, which we have narrowed down to five credit factors (see Chapter 2). In this thesis, we will use the factors size, value, low-risk, momentum, and carry. The factors that are causal (see Definitions 4.1 and 4.2) are used as features in the machine learning models. If all factors are causal, we can define the feature vector $z_{i,t}$ as

$$z_{i,t} := (siz_{i,t}, val_{i,t}, lr_{i,t}, mom_{i,t}, car_{i,t}). \tag{5.3}$$

Note that the composition of $z_{i,t}$ may change depending on the causal factors identified through causal discovery in the previous step of CFI.

We will focus on three popular machine learning methods from three different categories. The three methods are

1. Ordinary Least Squares (OLS),
2. XGBoost[2] (XGB) ,

[1] Tang [44] also uses causal discovery for an investment/trading strategy, but focuses on stocks among themselves.

[2] See Chen et al. [5]

3. Neural Networks (NN).

To evaluate effect of causal factors, we will also train the machine learning models using all factor regardless of the causal analysis as comparison which we will refer to as *regular*. Next, we aim to obtain feasible graphs via two causal discovery methods FCI and GES. For both causal discovery methods, we can obtain direct causal factors and causal factors. Thus, for each machine learning method, we aim to research five variations of CFI via backtesting:

1. **Regular:** Includes all available factors scores for each year as features,
2. **FCI:** Includes causal factors for each year as features obtained with FCI,
3. **FCI-Direct:** Includes direct causal factors for each year as features obtained with FCI,
4. **GES:** Includes causal factors for each year as features obtained with GES,
5. **GES-Direct:** Includes direct causal factors for each year as features obtained with GES.

In total we seek to predict the month-ahead returns with 15 different machine learning models. The results, the predictive performance of the machine learning models, are stated in Section 6.3.

As done by Coqueret et al. [9] and Gu et al. [18], it is common practice to scale features before using them in machine learning methods such as neural networks, as this leads to faster convergence. We follow Coqueret et al. [9], who use uniformization of the features within the [0,1] interval. At each point in time, each feature is uniformized across the cross-section of bonds in that specific month. They argue that it is important to scale features separately for each month and feature, resulting in scaled features that reflect information within each cross-section. This aligns with the investment strategy, where bonds are selected based on cross-sectional information. Furthermore, we have also normalized the target variable during the training periods.

Following Gu et al. [18], we employed an *expanding* window with the following specifications to fit our dataset:

- T_1: 5-13 years,
- T_2: 1 year,
- T_3: 1 year,
- Period Δt: 1 year.

Due to the size of our dataset, the first training period spans from 2009 to 2012, with the first validation period occurring in 2013. Therefore, we evaluate the total predictive performance from 2014 to 2023. In addition to the expanding window, we also applied a *moving* window, with the results included in Appendix C.

We now briefly introduce the three aforementioned machine learning methods and state the choice of possible *hyperparameters* and *architecture* in the case of neural networks. For each hyperparameter of interest, a range of possible values is chosen, which can be optimized during the hyperparameter tuning process (see Section 4.2.2). The architecture of a neural network refers to its structural design, including the number of layers, the number of neurons in each layer, and their connections, which together determine how the network processes input data to produce output.

Ordinary Least Squares (OLS)

OLS is the simplest form of a linear regression method (see Appendix A.1). For this method, there do not exist hyperparameters. OLS is included as a simple machine learning method to compare with the more advanced machine learning methods.

XGBoost (XGB)

XGB is one of the most popular tree-based machine learning methods. The mathematical concepts of tree-based machine learning methods can be found in Appendix A.2. Common hyperparameters for XGBoost are the learning rate, the number of trees, and the maximum depth of the trees. The range for the hyperparameters of XGBoost used in this thesis can be found in Table 5.1 and include commonly used values.

Table 5.1: Hyperparameters XGBoost

Learning rate	$\{0.01, 0.1\}$
Maximum depth	$\{3, 5\}$
# of trees	$\{100, 200, 500\}$

Neural Network (NN)

A neural networks (see Appendix A.3) are considered to be a very powerful machine learning method. We argue that we do not need a deep neural network with many hidden layers. Deeper neural networks often used for very complex tasks including many features. Also, a shallow neural network is more intuitive, and therefore explainable, than a deep neural network. This aligns with our goal to produce an explainable factor investing model. Therefore, a simple architecture for is proposed in this thesis: a shallow feed forward neural network with 1 hidden layer consisting of 8 perceptrons. The complete overview of hyperparameters of the neural network can be found in Table 5.2 and includes common values. The learning rate is low to prevent possible overfitting.

Table 5.2: Hyperparameters Neural Network

Learning rate	0.001
Epochs	100
Batch size	64
Patience	10
Activation function	ReLu
Optimizer	Adam

5.3 Investing Strategy in the Credit Market

The investment strategy for CFI uses predicted future returns as a composite signal. In our context, we use predicted month-ahead excess bond returns as the future returns. Therefore, the investment strategy has the following form:

1. Sort the corporate bonds based on predicted month-ahead excess returns $\hat{r}_{t+1}$ at month t,
2. Long on the top 10% of assets,
3. Repeat the process in the next month $(t + 1)$ by rebalancing the portfolio P.

Without loss of generality, we resort to the following parameters for the investment strategy. A standard value of 10% is used for rebalancing the portfolio. We choose a long-only portfolio due to liquidity constraints or client constraints from an asset managers perspective. Furthermore, we used equal weighting in the construction of the portfolio. Since our data is monthly, we use an investment horizon of one month, meaning we rebalance the portfolio every month.

The numerical results of backtesting traditional factor investing and CFI are presented in Sections 6.1 and 6.4, respectively. Portfolios are evaluated using the portfolio metrics discussed in Section 2.2.3.

For the traditional factor investing strategy, we also present a *decile analysis*, as done by Slimane et al. [3]. A factor portfolio consists of assets with the highest factor scores, as described in Section 2.1. It is also possible to create portfolios of assets that do not have the highest factor scores. In the decile analysis, we construct 10 portfolios using the following steps:

1. Sort the assets based on the factor score f at time t,

2. Divide the set into 10 deciles $D_1, \ldots, D_{10}$, and take a long position in these deciles to create decile portfolios $P_1, \ldots, P_{10}$,

3. Repeat the process at time $t+1$ by rebalancing portfolios $D_1, \ldots, D_{10}$.

The decile analysis functions as an intuitive method to compare the performance of different decile portfolios. Based on factor investing literature, we expect the first decile portfolio, which contains the bonds with the highest factor scores, to outperform the other deciles in terms of return.

6 | Numerical Results

This chapter presents the numerical results of backtesting Causal Factor Investing (CFI) in the European investment-grade credit market using an expanding window. All of the numerical results in this chapter do not consider transaction costs nor the feasibility of executing trades. Backtesting simulates the real-time execution of CFI using historical data, without knowledge of the true future values. In Appendix C, we provide the results of CFI employing a moving window. According to the experiments, the expanding window yields slightly better results compared to the moving window. Recall that we use the *Markit IBoxx Euro Corporate Senior Index*, introduced in Section 1.3, as the benchmark to compare the numerical results. We will refer to this index simply as the benchmark. To evaluate the portfolios we use the portfolio evaluation metrics introduced in Section 2.2.3. The portfolio metrics are duration, Z-spread, time to maturity (TTM), Rating, and the information ratio (IR).

Section 6.1 presents the results of traditional factor investing. The outcomes of the first step of CFI, the (direct) causal factors obtained, are stated in Section 6.2. Next, Section 6.3 contains the results of the second step of CFI: predicting future bond prices using machine learning methods. The third step of CFI is the investment strategy in which portfolios are constructed. The portfolio metrics are presented in Section 6.4.

6.1 Traditional Factor Investing

This section contains results of traditional factor investing. Section 6.1.1 provides an intuitive decile analysis of the five traditional factor portfolios. The portfolio performance of traditional factor investing is given in the next section (Section 6.1.2).

6.1.1 Decile Analysis

We will present the figures of the decile analysis for the five factors: carry, momentum, size, low-risk, and value. These figures will include the mean excess return of all decile portfolios, as well as the mean excess return of the benchmark, which is displayed as a red dashed line.

Carry

The results of the decile analysis for the carry factor are displayed in Figure 6.1a. The first decile clearly outperforms last decile portfolio. Bonds with high carry will generally have higher excess returns due to their higher risk profile. The last decile portfolio has a very weak performance. The decile analysis suggest that the carry factor is an important factor to consider.

Momentum

The results of the decile analysis for the momentum factor are displayed in Figure 6.1b. The last decile portfolio of the momentum factor stands out. It has a higher excess return than the benchmark and outperforms the first decile portfolio significantly. This suggest bonds that have been underperforming in the last six months, tend generate higher excess returns in the (lagged) month afterwards. The first decile surprisingly does not outperform the benchmark. Slimane et al. [3] also observe high returns of bonds with low momentum factor scores. They use the definition amended momentum to deal with high returns of low momentum decile portfolios (see Section 2.2.2).

Size

The results of the decile analysis for the size factor are displayed in Figure 6.1c. Here, the first and the last few decile portfolios outperform the benchmark minimally. We do not have convincing evidence that the first factor portfolio outperforms the last decile portfolio.

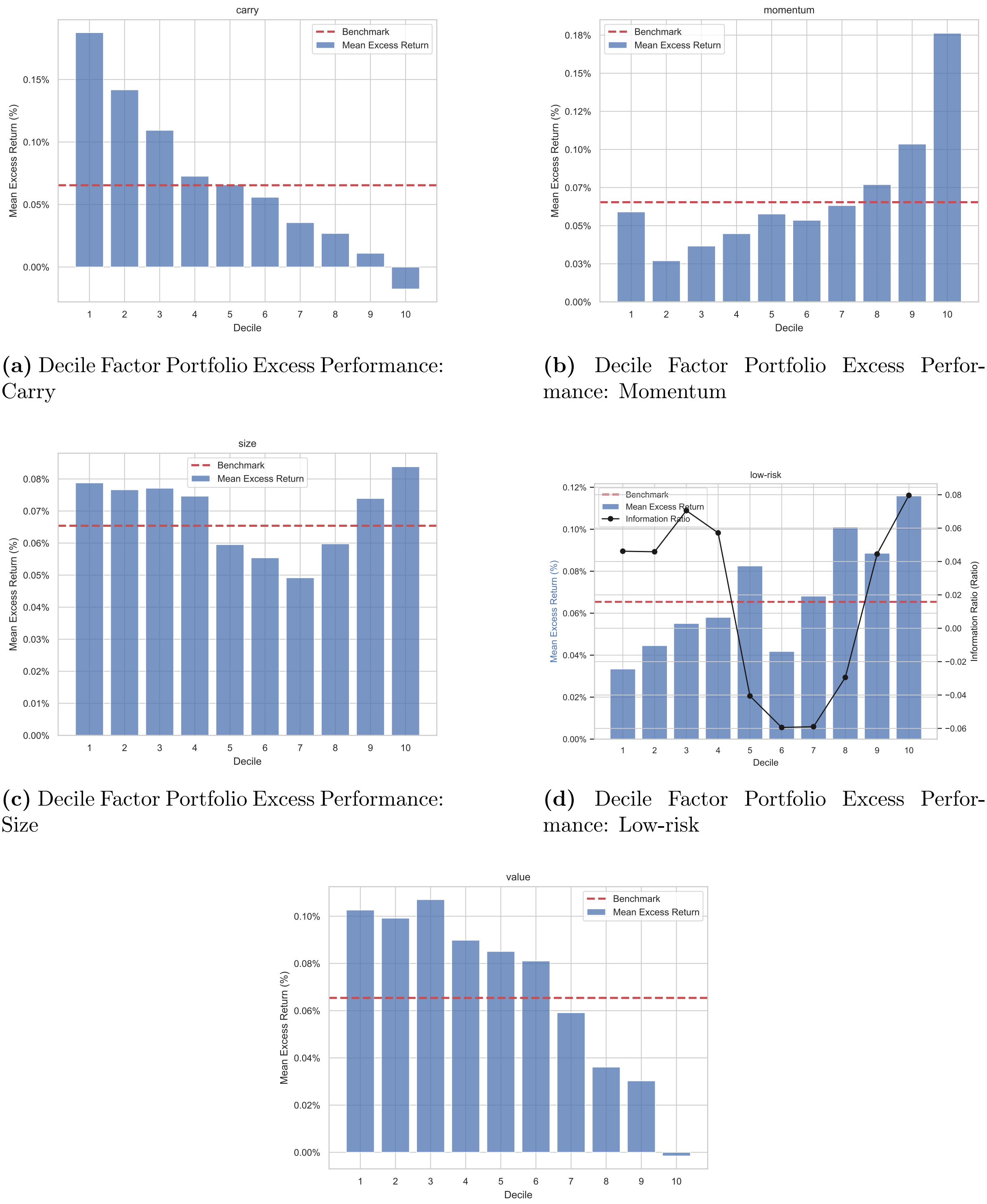
Low-Risk

The results of the decile analysis for the low-risk factor are displayed in Figure 6.1d. Here, we added the IR (Section 2.2.3) for each decile portfolio as a black line to visualize the risk measure in this context. The IR should be higher for the low deciles and lower for the higher deciles. Indeed, the IR is high for the lower deciles, but also for the higher deciles. The high IR for low deciles can be explained by the low standard deviation of the excess return over time. The high IR for the higher deciles can be explained by their high returns with respect to the standard deviation.

