PhD Dissertation by Hugo Félicien Thomas Cazaux

Transformer-Based Timeseries Forecasting for Company Returns with Multi-Provider ESG Data

Transformer-Based Timeseries Forecasting for Company Returns with Multi-Provider ESG Data

PhD Dissertation by
Hugo Félicien Thomas Cazaux

Department of Engineering | Reykjavik University

2025

Transformer-Based Timeseries Forecasting for Company Returns with Multi-Provider ESG Data

Hugo Cazaux

Thesis of 180 ECTS credits submitted to the Department of Engineering at Reykjavík University in partial fulfillment of the requirements for the degree of Doctor of Philosophy

October 8, 2025

Thesis Committee:

Dr. Eyjólfur Ingi Ásgeirsson, Supervisor
Professor, Reykjavik University, Iceland

Dr. Hlynur Stefánsson, Co-advisor
Professor, Reykjavik University, Iceland

Dr. Ralph Rudd, Co-advisor
Assistant Professor, Reykjavik University, Iceland

Dr. Sverrir Ólafsson, Co-advisor
Professor, Reykjavik University, Iceland

Dr. Marco Raberto, Co-advisor
Professor, University of Genoa, Italy

Dr. Patrice Bellot, Examiner
Professor, Aix-Marseille Université, CNRS, France

Dr. Linda Ponta, Examiner
Associate Professor, University of Genoa, Italy

ISBN (Print) 978-9935-539-88-5
ISBN (Electronic) 978-9935-539-89-2
ORCID 0000-0002-6026-5831

I dedicate this work to everyone who has supported me in the my endeavors: family, friends, teachers, colleagues.

Contents

List of Figures

List of Tables

Transformer-Based Timeseries Forecasting for Company Returns with Multi-Provider ESG Data

Hugo Cazaux

2022-05-23

Abstract

The goal of this thesis is to analyze the effectiveness and impact of ESG ratings through the lens of machine learning interpretability. We first start by motivating the thesis through studying the correlation between ESG ratings and controlled log returns. We demonstrate that there is a correlation between the growth of a company and commitment to sustainable initiatives. We then establish two new machine learning models: the Non-stationary inverted Transformer (NSiTransformer) and the Centralized Multi-Agent framework with Attention (CMAA). We compare these new models against state-of-the-art methods and benchmarks and develop ad-hoc interpretability tools to harness their insights. We then construct a dataset using traditional financial data and diverse ESG data from various providers. The NSiTransformer and CMAA are then applied to this new dataset, respectively for timeseries predictions at several prediction length and for fine-tuning of a timeseries predictor. We find that the inclusion of ESG ratings in the dataset, especially from various providers, improves the performance of the models. Through interpretability, we pinpoint which features of the dataset are contributing the most for a given prediction. We conclude that using interpretability of machine learning models is a valid approach to discover patterns that might escape traditional statistical analysis. We also conclude that ESG ratings are worth integrating in financial predictions and have the potential to increase performance. We compare this property to other slow-moving indicators that have been determined to be beneficial for financial predictions.

Acknowledgments

> If we see the laws of production as 'physical' or 'mechanical' and thereby apolitical, no distribution of wealth is inherently just or unjust.
>
> Guido Alfani

This work was funded by the Sustainability Institute and Forum (SIF), the Reykjavik University Teacher Assistant Grant, and the Reykjavik University Research fund. A special thanks to the SIF for the funding of the Sustainalytics database access.

Publications

Published:

- H. Cazaux, R. Rudd, H. Stefánsson, S. Ólafsson, M. Raberto, and E. I. Ásgeirsson, "Correlation study between returns and ESG ratings.," Journal of Impact & ESG Investing, vol. 5, no. 1, 2024.
- Cazaux, H., Rudd, R., Stefánsson, H., Ólafsson, S., & Ásgeirsson, E. I. (2024, December). Centralized Multi-Agent Proximal Policy Optimization with Attention. In 2024 International Conference on Machine Learning and Applications (ICMLA) (pp. 834-840). IEEE.
- H. Cazaux, R. Rudd, H. Stefánsson, S. Ólafsson, and E. I. Ásgeirsson, "Inverted transformers interpretability beyond attention visualization," in International Joint Conference on Neural Networks, 2025.
- H. Cazaux, R. Rudd, H. Stefánsson, S. Ólafsson, and E. I. Ásgeirsson, "Fine-tuning timeseries predictors using reinforcement learning," To be published in Recent Advances in Deep Learning, Taylor Francis, 2025.

In-review:

- Cazaux, H., Rudd, R., Stefánsson, H., Ólafsson, S., & Ásgeirsson, E. I. (2024). Non-Stationary iTransformer With Time2Vec Embeddings, Transactions on Artificial Intelligence. IEEE
- H. Cazaux, R. Rudd, H. Stefánsson, S. Ólafsson, and E. I. Ásgeirsson, "Controlled log returns prediction using nsitransformer on esg enhanced timeseries," in Submitted at Journal of Sustainable Finance Investment, Taylor Francis., 2025.

Part I

Introduction

Chapter 1

The Intersection of ESG Ratings and Financial Performance

This chapter discusses the motivations behind the creation of ESG ratings and this thesis. This chapter also details the research questions, objectives, methodological approach and the outline of the thesis.

1.1 Introduction

In the past decades, stakeholders have increased the pressure on firms to push a more sustainable agenda [1]. In the past, scholars argued that this call for more socially and environmentally responsible initiatives was a departure from the traditional goal of firms, that is to maximize shareholder wealth [2]. But recent research suggests that these initiatives might not be mutually exclusive, as the long-term benefits of adopting sustainable initiatives can outweigh the short-term cost of adopting them [3]. As such, stakeholder theory is often used as a tool to bridge the gap between firm performance and environmental impact. Stakeholder theory states that for a firm to be successful in the long run, it has to create value for all types of stakeholders, not only employees, suppliers, and shareholders but also customers, political groups and trade unions [4]. As the interests of these groups can greatly differ yet all are essential to maximize the efficiency of a firm, stakeholder theory suggests that as a firm, doing good helps you do well [5]. While a positive effect is still debated among scholars, there is an increasing demand to quantify the quality of these sustainable initiatives [6].

ESG (Environmental, Societal and Governance) metrics were introduced in 2004 to formalize some of the non-financial aspects of companies and how they influence society, including their environmental impact [7]. The ESG metrics are used to estimate how well companies are doing regarding sustainability, social

rights, consumer protection, animal welfare, ethics and governance. This impact can be difficult to quantify, and several methodologies have emerged to calculate those metrics. Private data brokers such as Bloomberg [8] or Reuters [9] propose their own metrics, and companies such as Sustainalytics [10] specialize in rating companies. ESG ratings can be calculated from yearly financial disclosure, publicly available information, or a mixture of both. This inconsistency in approach among the different data providers creates a complicated trust dynamic in the metrics, as the methodology employed is often opaque. Finance benefits from traditionally standardized data, as opposed to ESG metrics from different providers that might be on a different scale or evaluating criterias.

The initial strategy based on ESG ratings relied on screening [11]. Screening consists of selecting a limit based on the ESG ratings below which a fund or individual will not invest in an asset. Scholars found that from a risk perspective, ESG screening has a positive effect on performance, with a reduced volatility, drawdowns and CVaR. This strategy corroborates the findings of numerous studies [12] linking efficient sustainable initiatives to better access to finance [13] [14], lower cost of capital [15] [16] [17], increased customer and employee satisfaction [18], higher levels of institutional ownership [19], higher controlled returns [20], and greater social capital [21]. But this firm value is qualitative, and mathematical tools are necessary to demonstrate a quantitative benefit to green initiatives. Recent literature shows that investors use ESG ratings in both reporting and product strategy, despite a lack of reporting standards often cited as a severe drawback, weakening comparability and reliability. Recent studies also demonstrated ESG disclosure to benefit lower capital constraints [13], fewer analyst forecast errors [14], or specifically in industry-specific classifications a more accurate prediction of financial performance [22].

As ESG ratings have cemented themselves as a valuable lens through which corporate responsibility can be assessed, their integration in finance remains challenging. One of the most uphill problem is the lack of quality data [23] [24], which compounds with the innate complexity of financial data. This scarcity undermines the quality of the relationship that can be drawn between financial and ESG data [25], as the granularity can also vary depending on the data provider. Nonetheless, the consensus is that the relationship between sustainable initiatives and financial performance is an intricate, multi-faceted issue, and as such an excellent candidate for machine learning, a field that has become at the forefront of exploiting complex non-linear patterns.

Machine learning (ML) is classically defined by Arthur Samuel as the ”field of study that gives computers the ability to learn without being explicitly programmed”. ML subdivides into a multitude of algorithms and paradigms that have seen tremendous success in finance [26], healthcare [27], environmental science [28], language translation [29]. These models excel at modeling non-linear dependencies in data rich environments, and have become increasingly prominent

in virtually any field. However, the ubiquity of machine learning raises concerns about responsibility and accountability [30], as the "black-box" nature of these models remove a degree of control that previously was mathematical or human. In response to this, the research community proposed a number of interpretability techniques, to better understand the decisions taken by the models [31] [32].

Interpretability can be model-agnostic or model-specific [33] [34], but both have the same function on a fundamental level: provide an explanation of the inner working of a complex machine learning model, trained on a highly-dimensional, dense dataset [35]. Specifically, Murdoch & al. define interpretability as "the extraction of relevant knowledge from a machine-learning model concerning relationship either contained in data or learned by the model". This capacity greatly varies from one algorithm to the other, and can be further divided between local interpretability, and global interpretability. A linear regressor can be interpreted up to a high level of dimensions, as the weights of the sum of features represent the model itself [36]. Local interpretability often designates a set of techniques, with the Shapley values [37] at the forefront, that draw a relationship between the input data and the output data, treating the model itself as a close system [37]. But complex models can be built with interpretability in mind, and it can be extremely beneficial for the performance of the model and the understanding of the dataset.

This thesis places itself at the intersection of ESG ratings, finance, machine learning and interpretability. It posits that in a complex ecosystem ruled with non-linear relationships that statistical models fail to capture, sophisticated machine learning algorithms can propose valuable insights. The interpretability imbued in the architecture of the machine learning models proposed in this thesis is used as a tool to better understand the relationship between the features of the dataset. This high level idea turns the table on Arthur Samuel's quote, from giving computers the ability to learn, to giving us the opportunity to learn from them [38]. Finance and ESG ratings are the focus of this thesis, but the broader scope encompasses the idea that machine learning interpretability can be a tool to measure the importance of features in any given dataset and improve model accuracy. Beyond the human common sense, machine learning algorithms have proven to find similarities where we thought there was none [39], and harnessing this idea to help models focus on the important parameters of a dataset is a key contribution of this thesis.

1.2 Research Questions and Objectives

The research questions investigated in this thesis are the following:

RQ1 Do ESG ratings carry signal for future financial performance, and how does that vary by sector & materiality?

RQ2 How can advanced, interpretable ML, including RL fine-tuning, improve time-series forecasts?

RQ3 Can built-in interpretability quantify the specific contribution of ESG features?

The objectives of this thesis are the following:

- To establish a foundational understanding of the interplay between ESG ratings and financial performance and determine empirically the contribution of ESG ratings in financial predictions.
- To extend interpretable ML frameworks for time-series prediction and RL fine-tuning.
- To apply those frameworks to ESG-enhanced financial forecasting, quantifying ESG impact.
- To assess the practicality of interpretability tools to determine the contribution of a variable in the dataset.

1.3 Overview of Methodological Approach

The core idea developed in this thesis is to investigate what factors drive the value of an asset. This thesis starts by examining whether ESG ratings exhibit a meaningful relationship with financial performance through correlation-based analyses. The complexity of real-world financial systems and the non-linear dynamics at stake limit the ability of simple correlation analysis to capture these relationships.

Recognizing the limits of a linear methodology, the thesis transitions into advanced machine learning models to deepen the analysis. These models are rooted in cutting-edge frameworks, like the inverted transformer or multi-agent reinforcement learning, but are designed with interpretability at their core. This caution in implementation complements the initial ESG correlation study, as shown in the application of the models.

The broader idea behind this thesis is the concept that machine learning interpretability can be a lens to evaluate the relevance and usefulness of features and metrics. A constant challenge in machine learning is the sheer volume of data. This volume can be caused by extensive historical data, but also through an excess of features, some of which contribute little to nothing to the predictive power of the model. This idea is not only applicable to finance, but rather to a wide range of fields.

This approach has implications far beyond the realm of ESG ratings or financial analysis. In healthcare, it could optimize diagnostic processes by identifying

the minimal set of medical tests required for accurate diagnoses. In civil engineering, it could isolate the key structural metrics that influence the integrity of a building, guiding cost-effective maintenance. By applying machine learning interpretability to ESG metrics, this thesis establishes a broader conceptual framework for isolating the most impactful factors in any dataset, regardless of the domain.

By combining traditional statistical techniques with interpretable machine learning, this thesis exemplifies a comprehensive methodology that not only evaluates the role of ESG ratings in financial systems but also offers a scalable framework for tackling similar challenges in other fields. By contributing both to the methodology and the understanding of the importance of ESG metrics, this thesis places itself as a significant step forward in the intersection of finance, machine learning, and interpretability.

1.4 Thesis Outline

This thesis is divided in four major parts. The first and current part introduces the research questions and themes of the thesis. It serves as an introduction and contextualization of the thesis within the diverse fields that are discussed. The second part is centered around the first paper "Correlation Study between Returns and ESG Ratings" [40] and the preliminary study that was conducted in it. This study introduces ESG ratings in greater details than in the introduction, and proposes to calculate and discuss the correlation between controlled returns and ESG ratings. The third part is dedicated to 2 machine learning models developed to further analyze the role of ESG ratings in financial datasets. This part is divided in two distinct chapters: the first chapter details the non-stationary inverted transformer, a time-series prediction model, introduced in "Non-Stationary iTransformer With Time2Vec Embeddings",[41] and further developed in "Inverted Transformers Interpretability Beyond Attention Visualization" [42]. The second chapter is dedicated to reinforcement learning and introduces the centralized multi-agent reinforcement learning framework, which was presented in "Centralized Multi-Agent Proximal Policy Optimization with Attention". The fourth part is an application of both machine learning models to an ESG-enhanced financial dataset. This part first focuses on prediction after supervised learning, as detailed in "Controlled Log Returns Prediction Using NSiTransformer on ESG Enhanced Timeseries" [43]. The reinforcement learning framework is then used for fine-tuning, as proposed in "Fine-tuning Using Centralized Multi-Agent Proximal Policy Optimization" [44]. The various interpretability techniques embedded in the models are then used to glean insights about the contribution of ESG ratings in the predictive power of the models.

Part II

Motivation: Correlation Between ESG Ratings and Financial Performance

Chapter 2

Empirical Study: ESG Ratings and Financial Returns

This chapter details the initial correlation study between ESG ratings and financial returns. The goal is to propose a first empirical analysis of the relationship between ESG and financial features, which will be further explored in Chapters 1 and 7 using machine learning models.

2.1 Introduction

Investments and finance are critical components of society, for governments, companies, and individuals. The global asset management market, historically focused on maximizing return on investment [45], strategically balances risk and reward in a complex financial environment to optimize investors' wealth. Investments drive the activities of companies and are essential for their survival and create clear incentives for companies to tailor their activities to suit the priorities of investors. There is a growing focus on sustainability and ethical behavior of companies [46]. The investment industry is a key driver in moving companies to be more sustainable and help reaching the UN Sustainable Development Goals [1].

Multiple initiatives have already been undertaken to foster more sustainable investment practices. Specifically, the inclusion of Environmental, Social, and Governance (ESG) metrics [47] in asset management has seen steady growth, indicating a paradigm shift towards sustainable, socially responsible, and ethically governed investment strategies. The ESG metrics are used to estimate how well companies are doing regarding sustainability, social rights, consumer protection, animal welfare, business ethics, and governance. ESG ratings have been shown to influence the systemic and idiosyncratic risk of companies [48]. The Global Sus-

tainable Investment Alliance defines ESG integration as "the systematic and explicit inclusion by investment managers of environmental, social and governance factors into financial analysis" [49].

The research questions investigated in this chapter are the following: is there a correlation between the ESG ratings and returns of companies? Can the company sectors help identify groups with different relationships? Can the materiality issues provide more explanatory power?

This paper is structured as follows: Section 2.2 includes a contextualization and a review of the existing literature on ESG ratings. Section 2.3 describes the dataset used and the segmentation between complete data and materiality. Section 2.3 examines the methodology used to control for market factors and compute the correlation. Section 2.4 contains details on the results, average correlation per sector, and statistically significant correlation for individual companies. Section 2.6 contains results of the correlation between variations of returns and ESG metrics. Section 2.7 is a discussion of their implications for investors and companies alike. Section 2.8 is the conclusion of the study and section 2.9 presents the motivation for machine learning approaches.