When observing the mean excess return, it can be seen that the first decile underperforms the benchmark significantly. It seems that the low-risk bonds (decile 1), do not bring higher excess returns, but are less risky. The high returns of the 10^{th} decile can be explained through the fact that riskier bonds generate higher excess returns.

Value

The results of the decile analysis for the value factor are displayed in Figure 6.1e. The results show that the the first decile portfolio outperforms the last decile portfolio. It suggests that there is a value factor premium in our the European credit market.



(a) Decile Factor Portfolio Excess Performance: Carry

(b) Decile Factor Portfolio Excess Performance: Momentum

(c) Decile Factor Portfolio Excess Performance: Size

(d) Decile Factor Portfolio Excess Performance: Low-risk

(e) Decile Factor Portfolio Excess Performance: Value

Figure 6.1: Decile Factor Portfolio Excess Performance for Various Factors

6.1.2 Portfolio performance

In Figure 6.2 the cumulative excess returns are plotted. From this figure it is clear that the carry portfolio has the highest cumulative excess return. Table 6.1 shows the portfolio metrics, as a monthly mean, of the five factor portfolios.

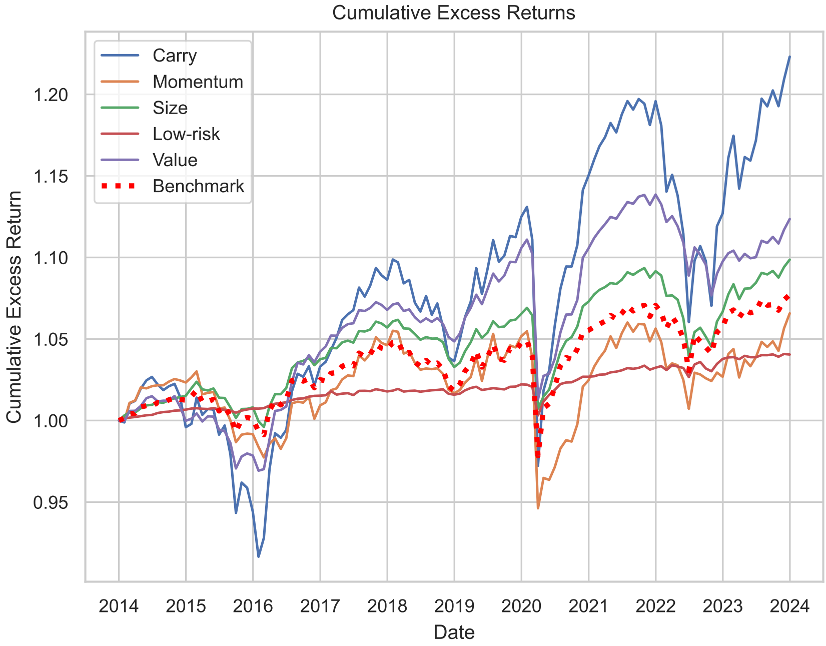


Figure 6.2: Cumulative Total Returns of Factor Portfolios

Table 6.1: Traditional Factor Portfolio Performance

Method	Return[1] (%)	Duration	Z-spread	TTM	Rating	IR
Carry	0.19	6.89	172.67	8.03	3.89	0.10
Momentum	0.06	6.65	90.92	7.62	3.60	-0.01
Size	0.08	4.44	98.12	4.85	3.62	0.05
Low-risk	0.03	1.71	26.97	1.75	2.78	-0.05
Value	0.10	4.25	112.51	4.66	3.18	0.09
benchmark	0.07	5.06	73.84	5.58	3.38	nan

Recognized by the positive IR and high excess return, the factor portfolios that outperform the benchmark are carry, size, and value. The negative IR of the momentum and low-risk portfolios implies underperformance compared to the benchmark. However, the risk metrics of all the portfolios differ from the benchmark. Depending on the risk-return profile or appetite, different portfolios may be of interest.

For example, let us evaluate the risk characteristics of the factor portfolio with the highest return: the carry portfolio. Bonds with high carry score are often riskier such that they generate higher returns. Indeed, the carry portfolio incorporates bonds with a TTM over 8 years, whereas the TTM of the benchmark is 5.58 years. The numerical rating the carry portfolio is 3.89, which is significantly higher than the benchmark rating of 3.38, indicating that the portfolio consists almost entirely of BBB-rated bonds, reflecting lower creditworthiness. Additionally, the duration of the carry portfolio is higher, implying that this portfolio is more sensitive to price changes and therefore riskier.

[1] Monthly average excess return over LIBOR

It is interesting to note that several risk metrics of the value portfolio are lower than those of the benchmark. This indicates that the value portfolio is less risky than the benchmark while still delivering a higher excess return.

6.2 Causal Analysis

In this section, we present the results from the causal discovery on factors and returns using FCI and GES. As described in Section 5.1, the causal analysis is performed on the factor scores and the month-ahead returns over the time periods $T_1 \cup T_2$.

Causal factors and direct causal factors, as defined in Definitions 4.1 and 4.2, are selected via Algorithm 5 from the obtained causal graphs. We will present the causal factors as well as the direct causal factors for both FCI and GES. All causal graphs, from which the causal factors are derived, are included in Appendix B.

There are certain years where the results of the causal analysis cannot be used due to a lack of sufficient causal factors. When no directed path exists from any factor to the return variable, there are no causal factors. Years without causal factors are highlighted in **bold text**. In such cases, the causal graph is omitted, and all five factors are included as features in the machine learning methods.

6.2.1 Fast Causal Inference (FCI)

The causal factors and direct causal factors, obtained via FCI, are stated in Table 6.2.

Table 6.2: (Direct) Causal Factors obtained from FCI

Period	Causal Factors
2009-2013	(siz, val, lr, mom, car)
2009-2014	(car)
2009-2015	(car)
2009-2016	(siz, val, lr, mom, car)
2009-2017	(siz, val, lr, mom, car)
2009-2018	(siz, val, lr, car)
2009-2019	(lr)
2009-2020	(siz, val, lr, mom, car)
2009-2021	(siz, val, lr, mom, car)
2009-2022	(siz, val, lr, mom, car)

Period	Direct Causal Factors
2009-2013	(siz, val, lr, mom, car)
2009-2014	(car)
2009-2015	(car)
2009-2016	(siz, val, lr, mom, car)
2009-2017	(siz, val, lr, mom, car)
2009-2018	(lr, car)
2009-2019	(lr)
2009-2020	(siz, val, lr, mom, car)
2009-2021	(siz, val, lr, mom, car)
2009-2022	(siz, val, lr, mom, car)

There are six periods where FCI did not find sufficient causal factors. This could suggest that the causal graph was inaccurate or that other latent factors are influencing the month-ahead returns. Since only four periods identified (direct) causal factors, we will not use FCI in CFI.

6.2.2 Greedy Equivalence Search (GES)

The causal factors and direct causal factors, obtained via GES, are stated in Table 6.3.

We observe that for all time periods, GES was able to obtain causal factors. Therefore, we will use the causal factors and direct causal factors identified via GES for CFI, rather than those obtained through FCI. From Table 6.3 we can see that there is always a minimum of two (direct) causal factors: momentum and carry. The size factor is the most frequently omitted causal factor. Regarding the direct causal factors, we find that momentum and carry are present every period. Additionally, for the periods (2009-2018) and (2009-2019), the low-risk factor is also included as a direct causal factor.

Table 6.3: (Direct) Causal Factors obtained from GES

Period	Causal Factors
2009-2013	(siz, val, lr, mom, car)
2009-2014	(val, lr, mom, car)
2009-2015	(siz, val, lr, mom, car)
2009-2016	(siz, val, lr, mom, car)
2009-2017	(siz, val, lr, mom, car)
2009-2018	(val, lr, mom, car)
2009-2019	(val, lr, mom, car)
2009-2020	(val, lr, mom, car)
2009-2021	(siz, val, lr, mom, car)
2009-2022	(mom, car)

Period	Direct Causal Factors
2009-2013	(mom, car)
2009-2014	(mom, car)
2009-2015	(mom, car)
2009-2016	(mom, car)
2009-2017	(mom, car)
2009-2018	(lr, mom, car)
2009-2019	(lr, mom, car)
2009-2020	(mom, car)
2009-2021	(mom, car)
2009-2022	(mom, car)

6.3 Empirical Asset Pricing via Machine Learning

In this section we will present the results of the prediction of month-ahead excess returns of bonds via machine learning methods. For the methodology we refer the reader to Section 5.2. We will compare the performance of the three different machine learning methods OLS, XGB, and NN. For each of these machine learning methods, the month-ahead returns are predicted with three sets of features:

1. **Regular:** Includes all available factors scores for each year as features

2. **GES:** Includes causal factors for each year as features obtained with GES

3. **GES-Direct:** Includes direct causal factors for each year as features obtained with GES

Note that, in total, we will consider 9 machine learning models instead of the initial 15 models. This is due to insufficient causal factors obtained with FCI (see Section 6.2). The results are stated in Table 6.4.

Table 6.4: Performance of machine learning methods on the prediction of month-ahead bond returns

Model	Factors	R^2_{OOS}	MSE
OLS	Regular	-0.0033	2.0323E-04
OLS	GES	-0.0034	2.0324E-04
OLS	D-GES	-0.0035	2.0326E-04
XGB	Regular	-0.0036	2.0330E-04
XGB	GES	-0.0035	2.0326E-04
XGB	D-GES	-0.0035	2.0327E-04
NN	Regular	-0.0017	2.0290E-04
NN	GES	0.0010	2.0235E-04
NN	D-GES	-0.0003	2.0263E-04

We can see that the GES NN model has the highest R^2_{OOS} and the lowest MSE compared to all other methods. This method is closely followed by D-GES NN, which has the second-highest R^2_{OOS} and lowest MSE. Moreover, all NN models outperform the other machine learning methods on both metrics. In the case of neural networks, GES and D-GES result in a higher R^2_{OOS} and lower MSE than regular neural networks. For OLS and XGB, the R^2_{OOS} and MSE are similar across all three types of factors: regular, GES, and D-GES.

Also, we observe that for all methods, the R^2_{OOS} is negative except for GES NN. There are several potential reasons for a negative R^2_{OOS} . One possibility is overfitting the training data, perhaps due to a lack of regularization techniques. Another possibility for a negative R^2_{OOS} is the complexity of predicting the month-ahead returns using only five features. Precisely predicting future returns, which are often exposed to noise, is difficult. Furthermore, recall from Definition 4.4 that the R^2_{OOS} metric compares the predictions of the model with a naive zero-value prediction. The most likely reason for a negative R^2_{OOS} in our numerical results is that a naive zero forecast may be a 'safe' prediction. Since excess returns tend to be small and centered around zero, a zero prediction can be difficult to outperform.

While a negative R^2_{OOS} is not desirable when focusing solely on predictive power, from a strategy perspective, precise month-ahead return predictions are not always necessary. CFI invests in bonds with the highest predicted future returns. If the machine learning model can distinguish whether a bond's return will be relatively high or low, without needing to predict the exact price, this prediction can still serve as a useful investment signal. It is possible for these predictions to have a negative R^2_{OOS} but still provide valuable information for the strategy.

6.4 Investment Strategy

In this section the results of Causal Factor Investing in the credit market are stated. We have followed the investment strategy of CFI as described Section 5.3. For more information on the portfolio metrics, we refer the reader to Section 2.2.3. The cumulative returns are displayed in Figure 6.3 and additional metrics are given in Table 6.5.