2.2 Literature Review

The search for a relationship between sustainability and corporate performance can be traced back to the 1970s [50]. Scholars have studied the impact on branding [51], market longevity [52], and equity valuation [48]. Results of these studies indicate that failing to communicate strong ESG performance, specifically expressing low carbon emissions and employee satisfaction, reduces the odds for external financing and increases both the systematic and stock-specific risks. Studies have discussed the authenticity of the disclosure by companies [53], finding that disclosure can weaken the negative or positive valuation effects on company. Scholars also examined the integration of ESG ratings in portfolio strategies [54] [55], showing that a strong ESG rating will attract long-term-oriented investors with a lower sensibility to immediate negative earnings [56].

While the exploration of ESG ratings and their financial implications has gained momentum in recent years, the underpinnings of this research lie in foundational asset pricing theories. Asset pricing theories, evolving over the decades, provide the scaffolding for understanding the determinants of asset prices and returns. The Capital Asset Pricing Model (CAPM) is a seminal theory in this domain, introduced by [57] and [58]. CAPM posits that the expected return on an asset is a function of its systematic risk, often measured by its beta relative to the market. While the model offers a simplistic view, it laid the foundation for subsequent models that incorporated multiple factors. Recognizing the limitations of CAPM, [59] introduced the Fama-French three-factor model, adding size and value fac-

tors to the market risk factor in CAPM. This model was further expanded into the Fama-French five-factor model by [60], incorporating profitability and investment factors, offering a more comprehensive understanding of asset returns.

As ESG ratings became more standardized and prevalent, the focus shifted towards these quantifiable metrics. Studies such as those by [61] and [62] respectively explored the difference in performance between ethical and traditional funds and the alignment between economic factors and corporate environmental management. Their findings were mixed, as on one hand older ethical funds were either under performing or matching the index, but on the other hand eco-efficiency relates positively to operating performance and market value. Some research indicated a positive relationship between high ESG scores and superior stock returns [63], [64]. Fewer studies delve into the causality of this relationship. [65], for instance, suggested that firms with a long-term focus on ESG issues tend to outperform their counterparts in the long run, hinting at a potential causal link between ESG practices and financial performance.

As the ESG landscape continues to evolve, tools and frameworks that offer a more standardized approach to materiality assessment are emerging. Among these, the Sustainable Accounting Standards Board (SASB) materiality map stands out. SASB's approach to materiality emphasizes the financial materiality of ESG issues, identifying which issues are likely to affect the financial performance of companies within specific sectors [66]. Studies have praised the data quality of the reporting [67], specifically through the narrative of linking financial data to non-financial data. The data has been used to evaluate performance of firms with different level of materiality ratings[22], finding that firms with strong ratings on material sustainability issues have better future performance than firms with inferior ratings. [68] concluded that scholars interested in understanding how sustainability information impacts economic value and stock prices need to incorporate a materiality lens into their analysis.

2.3 Dataset Description and Preprocessing

2.3.1 Data Collection

For this preliminary study, the primary source of data was Thomson Reuters (now known as Refinitiv Eikon), offering a comprehensive dataset to investigate the relationship between ESG ratings and stock returns. The data used in this work was pulled in February 2022 and consists of the companies listed on the S&P500 at this point in time. Below are the fields obtained through the Thomson Reuters data stream.

- Date: Captures the date of each data entry, essential for time-series analysis and understanding temporal trends.

- Instrument: Denotes the specific stock or financial instrument under consideration.
- Open Price, Close Price, High Price, Low Price: These columns detail daily stock prices, crucial for computing daily and subsequently, annual log returns.
- ESG Score: An aggregate rating based on a firm's adherence and performance across environmental, social, and governance dimensions.
- Environmental Pillar Score: Focuses solely on a company's environmental practices and impacts.
- Social Pillar Score: Reflects how a company fares in social responsibilities, such as labor practices, product responsibility, and community relations.
- Governance Pillar Score: Offers insights into a firm's governance structures, ethical practices, and overall corporate accountability.

While the stock data was updated on a daily basis, the ESG scores were updated annually at each new fiscal year. Revisions might happen if a scandal becomes public, or after a quarterly earnings call. General ESG policies often imply important structural changes in a company and long term plans, which might not reflect significantly over months. As such, this periodic update typically mirrors annual disclosures of key metrics. Scores range from 0 to 100.

In order to control the results with established financial frameworks, additional data will be incorporated. Established market factors from the Fama-French five-factors (FF5) models were obtained from the website of one of the creator of the model [69]. The values of these factors are constructed using the 6 value-weight portfolios formed on size and book-to-market, the 6 value-weight portfolios formed on size and operating profitability, and the 6 value-weight portfolios formed on size and investment. The portfolios used are from the proof of concept available on the Fama-French 5 data library [70]. The coefficients are:

- Rm-Rf : Excess return on the market.
- Small Minus Big (SMB): Average return of the nine small stock portfolios minus the average return on the nine big stock portfolios.
- High Minus Low (HML): Average return on the two value portfolios minus the average return on the two growth portfolios.
- Robust Minus Weak (RMW): Average return on the two conservative investment portfolios minus the average return on the two aggressive investment portfolios.

- Conservative Minus Aggressive (CMA): Average return on the two value portfolios minus the average return on the two growth portfolios.

To contextualize and provide further explanatory power to this study, SASB materiality data were gathered. Their website [71] provides a Materiality Finder tool used to look up individual companies and which of the 26 issues are recognized as significant, mapping out the relevant issues for each company in the S&P500. Tables 2.1 and 2.3 summarize the datasets created for this study.

Table 2.1: Summary of dataset variables with their type and frequency. Noticeably, the ESG metrics are the temporal bottleneck, as the scores provided by Reuters are updated yearly.

Variable	**Nature**	**Frequency**	**Description**
Date	Time-series	Daily	Date of data entry.
Instrument	Categorical	-	Specific stock or financial instrument.
Sector	Categorical	-	GICS sector of the company.
Open Price	Continuous	Daily	Opening price of the stock.
Close Price	Continuous	Daily	Closing price of the stock.
High Price	Continuous	Daily	Highest price of the stock during the day.
Low Price	Continuous	Daily	Lowest price of the stock during the day.
ESG Score	Continuous	Yearly	Aggregate ESG rating.
Environmental Pillar Score	Continuous	Yearly	Rating based on environmental practices.
Social Pillar Score	Continuous	Yearly	Rating based on social responsibilities.
Governance Pillar Score	Continuous	Yearly	Rating based on governance structures.
Rm-Rf	Continuous	Daily	Excess return on the market (FF5).
SMB	Continuous	Daily	Small Minus Big (FF5).
HML	Continuous	Daily	High Minus Low (FF5).
RMW	Continuous	Daily	Robust Minus Weak (FF5).
CMA	Continuous	Daily	Conservative Minus Aggressive (FF5).

There are significant gaps in the availability of ESG data. Out of the 501 unique companies on the S&P index, every company exhibited at least one year of missing ESG data. 10 companies had no ESG data available and were removed from the dataset, bringing the total to 491. Figure 2.1 provides a year-by-year breakdown of the number of companies with complete ESG data. Notably, the years 2000-2005 have the most pronounced data omissions. This can be explained by a lesser focus on ESG ratings from companies at this time. We place our cut off in 2006. The final dataset contains 198 firms. This dataset is then used to calculate the correlation between ESG Metrics and returns for each ticker.

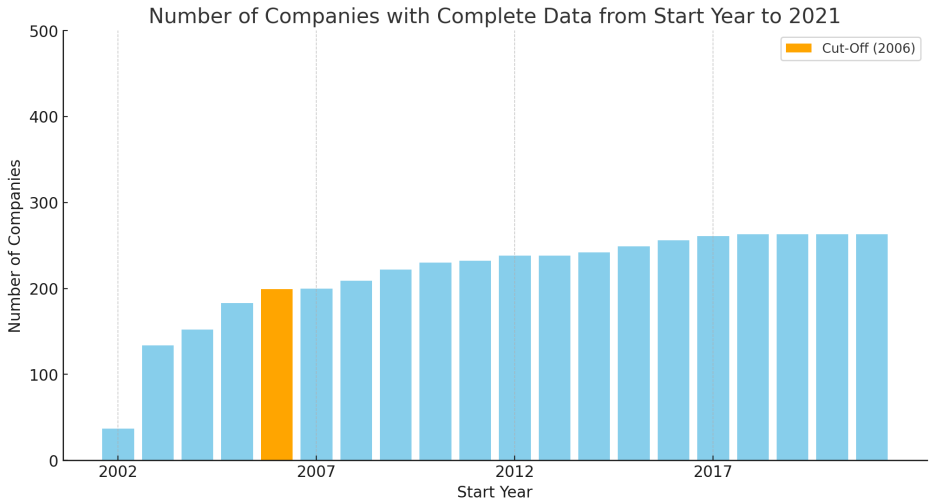


Figure 2.1: Number of Companies With Complete ESG Data Over The Years.

2.3.2 Sectors

The data is separated by sectors. The sectors are determined by the GICS [72]. There are 11 sectors: Industrials, Financial Services, Healthcare, Consumer Cyclical, Consumer Defensive, Real Estate, Utilities, Technology, Basic Materials, Energy, Communication Services. Table 2.2 presents the number of companies in each sector before and after accounting for missing ESG data. The most represented GICS sector is the Industrial sector with 32 companies, while the least represented is Communication Services with 5 companies. Sectors Industrials and Basic Materials retained respectively 43.1% and 70% of their population after the cut-off. The Technology sector was left with 11.3% despite having the second highest number of companies. The Communication Services was also left with 21.7%, but encompasses only 23 companies before cut-off.

Table 2.2: Number of Companies and Unique Tickers in Each Sector.

	# of Companies	# of Companies after Cut-off	% of Companies Left after Cut-off
Industrials	73	32	43,1%
Financial Services	67	32	47.8%
Consumer Cyclical	57	25	43.9%
Healthcare	64	25	39%
Utilities	28	17	60.7%
Consumer Defensive	36	16	44.4%
Basic Materials	20	14	70%
Real Estate	31	14	43,1%
Energy	20	10	45.2%
Technology	71	8	11.3%
Communication Services	23	5	21.7%
Total	491	198	40.32%

2.3.3 Materiality

Table 2.3: Summary of SASB Flags

Variable	Category
GHG Emissions	Environment
Air Quality	Environment .
Energy Management	Environment
Water & Wastewater Management	Environment
Waste & Hazardous Materials Management	Environment
Ecological Impacts	Environment
Human Rights & Community Relations	Social Capital
Customer Privacy	Social Capital
Data Security	Social Capital
Access & Affordability	Social Capital
Product Quality & Safety	Social Capital
Customer Welfare	Social Capital
Selling Practices & Product Labeling	Social Capital
Labor Practices	Human Capital
Employee Health & Safety	Human Capital .
Employee Engagement, Diversity & Inclusion	Human Capital
Product Design & Lifecycle Management	Business Model and Innovation
Business Model Resilience	Business Model and Innovation
Supply Chain Management	Business Model and Innovation
Materials Sourcing & Efficiency	Business Model and Innovation
Physical Impacts of Climate Change	Business Model and Innovation
Business Ethics	Leadership and Governance
Competitive Behavior	Leadership and Governance
Management of the Legal & Regulatory Environment	Leadership and Governance
Critical Incident Risk Management	Leadership and Governance
Systemic Risk Management	Leadership and Governance

Materiality refers to an individual factor within a sector that influences a firm's financial performance. The SASB Materiality Standards help to increase the granularity of the analysis. Specifically, the materiality standards were used to highlight which score correlates the most with performance in the returns. Materiality factors can be found on the SASB website using a tool called "Materiality Finder" [71]. This website was scraped for each company and the data stored in a binary vector. Each company can appear in multiple fields. Table 2.3 summarizes the different materiality issue recognized by SASB. Figure 2.2 presents the number of companies per SASB issue before and after the cut-off. The most represented

field is Product Design and Lifecycle Management with 117 companies. The least represented field is Competitive Behavior with 15 companies, with the exception of Customer Privacy and Physical Impacts of Climate Change that are not represented in the dataset of companies with complete data post cut-off.

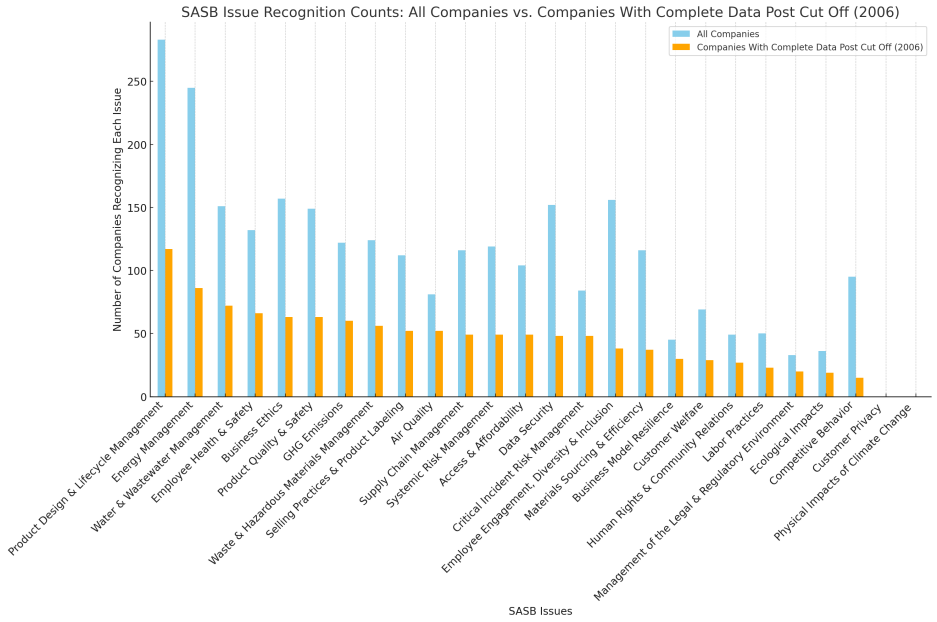


Figure 2.2: Number of Total Companies Per SASB Issue.

2.4 Methodology: Correlation Analysis

2.4.1 Data Integration and Analytical Model

Since the ESG data is yearly, to assess the annual performance of each stock, annualized returns were calculated. In order to obtain additive properties, returns are logged [73]. Daily log returns for a company are given by: $r_t = \ln\left(\frac{P_t}{P_{t-1}}\right)$ where r_t represents the log return at time t, P_t is the closing price at time t and P_{t-1} is the closing price at time $t-1$.

Given daily log returns, the annualized log returns for a company are computed as:

$$r_{annualized} = \sum_{i=1}^{n} r_i \tag{2.1}$$

where $r_{annualized}$ is the annual log returns, r_i is the ith daily log return, and n the number of trading days in a year. For the rest of this dissertation, annualized log returns will be referred to as returns.

2.4.2 Controlling for Market Factors

To isolate stock-specific characteristics, common market factors were controlled using the Fama-French Five-Factor Model. The model is given by:

$$R_{it} - R_f = \alpha_i + \beta_m(R_{mt} - R_f) + \beta_s \times \text{SMB} + \beta_v \times \text{HML} + \beta_{rmw} \times \text{RMW} + \beta_{cma} \times \text{CMA} + \epsilon_{it} \tag{2.2}$$

Where:

- R_{it}: Return on stock i at time t.
- R_f: Risk-free rate.
- R_{mt}: Market return at time t.
- SMB: Size factor (Small Minus Big), capturing the historical excess returns of small-caps over big-caps.
- HML: Value factor (High Minus Low), capturing the historical excess returns of value stocks over growth stocks.
- RMW: Profitability factor, capturing the difference in returns between companies with robust (high) and weak (low) operating profitability.
- CMA: Investment factor, capturing the difference in returns between companies with conservative and aggressive investments.
- α_i: Intercept, capturing stock i's abnormal return unexplained by the factors.
- ϵ_{it}: Error term for stock i at time t.

Using this model, the returns are controlled for diverse common market factors. The coefficients $\alpha_i, \beta_m, \beta_s, \beta_v, \beta_{rmw}, \beta_{cma}$ and ϵ_{it} are fit per ticker using linear regression. Table 2.4 presents the coefficients regressed for four tickers. ϵ_{it} is not included in the table as it is unique per observation.

Table 2.4: Coefficients for tickers AAPL, AMZN, MSFT, and GOOGL

Ticker	α_i	β_m,	β_s	β_v	β_{rmw}	β_{cma}
AAPL	-0.0052	0.0112	-0.0011	-0.0037	-0.0004	-0.0062
AMZN	-0.0053	0.0110	-0.0013	-0.0038	-0.0065	-0.0106
MSFT	-0.0057	0.0110	-0.0031	-0.0036	0.000059	-0.0030
GOOGL	-0.0042	0.0100	-0.0015	-0.0015	0.0002	-0.0076

2.4.3 Yearly Variation

To compute the correlation between the variation of both ESG metrics and log returns, the yearly difference for each is calculated and added to the dataset. For given year k, the difference is defined as:

$$r_{variation,k} = r_{annualized,k} - r_{annualized,k-1} \tag{2.3}$$

$$ESG_{variation,k} = ESG_k - ESG_{k-1} \tag{2.4}$$

The variations are then normalized using Standard Score:

$$\frac{X - \mu}{\sigma} \tag{2.5}$$

where: X is the data point, μ is the mean of either the returns or ESG metric, and σ the standard deviation of either the returns or ESG metric.