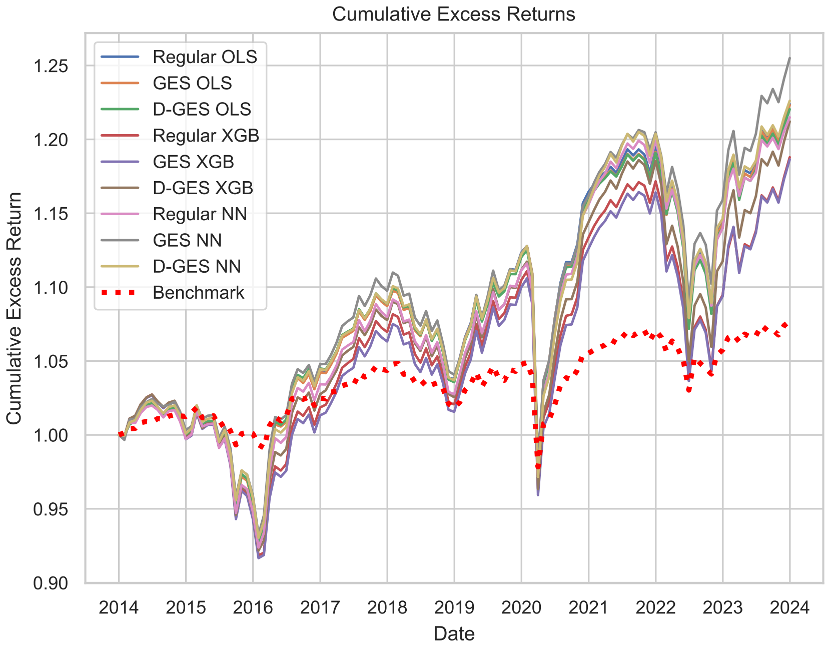


Figure 6.3: Cumulative excess returns machine learning portfolios 2014-2023

Observing the cumulative returns in Figure 6.3, we see that all of the machine learning portfolios exhibit higher excess cumulative returns than the benchmark. However, the benchmark shows greater resistance to fluctuations. During periods of negative economic sentiment, the machine learning portfolios consistently dip below the benchmark.

[1] Monthly average excess return over LIBOR

Table 6.5: Performance of machine learning portfolios

Model	Features	Return[1] (%)	Duration	Z-spread	TTM	Rating	IR
OLS	Regular	0.19	6.90	149.95	7.93	3.70	0.11
OLS	GES	0.19	6.84	152.03	7.87	3.73	0.11
OLS	D-GES	0.18	6.84	153.54	7.85	3.78	0.11
XGB	Regular	0.16	6.86	164.02	7.92	3.87	0.09
XGB	GES	0.16	6.86	164.16	7.93	3.87	0.09
XGB	D-GES	0.18	7.00	165.96	8.14	3.87	0.10
NN	Regular	0.18	6.92	151.83	7.94	3.82	0.11
NN	GES	0.21	6.76	157.94	7.76	3.83	0.13
NN	D-GES	0.19	6.98	158.23	8.07	3.83	0.11
Benchmark	-	0.07	5.06	73.84	5.58	3.38	-

From Table 6.5, we observe that all machine learning portfolios have a higher excess return than the benchmark. However, all portfolios also exhibit higher risk metrics compared to the benchmark. The three NN portfolios and the three OLS portfolios have similar returns, which are higher than those of the three XGB portfolios.

The GES NN portfolio stands out with the highest return of 0.21, and its risk metrics are low compared to all the other machine learning portfolios. Additionally, the GES NN portfolio has the highest IR of 0.13 among both machine learning and traditional portfolios. When comparing the GES NN portfolio with the carry portfolio, which has the highest return of 0.19 among the traditional portfolios, the GES NN portfolio not only has a higher return but also exhibits lower risk metrics

All nine machine learning portfolios have a higher risk profile than the benchmark. However, the GES NN portfolio offers the highest return among the nine machine learning portfolios and, notably, has relatively low risk compared to the other machine learning portfolios.

7 | Conclusion and Discussion

The goal of this thesis is to explore the integration of causal theory and machine learning into factor investing, aiming to enhance both the performance and explainability of the investment method. Factor investing traditionally relies on managing portfolios exposed to quantifiable characteristics, known as factors, which explain differences in returns across an asset class. However, recent criticisms highlight the possible shortcomings of traditional factor investing due to specification errors and the reliance on correlations rather than causality. Additionally, a traditional factor portfolio corresponds to a single factor portfolio, which increases transaction costs if an investor wants to invest in multiple factors. To address these concerns, we developed the novel Causal Factor Investing (CFI) framework, which consists of three main steps. First, it performs causal discovery to identify factors, newly defined as causal factors, that have a causal-effect relationship with asset returns. In the second step, these causal factors are incorporated as features in machine learning models to predict future asset returns. In the third step, these predicted returns serve as a composite signal for the investment strategy.

In Chapter 4, we specified CFI generically so that it can be used for different asset classes. As direct interventions are not feasible, we argued for causal discovery on factor scores and returns resulting in a causal graph. The notion of (direct) causal factor was defined based on possible directed paths in a causal graph. By modeling future returns as an additive prediction error model, machine learning methods can be used to predict future returns, serving as an investment signal, with (direct) causal factors as features.

The application of CFI in this thesis, detailed in Chapter 5, focuses on the European corporate bond market, where CFI was tested against traditional factor investing methods. The credit factors considered are size, low-risk, value, momentum, and carry. We used i.i.d causal discovery algorithms: Fast Causal Inference (FCI) and Greedy Equivalence Search (GES). FCI is a constraint-based method accounting for hidden confounders, while GES is a score-based method assuming causal sufficiency. We employed three machine learning methods: Ordinary Least Squares (OLS), Extreme Gradient Boosting (XGB), and Neural Networks (NN) to predict future bond returns using an expanding window.

In Chapter 6, we backtested CFI in the corporate bond market with an expanding window. This led to slightly better results than a moving window in terms of predicting future returns and portfolio performance. FCI produced insufficient causal graphs and was not further used, but GES provided sufficient causal graphs. For predictive accuracy of future returns, neural networks had the highest performance, with the highest R^2_{OOS} and lowest Mean Squared Error (MSE). Backtesting the CFI showed that all machine learning portfolios significantly outperformed the benchmark but with higher risk metrics. NN and OLS portfolios had the highest returns, followed by XGB portfolios. The best-performing portfolio, GES NN, had a 0.21 mean excess return, offering higher returns and lower risk metrics than the carry portfolio, the most competitive traditional portfolio. We observed that the use of (direct) causal factors improve portfolio profitability compared to regular factors with NN portfolios, but not with XGB or OLS. Nonetheless, (direct) causal factors provide a deeper understanding of factor-return relationships, addressing the black-box nature of machine learning.

Despite the promising results, this research has several limitations. In financial markets, there is a high likelihood of hidden confounders, which led us to use FCI. However, FCI did not always produce valid causal graphs. We resorted to GES, which assumes causal sufficiency and may therefore be prone to errors. Nevertheless, due to the greedy optimization nature of GES, which uses likelihood functions to score graphs relative to the data, we still believe GES provides valuable insights. In general, validating causal graphs obtained via causal discovery is challenging, as there is no true graph available. Additionally, exploring time-series causal discovery (approach 6), which might involve creating models for each asset individually, could lead to more accurate causal graphs and improved performance. Another limitation lies in the computational demands of machine learning and causal discovery methods; both FCI and GES have exponential computational complexity as the dimensionality increases. Also, transaction costs of rebalancing portfolios were not included in our analysis, which is another limitation.

The findings of this thesis suggest several options for future research. One potential direction is to extend the CFI framework to other asset classes, such as equities, to evaluate its adaptability and performance in different market environments. In such contexts, it could be easier to experiment with time-series causal discovery methods and higher-dimensional data. This can potentially initiate a comparative study of various causal discovery algorithms where also the reduction of computational time and resources is explored. Moreover, integrating additional factors, such as macroeconomic and environmental factors, or more asset characteristics might also enhance the accuracy of future return predictions.

From an investor's perspective, the machine learning portfolios indicate high returns but also a higher risk profile. Financial institutions often face risk and transaction constraints, making it interesting to see the maximization of returns and minimization of risk constraints as a optimization problem. Using optimization approaches on predicted future returns and risk constraints at time t, investors can construct optimized portfolios.

In conclusion, this thesis demonstrates the feasibility and potential benefits of integrating causal theory and machine learning into factor investing through the development of CFI. Trough the use of causal discovery and machine learning, CFI pioneers in addressing causal relations and model misspecifications in existing factor investing strategies. As the financial industry continues to evolve, CFI presents promising path forward for enhancing both performance and explainability in quantitative investing.

References

[1] Juan Alonso-Barba, Luis De La Ossa, Jose Gámez, and Jose Puerta. Scaling up the greedy equivalence search algorithm by constraining the search space of equivalence classes. In *Proceedings of the 11th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, ECSQARU'11, page 194–205, Berlin, Heidelberg, 2011. Springer-Verlag.

[2] Turan Bali, Amit Goyal, Dashan Huang, Fuwei Jiang, and Quan Wen. Predicting corporate bond returns: Merton meets machine learning. Research Paper 3686164, 2020.

[3] Mohammed Ben Slimane, Marielle De Jong, Jean-Marie Dumas, Hamza Fredj, Takaya Sekine, and Michael Srb. Traditional and alternative factors in investment grade corporate bond investing, 2019.

[4] Leo Breiman, Jerome Friedman, Richard Olshen, and Charles Stone. *Classification and Regression Trees.* Chapman and Hall/CRC, 1984.

[5] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016.

[6] Amina Cherief, Mohamed Ben Slimane, Jean-Marie Dumas, and Hamza Fredj. Credit factor investing with machine learning techniques, 2022.

[7] David Maxwell Chickering. Optimal structure identification with greedy search. *J. Mach. Learn. Res.*, 3:507–554, 2002.

[8] John Cochrane. Presidential address: Discount rates. *The Journal of Finance*, 66(4):1047–1108, 2011.

[9] Guillaume Coqueret and Tony Guida. *Machine Learning for Factor Investing: Python Version.* June 2023.

[10] Maria Correia, Scott Anthony Richardson, and Ayse Irem Tuna. Value investing in credit markets. *Review of Accounting Studies*, 17(3), 2012.

[11] Saul Doctor. Fact or fiction: Investigating factors in corporate credit, 2019.

[12] Gabriele D'Acunto, Paolo Bajardi, Francesco Bonchi, and Gianmarco De Francisci Morales. The evolving causal structure of equity risk factors. In *Proceedings of the Second ACM International Conference on AI in Finance*, ICAIF'21. ACM, November 2021.

[13] Eugene Fama and Kenneth R. French. The cross-section of expected stock returns. *The Journal of Finance*, 47(2):427–465, 1992.

[14] Eugene Fama and Kenneth R. French. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33(1):3–56, 1993.

[15] Eugene Fama and Kenneth R. French. A five-factor asset pricing model. *Journal of Financial Economics*, 116(1):1–22, 2015.

[16] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA, April 2011. PMLR.

[17] Lingyi Gu, Ellen Zhang, Andrew Heinz, Jingxuan Liu, Tianyue Yao, Mohamed AlRemeithi, and Zelei Luo. Re-examination of fama-french factor investing with causal inference method, 2023.

[18] Shihao Gu, Bryan Kelly, and Dacheng Xiu. Empirical Asset Pricing via Machine Learning. *The Review of Financial Studies*, 33(5):2223–2273, February 2020.

[19] Uzma Hasan, Emam Hossain, and Md Osman Gani. A survey on causal discovery methods for i.i.d. and time series data, 2024.

[20] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning.* Springer New York, 2009.

[21] Harald Henke, Hendrik Kaufmann, Philip Messow, and Jieyan Fang-Klingler. Factor investing in credit. *The Journal of Index Investing*, 11:jii.2020.1.085, April 2020.

[22] Patrick Houweling and Jeroen van Zundert. Factor investing in the corporate bond market. *Financial Analysts Journal*, 73(2), 2017.

[23] Clint Howard, Harald Lohre, and Sebastiaan Mudde. Causal network representations in factor investing. 2023.

[24] Antti Ilmanen. Expected returns: An investor's guide to harvesting market rewards. *Expected Returns: An Investor's Guide to Harvesting Market Rewards*, May 2011.