2.4.4 Correlation Analysis

With both the annualized and controlled returns in hand, we computed Pearson's correlation between returns and the various ESG metrics on a per-stock basis.

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \tag{2.6}$$

Where:

- r: Pearson's correlation coefficient, which measures the linear relationship between two datasets.
- x_i and y_i: Data values from the controlled or uncontrolled returns and an ESG metric being compared.
- $\bar{x}$ and $\bar{y}$: Mean values of controlled or uncontrolled returns and ESG metric being compared respectively.

A caveat that must be addressed when discussing correlation is the maximum attainable correlation for two given distributions. Pearson's correlation takes a value between $[-1, 1]$, but this is only true if the two random vectors X_1 and X_2 are of the same type [74]. The distribution model chosen for the annual log returns will be the normal distribution. This is an assumption commonly made within the Black-Scholes model [75]. A Kolmogorov–Smirnov goodness of fit test was performed on the ESG ratings comparing the underlying distribution of a sample to a given distribution. The highest scoring distribution ended up being Johnson's SU. To provide further interpretability to the coefficent, the maximum attainable

interval is calculated below. Starting with the upper bound, the first step is to calculate the covariance. Let $z \sim \mathcal{N}(0,1)$, then $X = \lambda \sinh(\frac{z-\gamma}{\delta}) + \xi$ and $Y = \sigma z + \mu$.

$$\operatorname{cov}(X,Y) = \mathrm{E}\left[(X - \mathrm{E}[X])(Y - \mathrm{E}[Y])\right]$$

$$= \operatorname{cov}(\lambda \sinh(\frac{z-\gamma}{\delta}) + \xi, \sigma z + \mu)$$

$$= \mathrm{E}(\lambda \sinh(\frac{z-\gamma}{\delta}) + \xi - \mathrm{E}(\lambda \sinh(\frac{z-\gamma}{\delta}) + \xi))(\sigma z + \mu - \mathrm{E}(\sigma z + \mu))$$

$$= \lambda\sigma \,\mathrm{E}(z \sinh(\frac{z-\gamma}{\delta}))$$

Let $z = \delta w$ where $\delta > 0$ and $\alpha = \gamma/\delta$ so $w \sim \mathcal{N}(0, 1/\delta^2)$ and $\mathrm{E}\left[z \sinh \frac{z-\gamma}{\delta}\right]$

$$= \int_{-\infty}^{\infty} w \exp(\pm w - \delta^2 w^2/2)\, dw$$

$$= \exp(\frac{1}{2\delta^2}) \int_{-\infty}^{\infty} w \exp(-\delta^2(w \mp 1/\delta^2)^2/2)\, dw = \pm \exp(\frac{1}{2\delta^2}) \frac{\sqrt{2\pi}}{\delta^3}$$

using standard Gaussian integral identities:

$$\mathrm{E}\left[z \sinh \frac{z-\gamma}{\delta}\right] = \exp(\frac{1}{2\delta^2}) \frac{\exp(-\alpha) + \exp(\alpha)}{2\delta} = \frac{\exp(\frac{1}{2\delta^2})}{\delta} \cosh \frac{\gamma}{\delta}.$$

Finally,

$$\operatorname{cov}(X,Y) = \frac{\lambda\sigma}{\delta} \exp(\frac{1}{2\delta^2}) \cosh \frac{\gamma}{\delta} \tag{2.7}$$

The process is now repeated for the lower bound with $z \sim \mathcal{N}(0,1)$, $X = \lambda \sinh(\frac{z-\gamma}{\delta}) + \xi$ and $Y = -\sigma z + \mu$.

$$\operatorname{cov}(X,Y) = -\frac{\lambda\sigma}{\delta} \exp(\frac{1}{2\delta^2}) \cosh \frac{\gamma}{\delta}. \tag{2.8}$$

We now have the variance for both distributions:

$$\operatorname{Var}(X) = \frac{\lambda^2}{2}(\exp(\delta^{-2}) - 1)\left(\exp(\delta^{-2}) \cosh\left(\frac{2\gamma}{\delta}\right) + 1\right) \tag{2.9}$$

$$\operatorname{Var}(Y) = \sigma^2 \tag{2.10}$$

Leading to $[\rho_{\min}, \rho_{\max}]$, with

$$\rho_{\min} = -\frac{\frac{\lambda\sigma}{\delta}\exp(\frac{1}{2\delta^2})\cosh\frac{\gamma}{\delta}}{\sqrt{(\frac{\lambda^2}{2}(\exp(\delta^{-2})-1)\left(\exp(\delta^{-2})\cosh\left(\frac{2\gamma}{\delta}\right)+1\right))(\sigma^2)}} \tag{2.11}$$

And

$$\rho_{\max} = \frac{\frac{\lambda\sigma}{\delta}\exp(\frac{1}{2\delta^2})\cosh\frac{\gamma}{\delta}}{\sqrt{(\frac{\lambda^2}{2}(\exp(\delta^{-2})-1)\left(\exp(\delta^{-2})\cosh\left(\frac{2\gamma}{\delta}\right)+1\right))(\sigma^2)}} \tag{2.12}$$

For a normal fit to the controlled returns of $(\mu, \sigma = 0.03, 0.06)$ and a Johnson SU's fit to the ESG metrics of $(\lambda, \gamma, \delta, \xi) = (1.39, -1.22, 7.91, -0.49)$ the calculations indicate $[\rho_{\min}, \rho_{\max}] = [-0.9998, 0.9998]$, analog to comparing 2 datasets with underlying normal distribution.

2.5 Results - Correlation between ESG Metrics and Returns

This section outlines the results of analyzing the correlation between investment returns and ESG scores. All the correlations are calculated for companies with a long enough rating history, and are averaged in their specific group. The discussion begins with a broad overview of these findings, then delves into specifics related to industry sectors and sustainability issues.

2.5.1 Preliminary Statistics and Distributions

Table 2.5 presents the descriptive statistics from the correlations between ESG ratings and uncontrolled or controlled annualized log returns. After controlling with the Fama-French 5 model, the means and standard deviations for score by score comparison has increased for every metric.

Table 2.5: Statistics for Correlation between (Un)controlled Returns and ESG Metrics (2006 cut-off)

	mean	std	min	25%	50%	75%	max
Uncontrolled							
ESG Score	0.03	0.22	-0.46	-0.11	0.04	0.18	0.70
Environmental Pillar Score	0.03	0.22	-0.50	-0.11	0.02	0.17	0.66
Governance Pillar Score	0.02	0.22	-0.50	-0.12	0.02	0.15	0.69
Social Pillar Score	0.02	0.23	-0.51	-0.16	0.02	0.16	0.72
Controlled							
ESG Score	0.36	0.23	-0.44	0.25	0.39	0.52	0.81
Environmental Pillar Score	0.37	0.26	-0.38	0.24	0.41	0.55	0.84
Governance Pillar Score	0.21	0.31	-0.57	0.02	0.25	0.45	0.79
Social Pillar Score	0.30	0.26	-0.58	0.15	0.30	0.47	0.79

As shown in Table 2.5, the uncontrolled annualized log returns present little to no correlation with the ESG metrics. The controlled annualized log returns present a much higher average correlation and higher standard deviation. The global ESG Score and Environmental Pillar Score appear to be the most correlated with the returns. As such, for the rest of the results, the correlations presented will be calculated using the returns controlled by Fama-French 5.

In Table 2.5, the most significant values are observed in the maximum correlations for ESG Score and Governance Pillar Score. The ESG Score records the highest maximum correlation in both uncontrolled (0.70) and controlled (0.81) returns, indicating situations where the ESG Score and market performance move together to a notable degree. Similarly, the Governance Pillar Score exhibits notable peak correlations (0.69 uncontrolled and 0.79 controlled), reflecting instances of concurrent movements between governance factors and return correlations. The Environmental Pillar Score also shows a particularly high maximum correlation in controlled returns (0.84).

Figure 2.3 represents the distribution of correlation between ESG metrics and annualized controlled log returns. The score distributions appear to be right-skewed across all metrics.