[25] Ronen Israel, Diogo Palhares, and Scott A Richardson. Common factors in corporate bond returns. *Forthcoming in the Journal of Investment Management*, 2017.

[26] Gergana Jostova, Stanislava (Stas) Nikolova, Alexander Philipov, and Christof Stahel. Momentum in corporate bond returns. *The Review of Financial Studies*, 26(7):1649–1693, 2013.

[27] Markus Kalisch and Peter Bühlmann. Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *J. Mach. Learn. Res.*, 8:613–636, May 2007.

[28] Steffen Lauritzen. *Graphical Models.* Oxford University Press, 1996.

[29] Marcos López de Prado and Vincent Zoonekynd. Why has factor investing failed? the role of specification errors. *SSRN*, January 2024.

[30] Seyed Mohammad Mansouri and Dalibor Eterovic. Machine learning and the cross-section of emerging market corporate bond returns. 2023.

[31] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, March 1952.

[32] Franz H Messerli. Chocolate Consumption, Cognitive Function, and Nobel Laureates. *New England Journal of Medicine*, 367(16):1562–1564, 2012.

[33] Raha Moraffah, Paras Sheth, Mansooreh Karami, Anchit Bhattacharya, Qianru Wang, Anique Tahir, Adrienne Raglin, and Huan Liu. Causal inference for time series analysis: problems, methods and evaluation. *Knowledge and Information Systems*, 63(12):3041–3085, December 2021.

[34] Ana Nogueira, Andrea Pugnana, Salvatore Ruggieri, Dino Pedreschi, and João Gama. Methods and tools for causal discovery and causal inference. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 12, January 2022.

[35] Cornelis Oosterlee and Lech Grzelak. *Mathematical Modeling and Computation in Finance: With Exercises and Python and MATLAB Computer Codes.* World Scientific, December 2019.

[36] Roxana Pamfil, Nisara Sriwattanaworachai, Shaan Desai, Philip Pilgerstorfer, Konstantinos Georgatzis, Paul Beaumont, and Bryon Aragam. Dynotears: Structure learning from time-series data. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1595–1605. PMLR, August 2020.

[37] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, 1988.

[38] Jonas Peters. Causality (lecture notes), September 2015.

[39] Thomas Richardson and Peter Spirtes. Ancestral graph Markov models. *The Annals of Statistics*, 30(4):962 – 1030, 2002.

[40] Russel. The value effect. *FTSE Russel Internal Research Paper*, 2018.

[41] Agathe Sadeghi, Achintya Gopal, and Mohammad Fesanghary. Causal discovery in financial markets: A framework for nonstationary time-series data. *SSRN Electronic Journal*, December 2023.

[42] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search.* MIT press, 2nd edition, 2000.

[43] Peter Spirtes and Thomas S. Richardson. A polynomial time algorithm for determining dag equivalence in the presence of latent variables and selection bias. In *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*, volume R1 of *Proceedings of Machine Learning Research*, pages 489–500. PMLR, January 1997.

[44] Ruijie Tang. Trading with time series causal discovery: An empirical study, 2024.

[45] Thomas Verma and Judea Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, UAI '90, page 255–270, USA, 1990. Elsevier Science Inc.

[46] Fadi Zaher. *Fixed Income Factor Investing*, pages 173–204. Springer International Publishing, Cham, 2019.

[47] Alessio Zanga and Fabio Stella. A survey on causal discovery: Theory and practice, 2023.

[48] Jiji Zhang. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16):1873–1896, 2008.

[49] Fujin Zhu. *On Causal Discovery and Inference from Observational Data.* Phd thesis, University of Technology Sydney, Centre for Artificial Intelligence (CAI), School of Computer Science, Faculty of Engineering and Information Technology (FEIT), August 2019.

[50] Shunwei Zhu, Chunyang Zhou, Hailong Liu, and Yangyang Ren. Commodity factor investing via machine learning. *Pacific-Basin Finance Journal*, 83(C), 2024.

A | Mathematical Concepts of Machine Learning Methods

This appendix covers the basic mathematical theory of three supervised machine learning methods used in thesis. Let us introduce the generic notation used in this chapter. Let us assume we have a dataset $(\mathbf{X}, \mathbf{y})$ of N data points $(\mathbf{x}_i, y_i)$ where $\mathbf{x_i} \in \mathbb{R}^K$ and $y_i \in \mathbb{R}$ with $i \in \{1, \dots, N\}$. In matrix notation, $\mathbf{X}$ is an $N \times K$ matrix and $\mathbf{y}$ is an N-dimensional vector. Here $\mathbf{y}$ is called the *target variable* which we try to predict with the *features* $\mathbf{X}$. The general equation in supervised learning is

$$\mathbf{y} = g(\mathbf{X}) + \epsilon. \tag{A.1}$$

The goal is to model the function g such that error ϵ is minimized while maintain high predictive out-of-sample accuracy.

One of the most basic models, ordinary least squares (OLS), is covered in Section A.1. The two subsequent sections elaborate on more complex machine learning methods. The mathematical concepts of tree-based methods and neural networks are covered in Section A.2 and Section A.3, respectively.

A.1 Ordinary Least Squares

We initiate this chapter with the most simple model, a simple linear model, which is estimated using ordinary least squares (OLS). While we anticipate that this model may not perform optimally in our context it serves as a baseline for comparison with more complex methods.

The simple linear model assumes that the target variable y can be decomposed as a linear combination of features $\mathbf{x}$. In matrix notation we have

$$\mathbf{y} = \mathbf{X}\beta + \epsilon \tag{A.2}$$

where ϵ is the error. The optimal choice of β is the one that minimizes the error term. The OLS method seeks to minimize the sum of squared residuals, which is given by $L := \sum_{i=1}^{N} \epsilon_i^2$. To find the optimal β, OLS intuitively differentiates L with respect to β and find the minimum by setting the gradient equal to 0. The analytical solution is

$$\beta^* = \mathbf{X}'\mathbf{X}^{-1}\mathbf{X}'\mathbf{y} \tag{A.3}$$

and does not require complex numerical calculations.

A.2 Tree-based Methods

Simple trees

This section contains the fundamental concepts related to simple trees. As mentioned in the introduction of this chapter, trees are structured to divide observations into groups with similar behaviors. The target variable in tree models can assume either discrete values, in the case of *classification* trees, or continuous (numerical) values, in the case of *regression* trees. This thesis focuses exclusively on regression trees. To provide a clearer understanding, an illustrative example is displayed in Figure A.1 and further explained in the following paragraph.

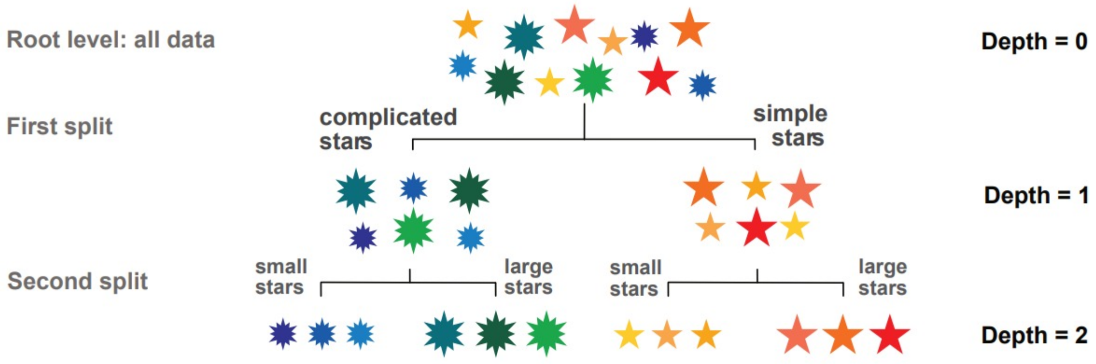


Figure A.1: Visual example of a simple tree. Adopted from Coqueret et al. [9].

In this example we have 12 starts with, for each star, three variables: size, color and complexity. Our dependent variable is the color and the two features are size and complexity. In each of the three layers, which we call depth d, a split is made to obtain the most homogeneous subsets in the final layer of the tree. Here, the first split is made based on complexity, and the second split is made according to size. If one would make the split the other way around, it would have created groups of mixed colors, which is not the preferred result. This visual image can be linked with factor investing by letting the color represent the return of an asset and the features represent bond factor scores. By finding the optimal splitting order and splitting point of the factor scores we can split our dataset into subsets with different month-ahead returns.

We follow Coqueret et al. [9] which is based on the standard literature on simple trees of Breiman et al. [4] and Hastie et al. [20]. Simple *regression* trees cut the feature space X into rectangles or regions R_m and then assign an output for the target variable, for instance a constant c_m, in each of them.

Let us start with an example regression problem where we have target variable Y and two features X_1 and X_2. The feature space is split sequentially into regions R_m with $m \in \{1, \ldots, M\}$ and the target variable is modelled by a certain output in each region; in this example we just denote this as Exit m. This continues until it some stopping rule is triggered. In this case, the first splitting is done at $X_1 > 0.15$ and the seconds split divides the region $X_1 > 0.15$ into two regions based on $X_2 > 0.5$. In Figure A.2 one can see the the regions on the left panel, and the decision tree on the right panel.

The fitted regression model $\hat{f}$ predicts Y with an output variable (Exit m), in this case a constant c_m, in each region R_m. Our fitted (trained) regression model can be expressed as

$$\hat{f}(X) = \sum_{m=1}^{M} c_m \mathbb{1}\{X \in R_m\} \tag{A.4}$$

It remains to determine the optimal splitting points. We try to find the splitting points that minimize the total variation within the two corresponding clusters after the split. We call these clusters *child* clusters and they do not have to be the same size. This requires two steps:

1. Find the optimal splitting point for each feature x_i^k with $k \in \{1, \ldots, K\}$ such that the clusters are homogeneous

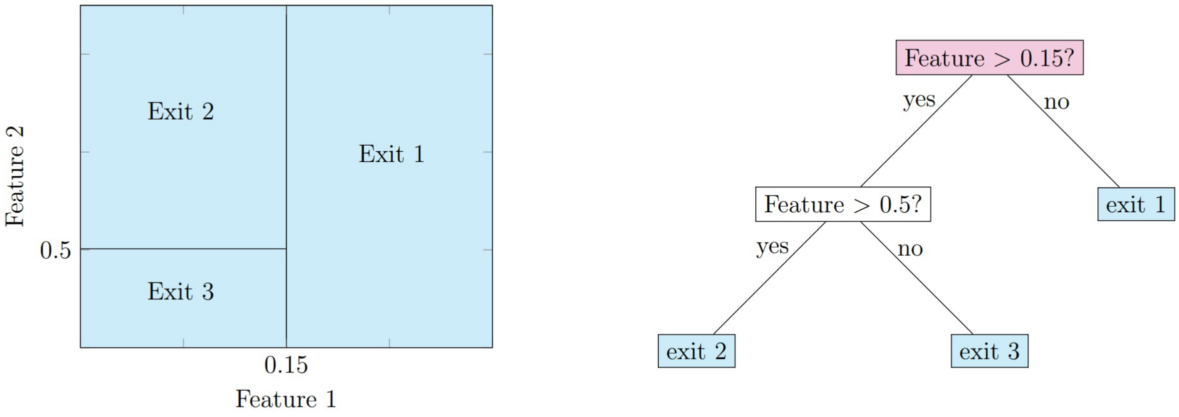


Figure A.2: Regression tree example. Adopted from Cherief et al. [6]

2. Select the feature that causes the highest homogeneity in the child clusters with respect to the target variable

One question that immediately comes to mind is how do we measure homogeneity in the clusters. As we aim to have clusters similar y_i inside, we seek to minimize the *dispersion* inside the two clusters. Variance cannot be used since the summation of two variance values does not take into account the relative size of the clusters. Therefore, often *total variation* is used, where the variance is multiplied by the number of elements in the clusters (see Coqueret et al. [9]).