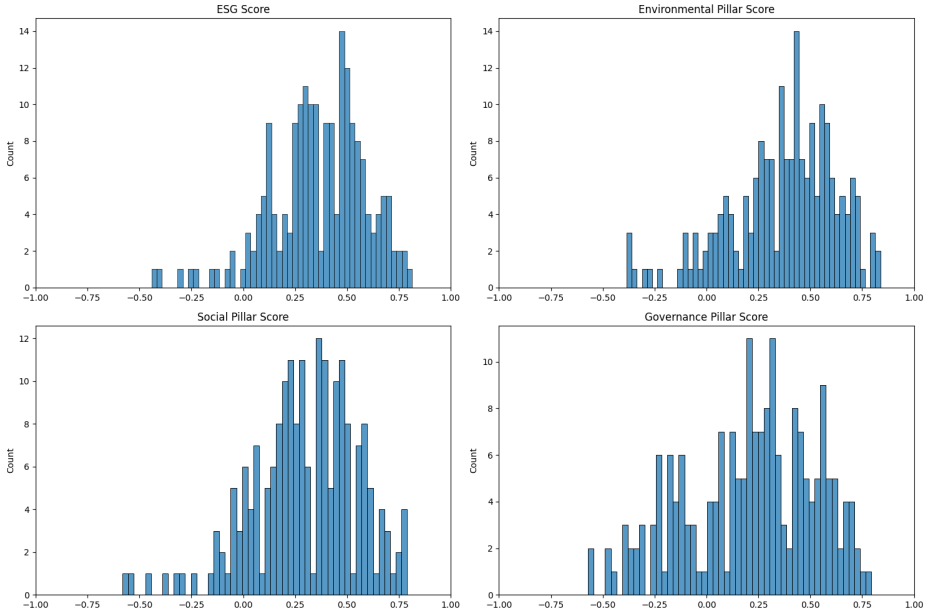


Figure 2.3: Distribution of Correlation between ESG Metrics and Controlled Annualized Log Return.

Table 2.6 presents the number of companies that have statistically significant correlation for a threshold at $p < 0.05$. The Environmental Pillar Score and ESG Score, being the most correlated, naturally have a higher number of statistically significant correlations. Since the effect is weaker in the Social Pillar Score and Governance Pillar Score, there are fewer companies with a statistically significant correlation.

Table 2.6: Statistical Significant Correlation between Controlled Annualized Returns and ESG Metrics

Metric	Correlation Significant Count	% of Companies	≤ 0 Count	≥ 0 Count
ESG Score	77	38.69%	0	77
Environmental Pillar Score	81	40.70%	0	81
Social Pillar Score	56	28.14%	3	53
Governance Pillar Score	53	26.63%	3	50

2.5.2 Correlation between ESG Metrics and Returns By Sectors

A deeper dive into the distribution of correlations within each sector illuminates the significant spread and variability. Heatmaps were used to highlight sectors and metrics with the most significant correlations. All subsequent heatmaps are on the same color ranging from $[-0.5, 0.5]$.

Figure 2.4 displays the heatmap of correlations between controlled annualized log returns and ESG metrics by sector. The returns have overall a weak to medium positive correlation with the ESG metrics. Financial Services exhibit a correlation of 0.46 with the ESG Score, indicating a notable association. The Healthcare sector shows a strong correlation as well, with a 0.41 correlation to the ESG Score, and is similarly aligned with the Environmental and Social Pillar Scores at 0.39 and 0.31 respectively. On the lower end, the Technology sector shows a distinctively weaker correlation, particularly with the Governance Pillar Score at 0.12. Consumer Cyclical stands out with a 0.40 correlation to the ESG Score and a 0.34 correlation to the Governance Pillar Score. Communication Services also demonstrate a substantial correlation with the ESG Score at 0.42.

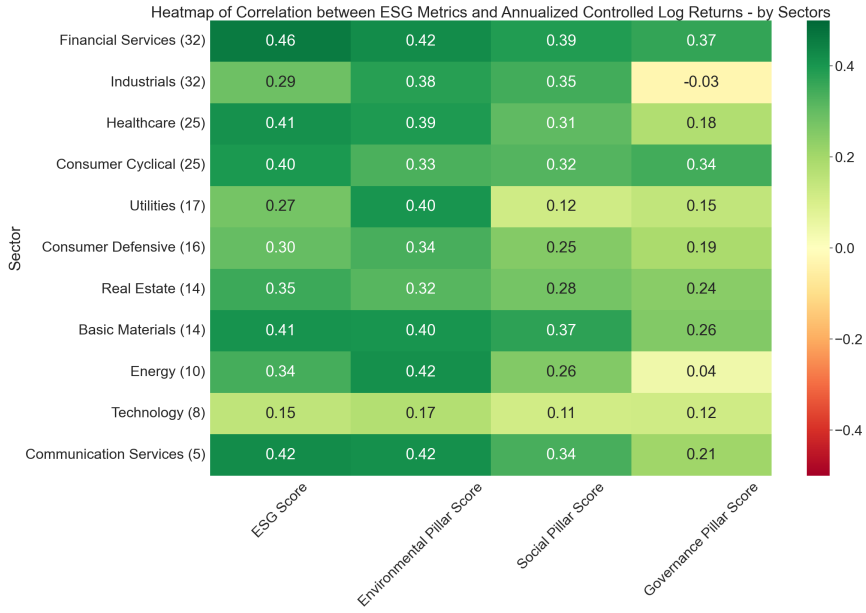


Figure 2.4: Heatmap of Correlations between Controlled Annualized Log Returns and ESG Metric by Sector.

2.5.3 Correlation between ESG Metrics and Returns by Materiality

In order to provide further granularity in this study, the 26 materiality issues isolated by SASB were used to group companies together. The correlation between returns and diverse ESG metrics is then computed individually for each company. These correlations are then averaged across all the companies presenting the corresponding SASB issue.

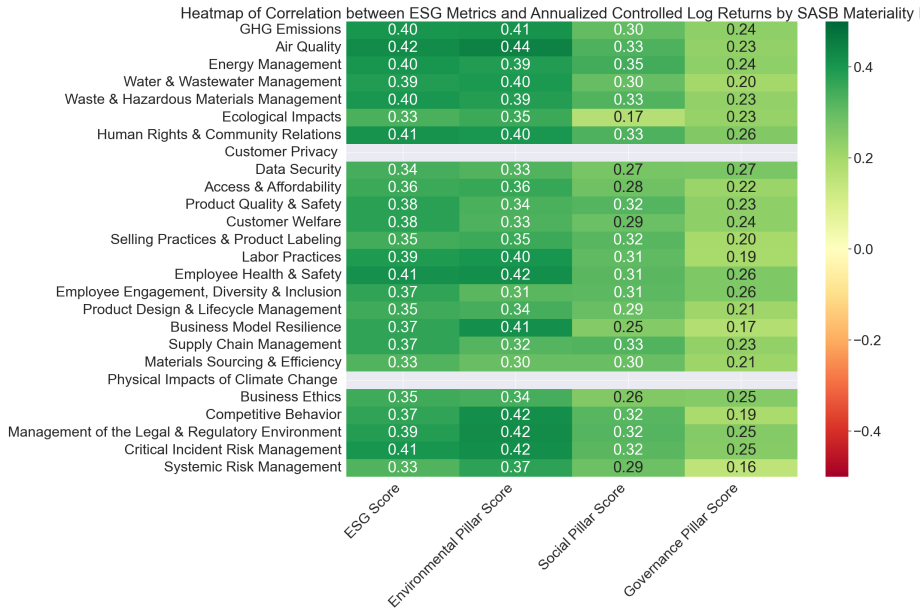


Figure 2.5: Heatmap of Correlations Between Controlled Annualized Log Return and ESG Metrics by Materiality.

Figure 2.5 displays the heatmap of correlations between controlled annualized log returns and ESG metrics by SASB materiality issues. Once controlled, the correlation appears to be much stronger on a score by score comparison. Overall, the ESG metrics maintain a generally positive correlation with the returns. Per scores, the correlations appear to be very close to each other, following the trend of Figure 2.4. Specific SASB issues such as GHG Emissions (0.40), Air Quality (0.42), and Human Rights & Community Relations (0.41) show a notable association with the ESG Score. These issues, along with Critical Incident Risk Management (0.41), are among the most correlated within the Environmental Pillar Score. This correlation

underscores the significance of these environmental and social issues in relation to financial performance.

In the context of the Social Pillar Score, Human Rights & Community Relations (0.40) and Employee Health & Safety (0.42) emerge as highly correlated issues, reflecting the importance of these aspects in corporate social responsibility. Additionally, the Governance Pillar Score reveals a strong correlation with issues such as Management of the Legal & Regulatory Environment (0.39) and Critical Incident Risk Management (0.42), which are integral to governance and risk oversight within organizations.