Let us start with a sample $(\mathbf{x_i}, y_i)$ of the dataset of size I. In the first step, we seek the optimal splitting point for each feature x_i^k for every k by reducing the total variation in the two clusters. That is, we solve

$$\underset{c^k}{\operatorname{argmin}} \underbrace{\sum_{x_i^k < c^k} \left(y_i - m_I^{k,-}\left(c^k\right)\right)^2}_{\text{Total dispersion of first cluster}} + \underbrace{\sum_{x_i^k > c^k} \left(y_i - m_I^{k,+}\left(c^k\right)\right)^2}_{\text{Total dispersion of second cluster}} . \tag{A.5}$$

Here $m_I^{k,-}$ and $m_I^{k,+}$ are the average values of Y conditional on X^k being smaller or than c^k respectively. If we let $\#\{\cdot\}$ be the cardinal function that counts the number of instances based on inserted argument, $m_I^{k,-}$ and $m_I^{k,+}$ can be expressed as

$$m_I^{k,-}\left(c^k\right) = \frac{1}{\#\{i, x_i^k < c^k\}} \sum_{\{x_i^k < c^k\}} y_i \quad \text{and} \tag{A.6}$$

$$m_I^{k,+}\left(c^k\right) = \frac{1}{\#\{i, x_i^k > c^k\}} \sum_{\{x_i^k > c^k\}} y_i. \tag{A.7}$$

The optimal splitting point for feature k is the solution of equation (A.5) and is denoted as c^{k*}. This optimal splitting point is found for all $k \in \{1, \ldots, K\}$. In step 2 we select the optimal splitting point that has the lowest total dispersion of the two child clusters. We split the sample by the splitting point chosen in step 2 to obtain two new clusters of data points and we repeat the process again on these clusters.

Pruning

The process described in the previous section can continue until the tree is fully grown. In such cases, every instance belongs to a separate leaf, or some clusters cannot be split further. Fully grown trees, featuring numerous variables, almost perfectly fit the trained data. However, this scenario is not desirable as it leads to overfitting. The most valuable splits are typically those made at the beginning of the algorithm, as they capture the most general information. Therefore, it is prudent to limit the size of the tree through a process known as *pruning*, by implementing a stopping criterion.

One can set certain *hyperparameters* to characterise the stopping criterion of your regression tree. Hyperparameters are parameters used for tuning the model; they allow to control the learning process. The most common ways of pruning a tree are listed below:

- Fix the depth d of the tree.
- Impose a minimum gain for each split. The split must result in a significant reduction in dispersion; otherwise, the split will not be executed.
- Impose a minimum size of a cluster in each *leaf* node. A leaf node is a terminal node of the tree.
- Impose a minimum size of of a cluster before each split.

Random forest

Random forests are an ensemble learning method leveraging the power of multiple decision trees to improve predictive accuracy and control over-fitting. The underlying principle of random forests is to build multiple decision trees at training time and use the combined result as output.

Random forests makes use of different decision trees, which lead to different prediction models, in two ways:

1. One can create each tree from a different random sample of the data. This sampling is typically performed with replacement, known as *bootstrap sampling*. Each tree in a random forest learns from a unique subset of data points, and this randomness helps to make the model more robust than a single decision tree.
2. Random subsets of features can be selected during each split in the learning process. This is mostly used when a large number of features are available.

Therefore, one can train multiple different trees on a large dataset. The fitted random forest, denoted as $\hat{f}_{RF}$ can be defined as weighted combination, usually equal weighted, of the different trained decision trees as in equation (A.4). If we use L decision trees $\hat{f}_i$ we can express the fitted random forest as

$$\hat{f}_{RF}(X) = \frac{1}{L}\sum_{i=1}^{L}\hat{f}_i(X). \tag{A.8}$$

Boosted trees

Similarly as random forests (see Section A.2), boosting makes use of multiple trees. However, boosting is slightly more refined. Rather than relying on averaging multiple trees, boosting is an iterative process where the model is improved whenever a new tree is added in each iteration. In this thesis we focus on *XGBoost*, which is a highly effective machine learning method that is widely used across a variety of domains. Again, we follow the notation of Coqueret et al. [9] which is based on the original algorithm XGBoost.

A.3 Neural Networks

Neural networks, named after the neuronal structures found in the brain, are powerful tools in the field of machine learning. In this thesis we focus on the traditional feed forward neural network, which is simple. This section outlines the fundamental mathematical concepts underlying neural networks from a high-level perspective starting from the basic fundamentals of a neural network. The theory is in line with the book of Coqueret et al. [9].

The Perceptron

A *perceptron* is a fundamental building block of neural networks, originally introduced by 1958. The perceptron can be viewed as a single 'neuron' in a neural network, and its mathematical formulation serves as the basis for more complex models. Generally speaking, a perceptron takes inputs, applies a calculation and gives an output which we have visualized in Figure A.3. Mathematically, a perceptron maps an input vector $\mathbf{x}$ to an output y by computing a weighted sum of these features. Then, a bias term b is added and is followed by usage of the activation function. This can be expressed as:

$$y = \phi(\mathbf{w}^\top \mathbf{x} + b), \tag{A.9}$$

where y is the output $\mathbf{w}$ represents the weights associated with each input feature, b is the bias term, and $\phi(\cdot)$ is the activation function. The role of the activation function ϕ is to introduce non-linearity into the model, allowing the perceptron to capture more complex patterns in the data.

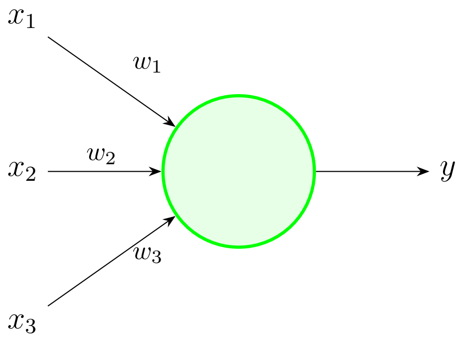


Figure A.3: Visual representation of a perceptron

Activation Functions

Activation functions require to be non-linear and differentiable preferably on the whole domain. Logically, a non-linear activation function is necessary to allow for non-linear relations in the model. Differentiability is required during the training process, i.e. updating the weights, using the stochastic gradient descent which is covered in Section A.3. Similar as Gu et al. [18], this thesis will make use of the *Rectified Linear Unit* (ReLU) activation function:

$$\text{ReLu}(x) = max\{0, x\}. \tag{A.10}$$

It is resistant to a vanishing gradient descent because the derivative is for $x > 0$ is equal to 1 (see Glorot et al. [16]). In addition, is computationally efficient, making it on of the most widely used activation functions in deep learning. Two other common activation functions are:

- **Sigmoid Function:** Defined as $\sigma(z) = \frac{1}{1+e^{-z}}$, this function maps input values to the range $(0, 1)$, making it suitable for binary classification problems.

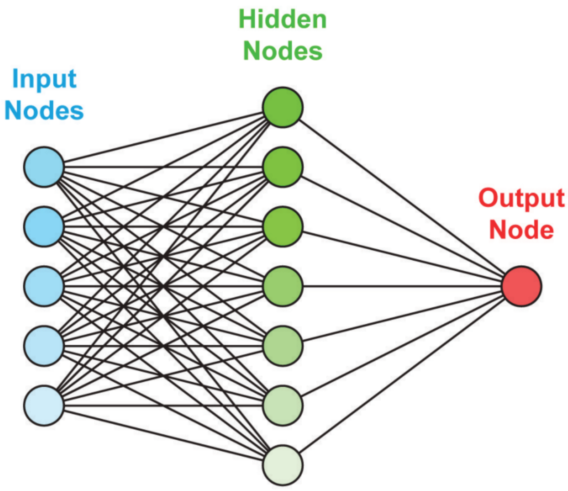


Figure A.4: Visual representation of a multi-layer perceptron. Adopted from Coqueret et al. [9].

- **Hyperbolic Tangent (tanh):** The tanh function, given by $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$, maps inputs to the range $(-1, 1)$. It is a rescaled version of the sigmoid function with zero-centered outputs.

Multi-Layer Networks

When various single perceptrons are chained to eachother, we obtain a network, which allows for greater complexity and has the possiblity to approximate many functions. A multi-layer neural network, also known as a multi-layer perceptron (MLP), is constructed of multiple layers of perceptrons. One single layer can be defined as a group of perceptrons, usually displayed as a vertical stack, that depend on the same previous input. Simple feed forward neural networks are typically categorized into an input layer, one or more hidden layers, and an output layer. Each perceptron in a hidden layer receives input from the previous layer and applies an activation function to generate an output, which is then passed to the next layer. A perceptron can be connected to multiple other perceptrons of a different layer. A visualization of a multi-layer is shown in Figure A.4.

Let $\mathbf{y}^{(l)}$ denote the output of the l-th layer, and $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ represent the weight matrix and bias vector for the l-th layer, respectively. The forward propagation through the network can be described as:

$$\mathbf{y}^{(l+1)} = \phi(\mathbf{W}^{(l)}\mathbf{h}^{(l)} + \mathbf{b}^{(l)}), \tag{A.11}$$

where $\mathbf{h}^{(0)} = \mathbf{x_i}$ represents the input features. The final output of the network is obtained after the last layer's output have been computed. It is important to note that the final output of the neural network, $g(\mathbf{x}, \theta)$, is a composition of simple functions with parameters θ. θ are all the parameters of the neural network.

Multi-layer networks have the capacity to approximate any continuous function, known as the *universal approximation theorem.* The depth and the width, the amount of perceptrons on each layer, influence its performance. Deeper networks are generally more powerful, though they may require more data and computational resources to train.

Gradient Descent

Now that we have discussed how the neural network operates, it is left discuss how to train the neural network. Training a neural network involves optimizing the weights $\mathbf{W}$ and biases $\mathbf{b}$

such that the network's predictions $\hat{\mathbf{y}}$ closely match the target variable $\mathbf{y}$. This optimization is typically framed as a minimization problem, where the objective is to minimize a loss function $\mathcal{L}(\theta)$. Generically, the loss function can be defined as the distance of the target output y and the output of the neural network $g(\mathbf{x}, \theta)$:

$$\mathcal{L}(\theta) = ||y - g(\mathbf{x}, \theta)|| \tag{A.12}$$

Gradient descent is the most common optimization algorithm used to train neural networks. The algorithm iteratively updates the network's parameters in the direction that reduces the loss function. The iterative updating rule for each parameters θ is given by:

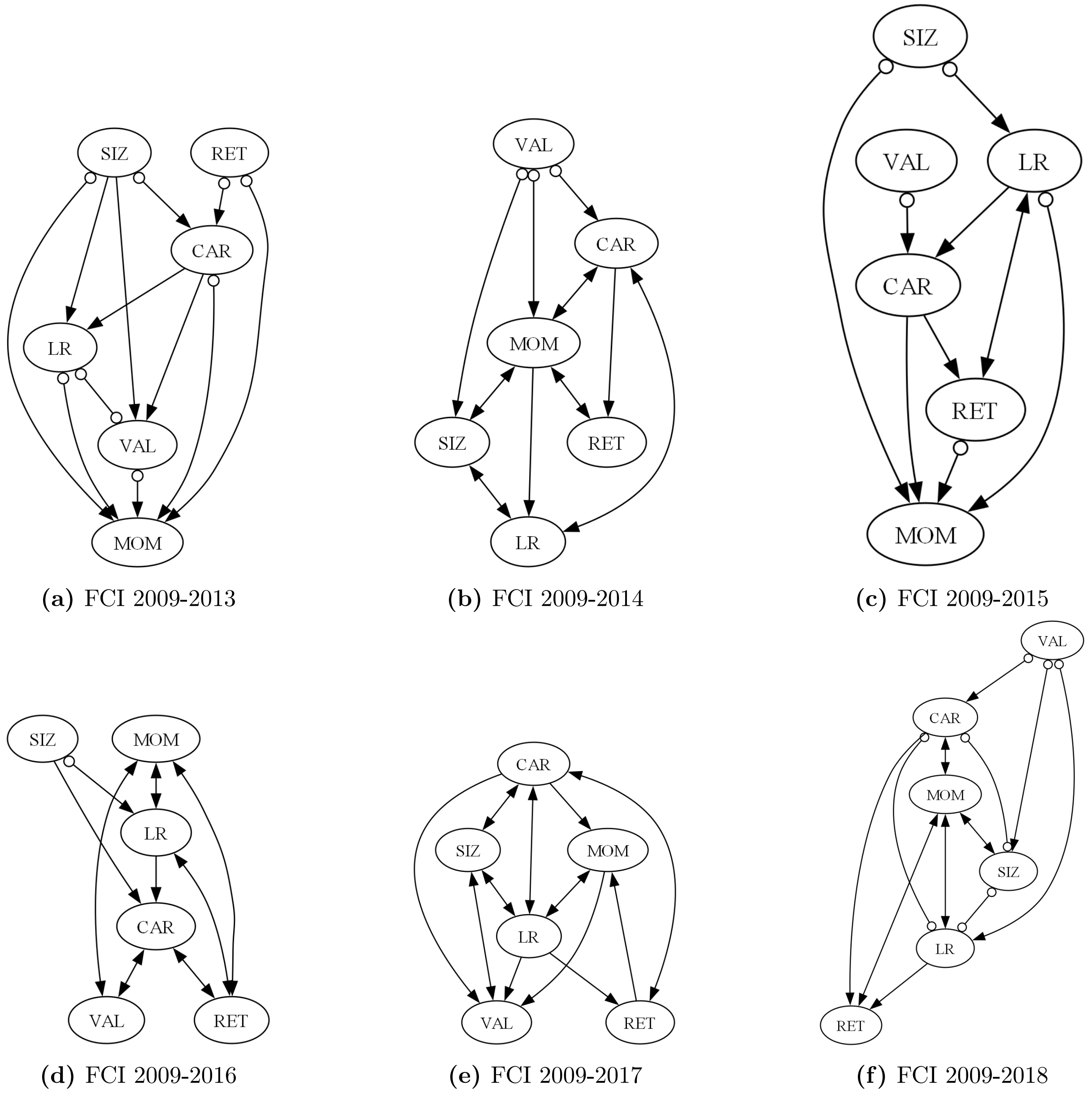
$$\theta \leftarrow \theta - \eta \frac{\partial \mathcal{L}}{\partial \theta}, \tag{A.13}$$

Here, $\eta > 0$ is the learning rate, a hyperparameter controlling the step size of the updates. The gradient $\frac{\partial \mathcal{L}}{\partial \theta}$ is computed via *backpropagation* by applying the chain rule through the layers of the network.

B | Graphs

B.1 FCI

Expanding Window



(a) FCI 2009-2013

(b) FCI 2009-2014

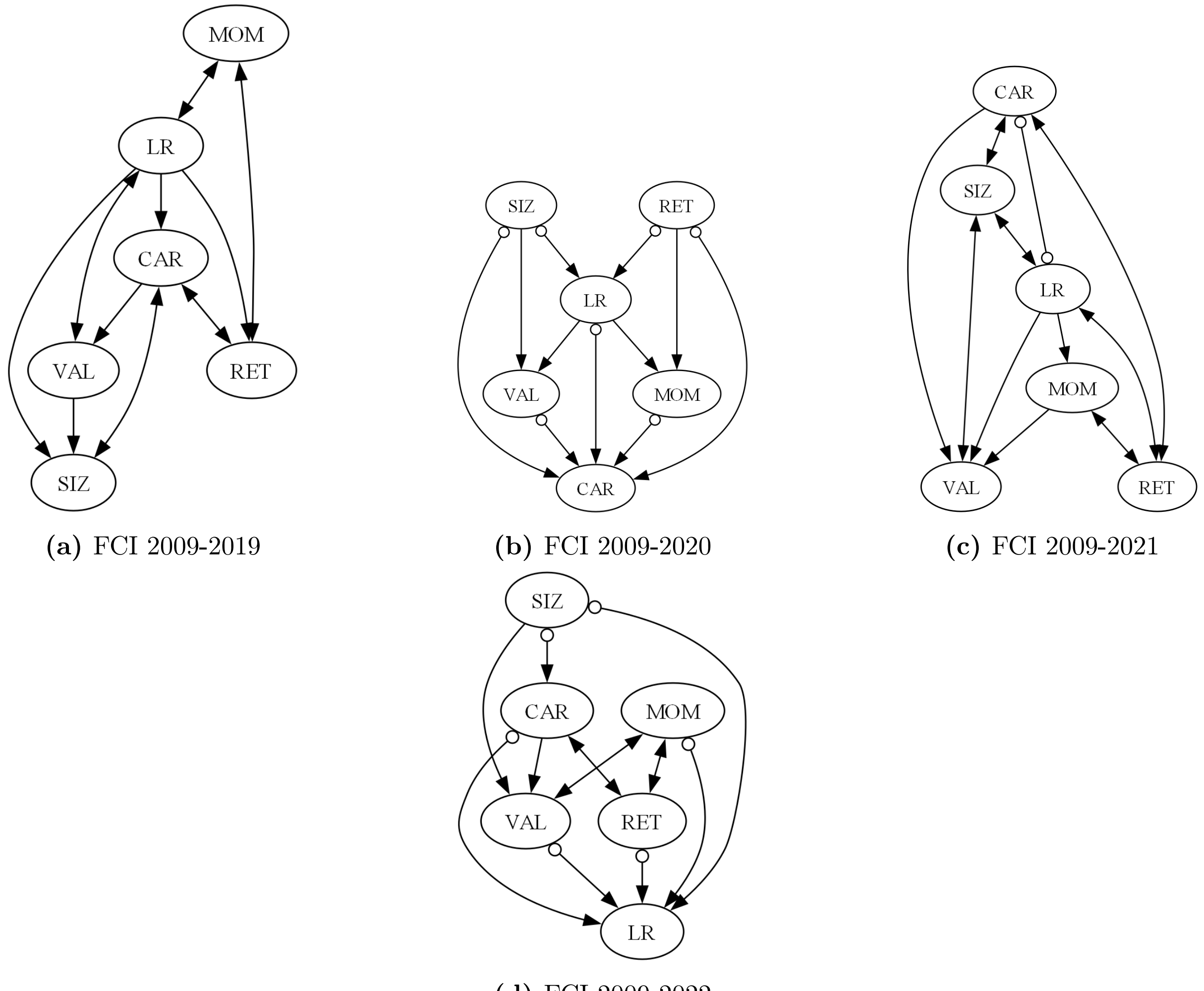
(c) FCI 2009-2015

(d) FCI 2009-2016

(e) FCI 2009-2017

(f) FCI 2009-2018

Figure B.1: Causal graphs obtained via FCI expanding window (1/2)



(a) FCI 2009-2019

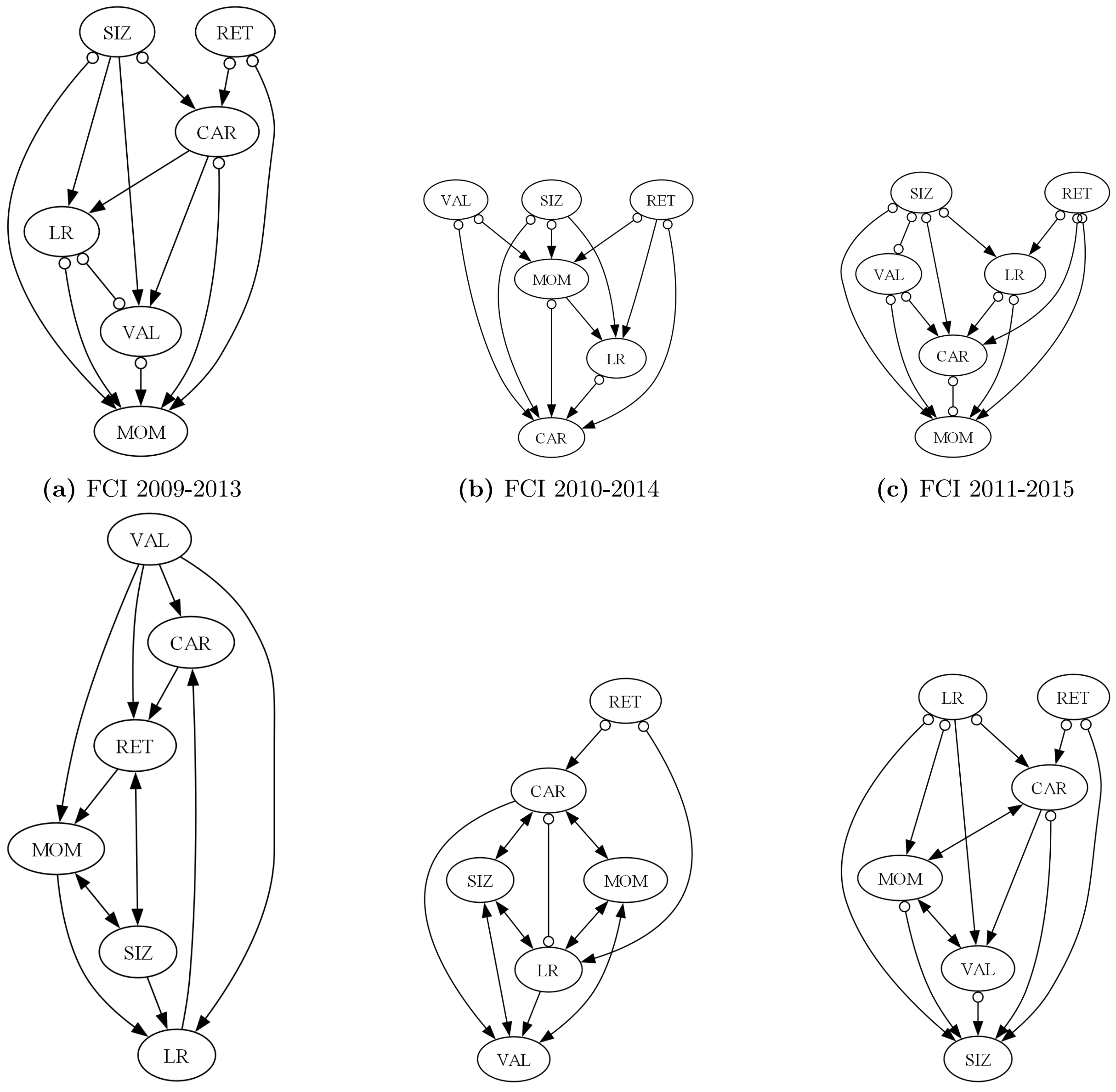
(b) FCI 2009-2020

(c) FCI 2009-2021

(d) FCI 2009-2022

Figure B.2: Causal graphs obtained via FCI expanding window (2/2)

Moving window



(a) FCI 2009-2013

(b) FCI 2010-2014

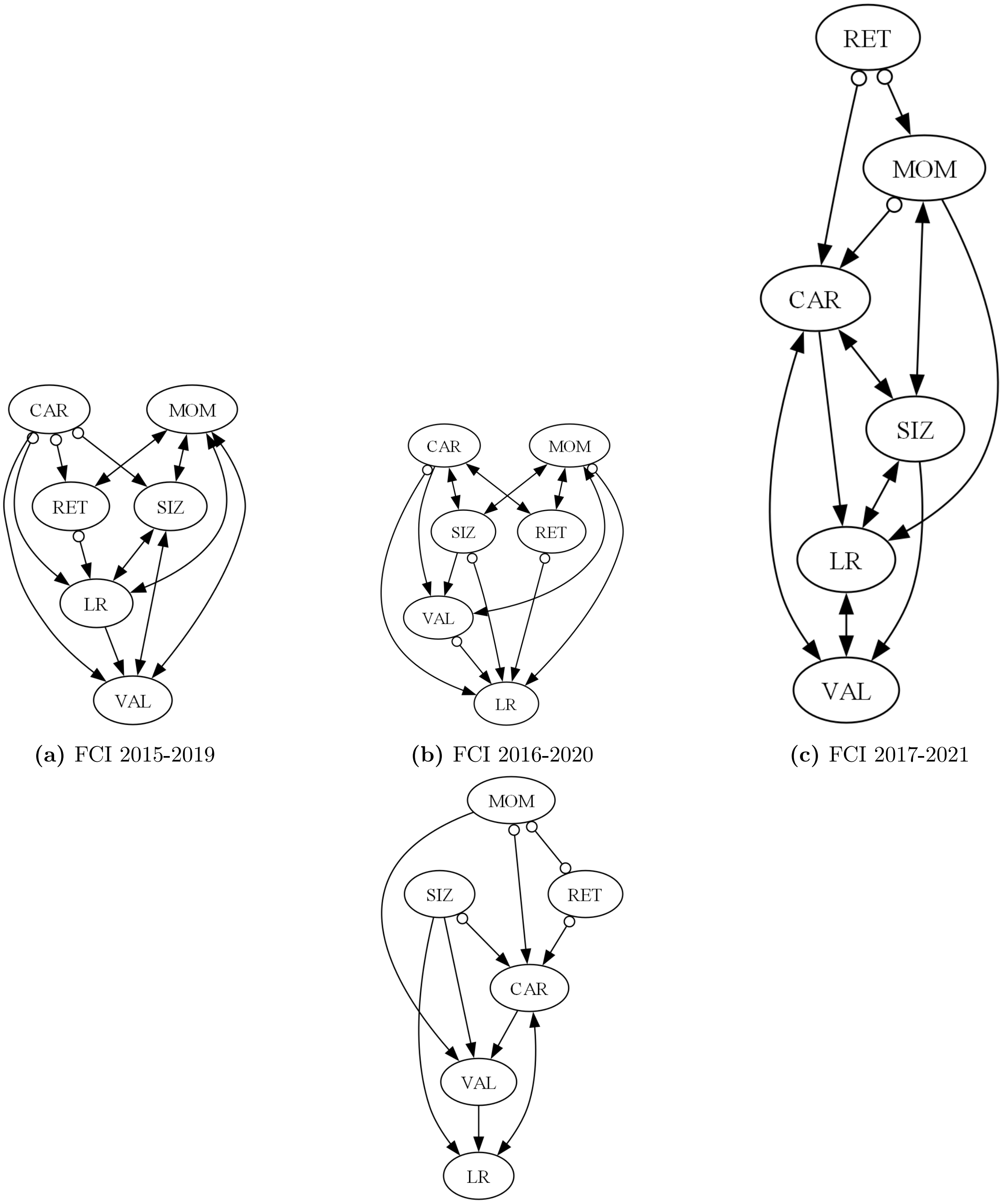
(c) FCI 2011-2015

(d) FCI 2012-2016

(e) FCI 2013-2017

(f) FCI 2014-2018

Figure B.3: Causal graphs obtained via FCI moving window (1/2)