2.6 Results - Correlation between Variations of Returns and ESG Metrics

In this section, the correlation between the variations year per year of returns and ESG metrics is calculated over the integrality of companies. The variations are then normalized to avoid scale effect. Table 2.7 presents the global results, which shows an overall neutral correlation. The study is then refined with sectors and materiality issues. Finally, a time-lagged version of the correlations is proposed to explore potential delays between ESG initiatives and returns. When calculating the variations, every company out of the 491 that has an history ≥ 2 is used, bringing the number of companies considered to 263.

Table 2.7: Correlation between Variations of Returns and ESG Metrics

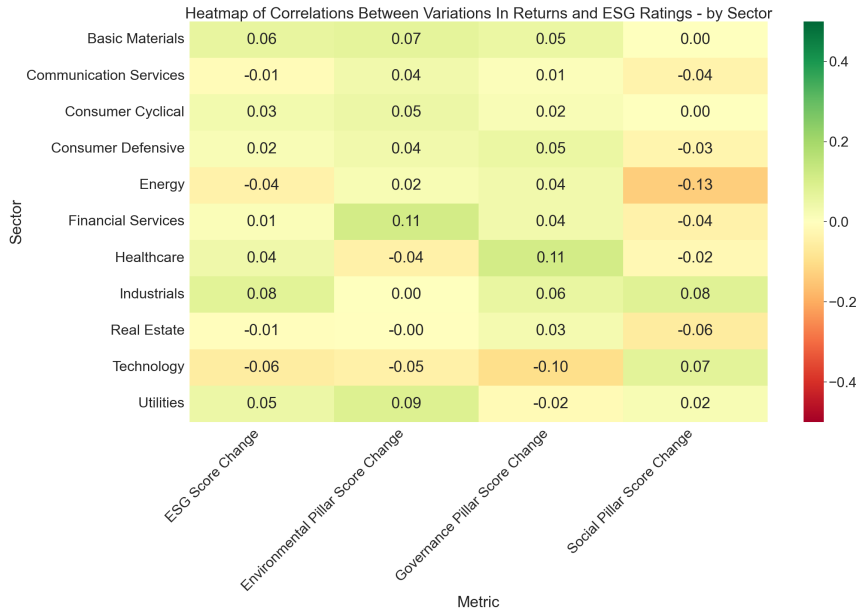
Metric	Correlation	P-Value
ESG Score Change	0.027	9.13e-10
Social Pillar Score Change	-0.005	2.34e-01
Governance Pillar Score Change	0.037	8.56e-17
Environmental Pillar Score Change	0.029	2.77-11

2.6.1 Correlation between Variations of Returns and ESG Metrics by Sectors

Figure 2.6 breaks down the correlation between the variation in returns and ESG metrics for companies in a given sector. The correlation between the variations appears to be neutral. The coefficients remain weak, with the Energy sector having a slightly higher negative correlation with the variation in the Social Pillar score. The Technology sector also has a slight negative correlation with the changes in Governance score. The Financial Services sector has the highest correlation between variations of returns and Environmental Pillar Score Change with 0.11. The

Healthcare sector has the highest correlation with the Governance Pillar Change. The strongest correlation on the heatmap is the Energy sector with the Social Pillar Score, standing at -0.13.

Figure 2.6: Heatmap of Correlations Between Variations of Returns and ESG Metrics by Sectors.



2.6.2 Correlation between Variations of Returns and ESG Metrics by Materiality

Figure 2.7 breaks down the correlation between the variation in returns and ESG metrics for companies for a given materiality issue. The correlation remains weak at this level too, indicating that there is no linear relationship identifiable between the variations in returns and variations in ESG metrics at the granularity studied in this article.

Figure 2.7: Heatmap of Correlations Between Variations of Returns and ESG Metrics by SASB Materiality Issues.

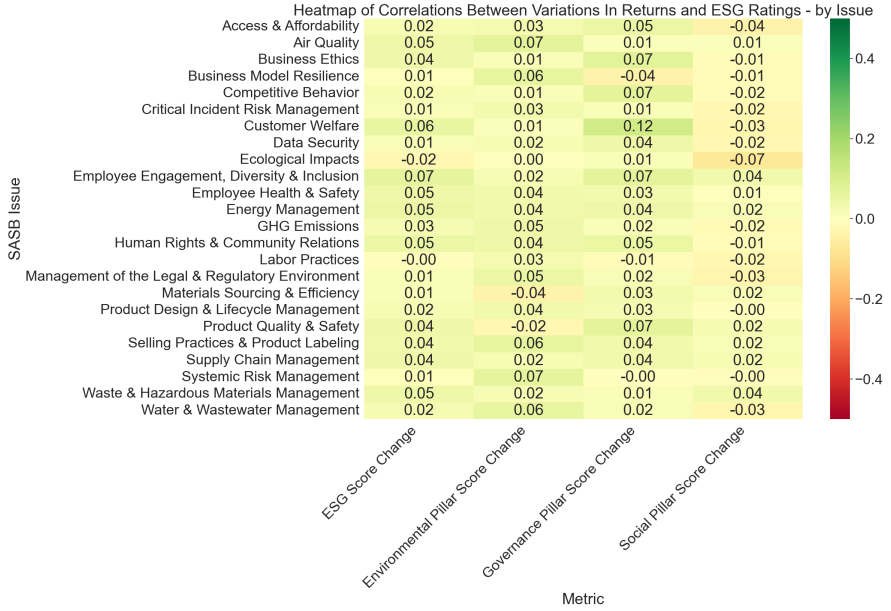


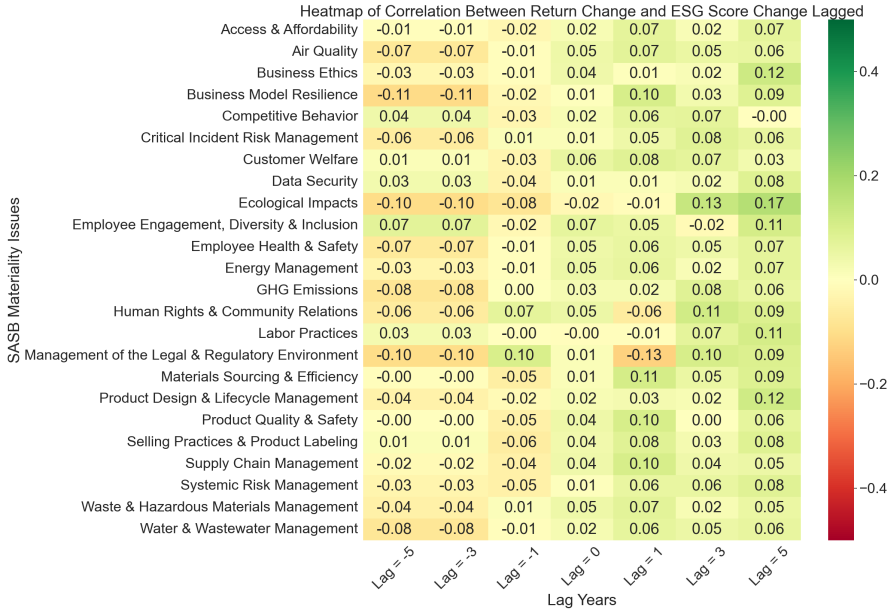
Figure 2.8 breaks down the correlation between the variation in returns and ESG score for companies for a given materiality issue with different time lags. Time lag for a given year t is defined as the correlation between $ESG_{(t+Lag)}$ with the returns R_t, with $Lag = -5, -3, -1, 0, 1, 3, 5$. Negative values of Lag effectively represent lagging the return variations as opposed to lagging the rating variations. When the ESG ratings variations are lagged by one, three or five years, the correlation with the changes in returns is neutral for most issues. Business Model Resilience, Supply Chain Management and Materials Sourcing & Efficiency appear to be the most correlated after one year. Ecological impact and Business Ethics are weakly positively correlated after 5 years. Management of the Legal & Regulatory Environment is the least correlated issue after one year.

When the returns are delayed, the correlation is weak to neutral with a one year delay. When delayed three years, there is a weak negative correlation for certain issues, including Supply Chain Management, Customer Welfare and Selling Practices & Product Labelling. At a five year delay, there is a weak negative correlation, with the most correlated being Management of the Legal & Regulatory Environment and Business Model Resilience. The other issues remain neutral.

The introduction of time-lagged analysis in these correlations reveals interest-

ing cross-correlation dynamics between ESG scores and company returns. Cross-correlation in timeseries analysis helps in understanding how two variables, like ESG scores and returns, are related and interact over different time lags. In this context, it suggests that ESG factors may not change at the same time as return but could have similar variations over extended periods. This cross-correlation analysis is particularly insightful for identifying which ESG factors change in a similar fashion to returns. For instance, issues like Labor Practices and Employee Engagement show stronger correlations at different time lags, suggesting that the effect of these ESG aspects on financial performance unfolds over a longer horizon.