(a) FCI 2015-2019

(b) FCI 2016-2020

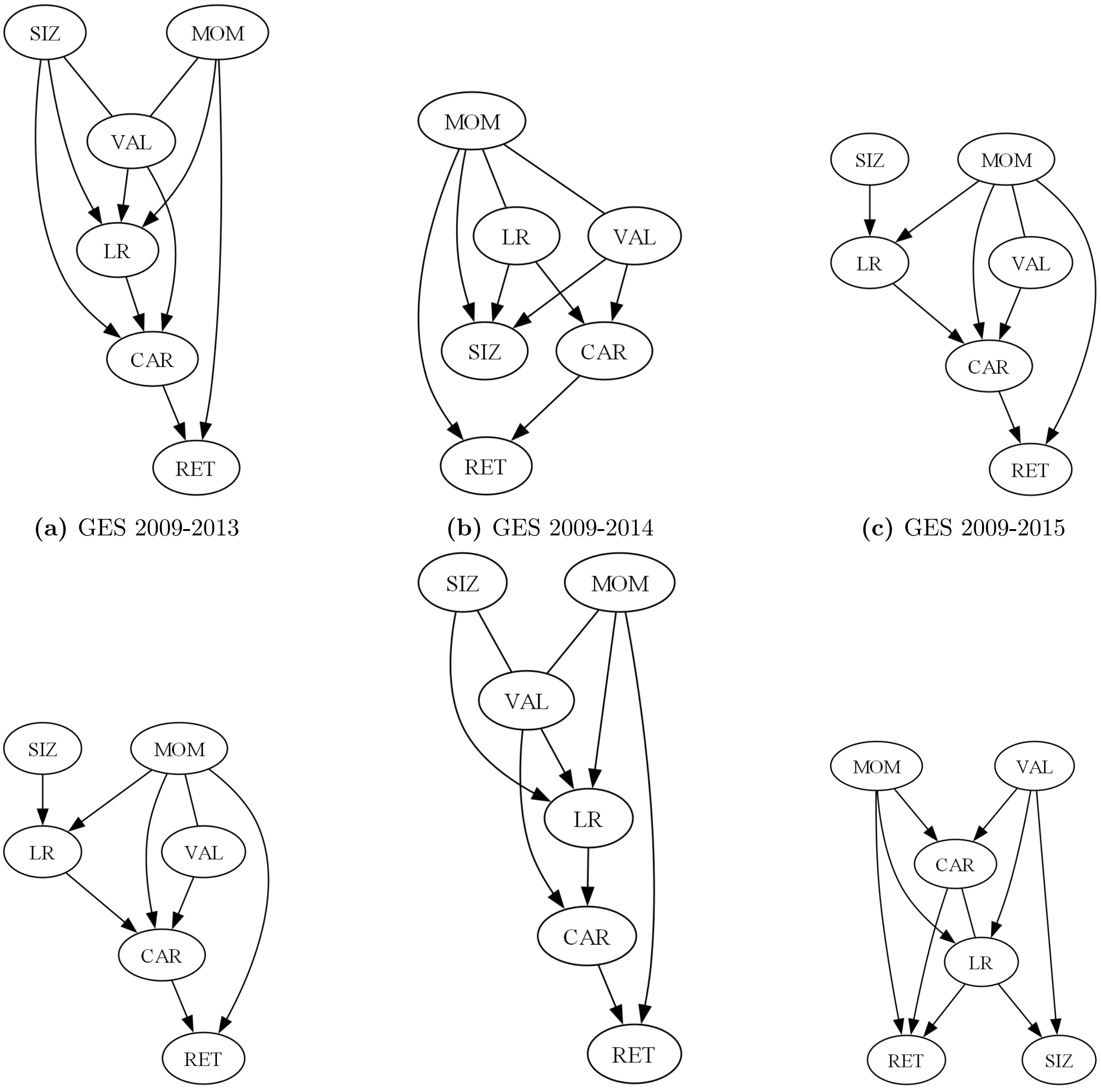
(c) FCI 2017-2021

(d) FCI 2018-2022

Figure B.4: Causal graphs obtained via FCI moving window (2/2)

B.2 GES

Expaning Window



(a) GES 2009-2013

(b) GES 2009-2014

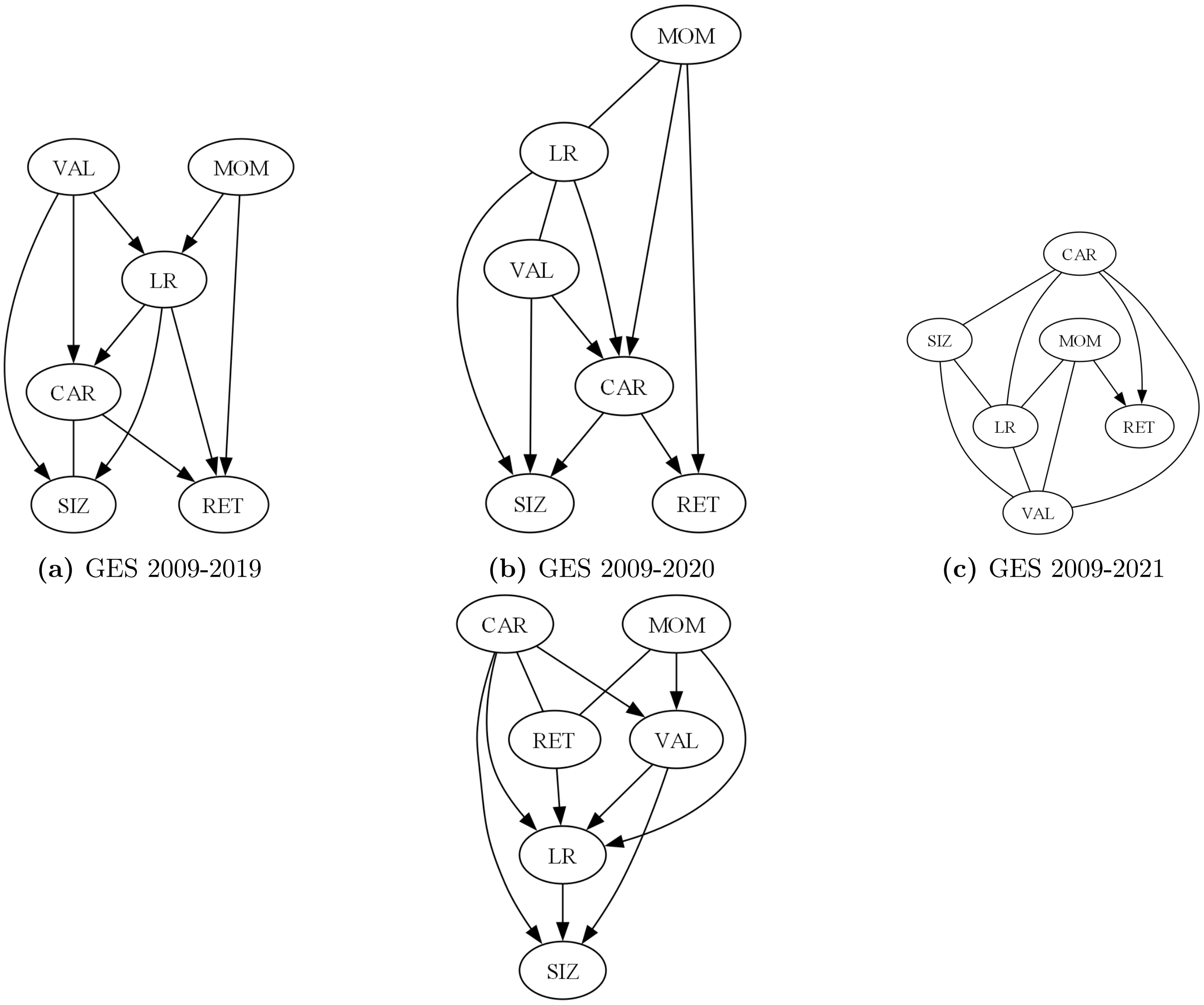
(c) GES 2009-2015

(d) GES 2009-2016

(e) GES 2009-2017

(f) GES 2009-2018

Figure B.5: Causal graphs obtained via GES expanding window (1/2)



(a) GES 2009-2019

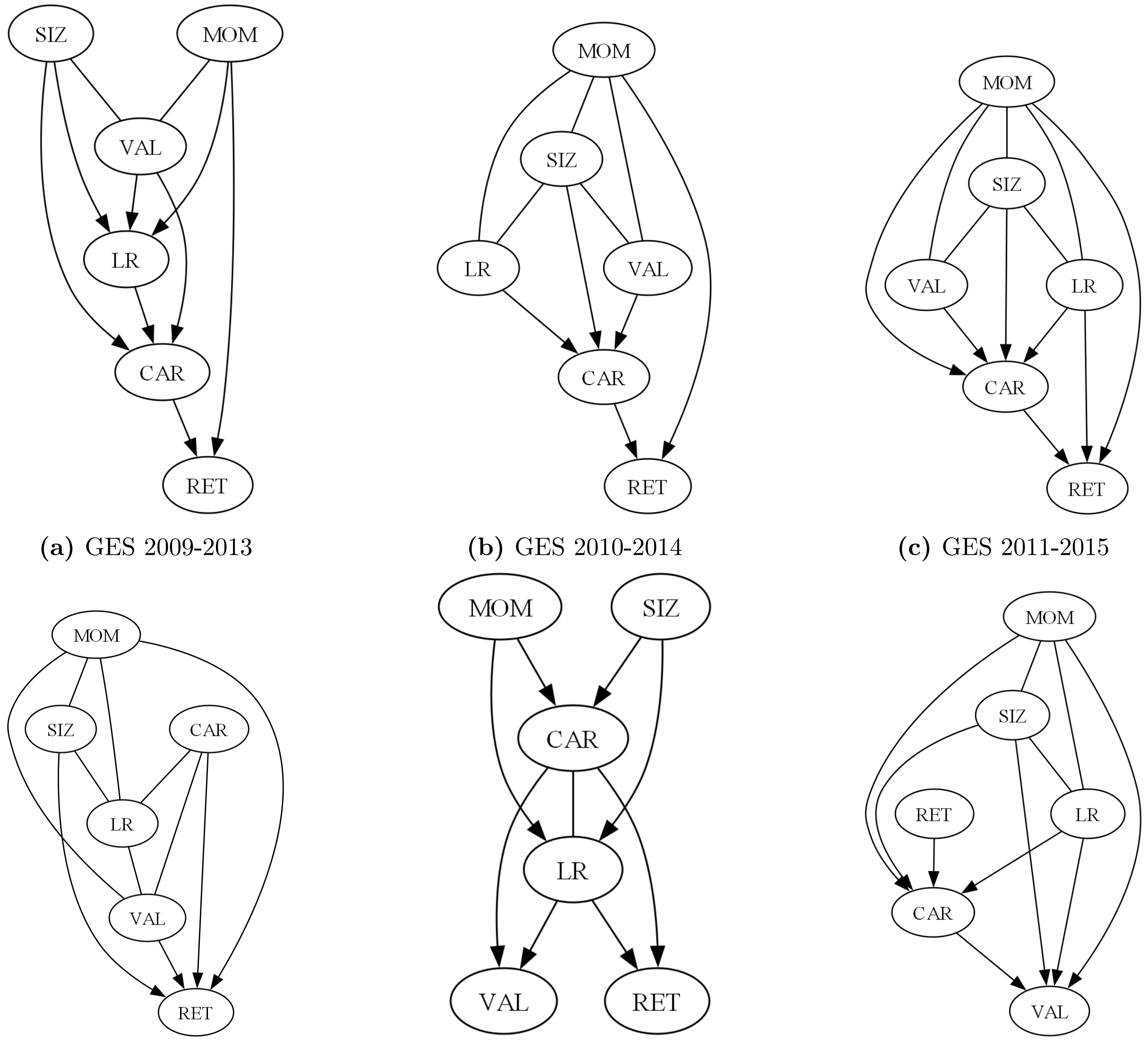
(b) GES 2009-2020

(c) GES 2009-2021

(d) GES 2009-2022

Figure B.6: Causal graphs obtained via GES expanding window (2/2)

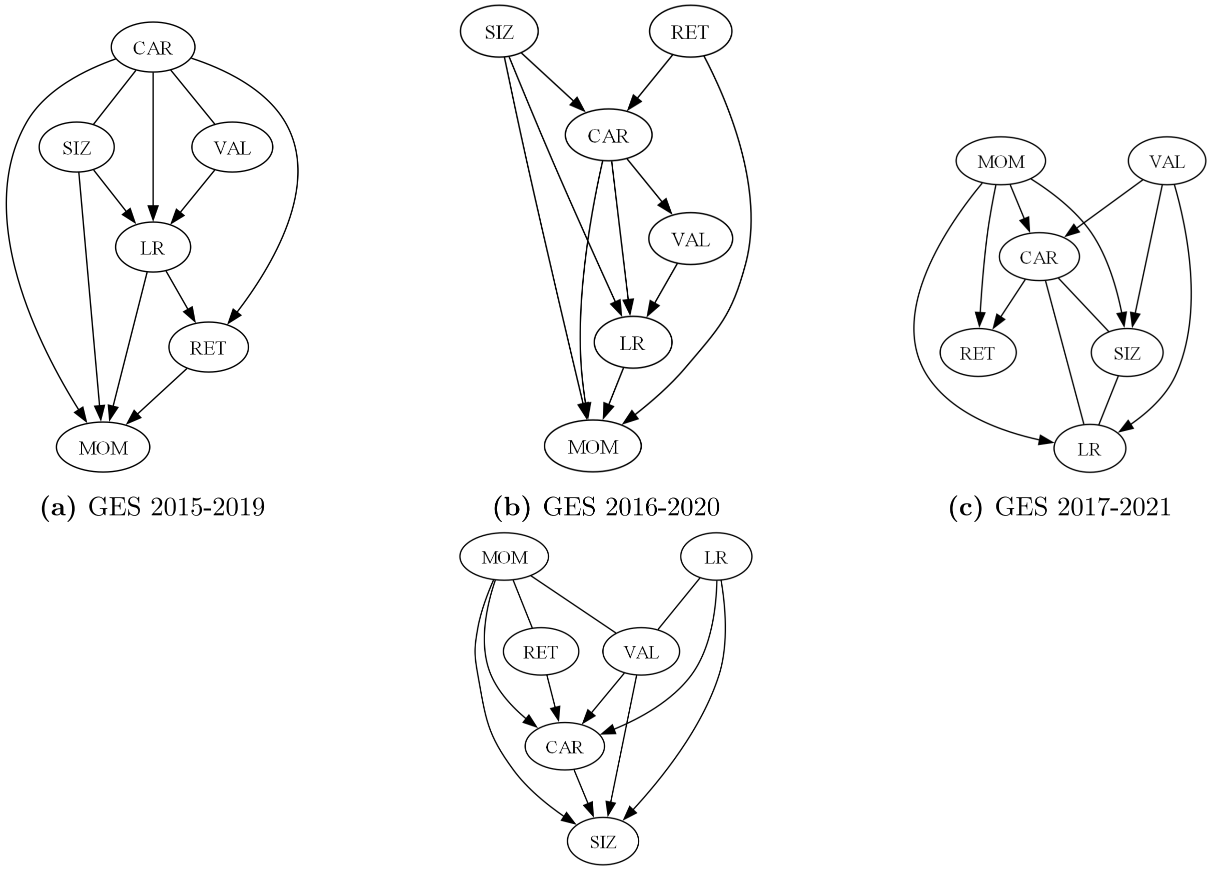
Moving Window



(a) GES 2009-2013 **(b)** GES 2010-2014 **(c)** GES 2011-2015

(d) GES 2012-2016 **(e)** GES 2013-2017 **(f)** GES 2014-2018

Figure B.7: Causal graphs obtained via GES moving window (1/2)



(a) GES 2015-2019

(b) GES 2016-2020

(c) GES 2017-2021

(d) GES 2018-2022

Figure B.8: Causal graphs obtained via GES moving window (2/2)

C | Numerical Results with Moving Window

Causal Analysis

Here we present the (direct) causal factors obtained via FCI and GES using a moving window.

FCI

Table C.1: (Direct) Causal Factors obtained with FCI (moving window)

Period	Causal Factors
2009-2013	(siz, val, lr, mom, car)
2010-2014	(siz, val, lr, mom, car)
2011-2015	(siz, val, lr, mom, car)
2012-2016	(siz, val, lr, mom, car)
2013-2017	(siz, val, lr, mom, car)
2014-2018	(siz, val, lr, mom, car)
2015-2019	(car)
2016-2020	(siz, val, lr, mom, car)
2017-2021	(siz, val, lr, mom, car)
2018-2022	(mom)

Period	Direct Causal Factors
2009-2013	(siz, val, lr, mom, car)
2010-2014	(siz, val, lr, mom, car)
2011-2015	(siz, val, lr, mom, car)
2012-2016	(siz, val, lr, mom, car)
2013-2017	(siz, val, lr, mom, car)
2014-2018	(siz, val, lr, mom, car)
2015-2019	(car)
2016-2020	(siz, val, lr, mom, car)
2017-2021	(siz, val, lr, mom, car)
2018-2022	(mom)

There are seven periods where FCI could not provide sufficient causal factors. This could imply that the causal graph was wrong or that other latent factors are the cause of month-ahead returns. Because there are only three years where (direct) causal factors could be identified, we will not use FCI in CFI using a moving window.

GES

The causal factors and direct causal factors, obtained via GES, are stated in Table C.2.

Only in two periods (2014-2018 and 2016-2020) did the GES algorithm fail to provide sufficient causal factors. This is significantly fewer than the seven insufficient graphs produced by the FCI algorithm. Therefore, we will use the causal factors and direct causal factors identified via GES for CFI, rather than those obtained via FCI. In the periods where a usable causal graph is available, there is always a minimum of three causal factors. The causal factors omitted by the causal selection algorithm (Table C.2) include value (2013-2017), momentum (2015-2019), low-risk (2017-2021), and size and carry (2018-2022).

Table C.2: (Direct) Causal Factors obtained from GES (moving window)

Period	Causal Factors
2009-2013	(siz, val, lr, mom, car)
2010-2014	(siz, val, lr, mom, car)
2011-2015	(siz, val, lr, mom, car)
2012-2016	(siz, val, lr, mom, car)
2013-2017	(siz, lr, mom, car)
2014-2018	(siz, val, lr, mom, car)
2015-2019	(siz, val, lr, car)
2016-2020	(siz, val, lr, mom, car)
2017-2021	(siz, val, mom, car)
2018-2022	(val, lr, mom)

Period	Direct Causal Factors
2009-2013	(mom, car)
2010-2014	(mom, car)
2011-2015	(lr, mom, car)
2012-2016	(siz, val, mom, car)
2013-2017	(lr, car)
2014-2018	(siz, val, lr, mom, car)
2015-2019	(lr, car)
2016-2020	(siz, val, lr, mom, car)
2017-2021	(mom, car)
2018-2022	(mom)

Empirical Asset Pricing via Machine Learning

The results are stated in Table C.3. The predictive performance of the machine learning methods using a moving window is slightly less accurate then using a expanding window as in Chapter 6.

Table C.3: Performance of machine learning methods on the prediction of month-ahead returns (moving window)

Model	Features	R^2_{OOS}	MSE
OLS	Regular	-0.0025	2.0307E-04
OLS	GES	-0.0032	2.0321E-04
OLS	D-GES	-0.0041	2.0338E-04
XGB	Regular	-0.0027	2.0311E-04
XGB	GES	-0.0031	2.0318E-04
XGB	D-GES	-0.0032	2.0320E-04
NN	Regular	-0.0024	2.0305E-04
NN	GES	-0.0013	2.0281E-04
NN	D-GES	-0.0025	2.0306E-04

Investment Strategy

In this section the results of Causal Factor Investing in the credit market using a moving window are stated. We have followed the investment strategy of CFI as described Section 5.3. For more information on the portfolio metrics, we refer the reader to Section 2.2.3. The cumulative returns are displayed in Figure C.1 and additional metrics are given in Table C.4.

Compared to results using an expanding window, the machine learning portfolios have slightly worse returns and slightly better risk metrics.

[1] Monthly average excess return over LIBOR

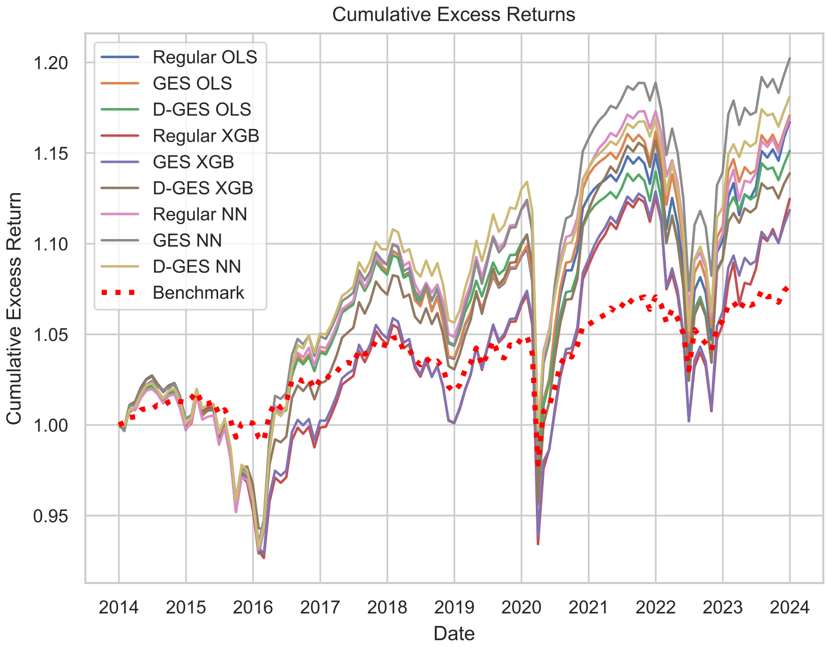


Figure C.1: Cumulative excess returns machine learning portfolios 2014-2023 (moving window)

Table C.4: Performance of machine learning portfolios (moving window)

Model	Features	Return[1]	Duration	z-Spread	TTM	Rating	IR
OLS	Regular	0.15	6.64	141.15	7.60	3.61	0.08
OLS	GES	0.15	6.69	140.65	7.65	3.65	0.08
OLS	D-GES	0.13	6.53	139.59	7.46	3.62	0.07
XGB	Regular	0.11	6.53	155.50	7.49	3.81	0.05
XGB	GES	0.11	6.71	146.21	7.70	3.79	0.04
XGB	D-GES	0.12	6.49	145.71	7.45	3.77	0.06
NN	Regular	0.14	6.35	138.34	7.22	3.70	0.09
NN	GES	0.17	6.39	143.32	7.28	3.76	0.10
NN	D-GES	0.15	6.44	145.20	7.37	3.69	0.09
Benchmark	-	0.07	5.06	73.84	5.58	3.38	nan