Figure 2.8: Heatmap of Correlations Between Variations of Returns and ESG Score by SASB Materiality Issues With Time-Lag. At $Lag = -5$ variations in returns are effectively lagged by 5 years and at $Lag = 5$ variations in ESG Score are lagged by 5 years. $Lag = 0$ corresponds to no lag.



2.7 Discussion and analysis of the results

A first observation that can be drawn from the data is that the controlled returns seem to have a much higher correlation with the ESG ratings than the uncontrolled returns. One possible explanation could be that the market factors trimmed using Fama-French 5 acted as signal noise between the two variables.

Another notable finding is the absence of correlation between the variation of ESG ratings and variation of annualized log returns, regardless of controlling. This indicates that ESG ratings and annualized log returns tend to not increase or decrease together year per year. A plausible explanation could be a parallel with a company acting for growth or for profit. It was evidenced in previous study [76] that a company may experience higher growth with stagnating profitability and vice-versa depending on the business plan. Increasing sustainability or log returns in one's business requires concentrated efforts and could have an opportunity cost in the other area.

It was observed in this study that there appears to be no significant correlation at lag periods of one, three, or five years. This lack of correlation over time suggests that the impact of ESG factors on financial performance might not be immediate, but rather indirect or lagged. It raises critical questions about the temporal nature of ESG integration in financial analysis. One hypothesis could be that the benefits of high ESG ratings, such as enhanced reputation, better stakeholder engagement, and risk mitigation, may appear over a longer period. This delay could also be indicative of the market's slow adjustment, reflecting a lag in the incorporation of ESG considerations into investment decisions.

In the context of asset pricing, the relationship between ESG ratings and returns can be understood through the lens of systematic versus unsystematic risk. Systematic risk, which affects the entire market or a large segment of the market, can be paralleled by broad environmental concerns that impact multiple industries, while unsystematic risk is specific to individual companies or sectors. For instance, the stronger correlation of the Governance metric in the Industrials sector could suggest that governance practices are a significant source of unsystematic risk, affecting firm-specific returns and investment decisions.

Materiality becomes particularly relevant when considering the correlation of ESG metrics with sector performance. Material ESG factors vary by industry and can have a direct impact on a firm's risk profile and cost of capital. For example, environmental risks are highly material for the Industrials sector, suggesting that a high Environmental Pillar Score might display an attempt at mitigating those risks, potentially lowering the cost of equity for firms with strong environmental practices.

The regulatory environment is another factor to consider, as it can significantly affect company risk. Firms with high Governance Pillar Scores may be better prepared to face upcoming regulations. In sectors like Financial Services, for instance, the slight negative correlation with Governance Pillar Scores could reflect a market perception that less-regulated firms might experience short-term gains. However, this could expose investors to higher long-term risks, if regulatory scrutiny were to increase.

In the long-term investment horizon, High ESG ratings can signal a company's commitment to sustainability and resilience, which can be crucial for long-term

value creation. This is particularly relevant for metrics like Business Model Resilience, which shows a high correlation with returns, suggesting that companies prioritizing long-term sustainability initiatives may enjoy more stable returns over time. ESG ratings can aid in portfolio construction and diversification. By using ESG scores to identify companies that are potentially less exposed to ESG-related risks or are better managed, investors could reduce the risk profile of their portfolios and enhance their resilience to market shocks driven by ESG factors.

Lastly, when considering risk-adjusted returns, incorporating ESG ratings could provide a more comprehensive evaluation of an investment's performance. For example, a firm with high ESG ratings might demonstrate lower volatility and have a more favorable risk-adjusted return profile. This could make ESG ratings an integral component of risk management and asset pricing models, helping investors to identify opportunities for improved risk-adjusted returns within their portfolios.

2.8 Conclusion

This chapter evaluated the correlation between annualized log returns and ESG metrics among the companies in the S&P500. Annualized log returns were controlled using Fama-French 5 to remove market factors. The correlation between the variations of both data year by year was also computed. Our findings indicate a variable correlation between controlled log returns and ESG metrics. Sector-specific analysis revealed that company sector does influence the relationship between ESG metrics and returns. Finally, incorporating materiality issues enhances the explanatory power of the ESG-returns correlation by focusing on the most relevant ESG factors for each sector and therefore refining the correlation analysis. This highlights the importance of sector-specific and company specific ESG considerations in financial analysis, aligning with contemporary asset pricing models.

2.9 Limitations of Correlation-Based Methods and Motivation for Machine Learning Approaches

The correlation study provided valuable insights into the relationship between ESG ratings and financial performance, but the limitations of the correlation metrics to static, linear interactions are highly restrictive. These metrics are valuable in a higher level study but more complex interdependencies can be captured by more sophisticated models. The temporal granularity of the dataset can also obscure short-term dynamics that might take place in a rapidly evolving industry. This imbalance between financial data, which is available at high frequency, and ESG ratings that are typically updated annually, limits the use of correlation and forces further hypotheses to reach a result. The findings from the correlation study also revealed a rich diversity in strength and directions of correlations across sec-

tors and materiality dimensions. This result suggests that there is no static model that can be defined for all assets that would encompass the relationship between ESG ratings and financial performance.

In order to better understand this relationship, two machine learning frameworks were developed. The first framework proposes is the non-stationary inverted transformer, which implements the latest innovations in timeseries prediction. This supervised learning frameworks harnesses Time2Vec and de-stationary factors to model intricate relationships between features. The second framework is based on proximal policy optimization, a reinforcement learning algorithm. This framework uses a centralized mixture-of-experts approach to train a superagent in a suitable environment. Both of the frameworks were developed with interpretability in mind, with techniques specific to each of the models used to glean insights on how each feature influences the final predicition. These two frameworks, and their respective tools and benchmarks, are detailed in Part III.

Part III

Technical Framework: Machine Learning Models for ESG Analysis

Chapter 3

Timeseries Models: iTransformer and Variants

This chapter presents innovations on the inverted transformer, which is a state-of-the-art timeseries model, and proposes the non-stationary inverted transformer. This chapter also develops interpretability techniques inspired by previous work on the standard transformer architecture.

3.1 Introduction

Transformers have revolutionized the natural language processing with their self-attention mechanism and layered feed-forward networks. The self-attention mechanism enables the model to weigh the importance of each input token in relation to others, allowing it to capture long-range dependencies, while the feed-forward networks refine these relationships across layers. Their application to timeseries forecasting has been lagging behind, especially in the context of larger lookback windows.

This lag was highlighted by the surprising effectiveness of linear forecasters, which outperformed the previous attempts at adapting the transformer architecture to timeseries prediction [77]. With an affordable computation cost and a strong base of interpretability, the linear forecasters outperformed the modified transformer architectures, especially for long-term predictions.

The inverted transformer (iTransformer) architecture is among the state-of-the-art models in timeseries analysis [78]. By inverting the typical duties of the attention mechanism and feed-forward networks of the standard transformer architecture, the architecture is better equipped to forecast series with larger lookback windows. The iTransformer currently ranks first in the long-term forecasting task of the Timeseries Analysis benchmarks [79]. As the iTransformer does not intro-

duce any adaptation to the basic components, this architecture also benefits from the tools developed for the original Transformer architecture.

Despite the success of this models, effectively handling non-stationarity and computational complexity remains a challenge in multivariable timeseries forecasting [80]. The integration of de-stationary mechanisms, like those introduced in the NSTransformer to model inter-tokens relationships, into architectures such as the iTransformer offers a potential solution to these issues. By learning scaling and shifting factors for inter-variable relationships, models can better adapt to non-stationary behaviors in the data.

Moreover, computational efficiency is crucial when dealing with high-dimensional timeseries data. Introducing sparsity into attention mechanisms, such as using top-k sparse attention, can significantly reduce computational complexity from $O(N^2)$ to $O(Nk)$, where N is the number of variables and k is a small constant. This allows the model to focus on the most relevant inter-variable relationships without incurring prohibitive computational costs.

This paper leverages the similarity between the original Transformer architecture and the iTransformer to adapt and extend Transformer-specific interpretability methods. Specifically, we explore the application of Chefer's generic method for transformer interpretability [81]. This paper reformulates the original method for a regression problem and a continuous output, and adapts it to the inverted transformer architecture. The result is a continuous relevance map highlighting critical variables that are influential in the predictive power of the model.

The core research questions in this chapter are the following: can de-stationary attention be extended to the iTransformer architecture? Can Time2Vec embedding improve the performance of the iTransformer? What improvements in forecasting performance and efficiency can be achieved through this integration?

The chapter is structured as follow: Section 3.2 includes a contextualization and a review of the existing literature. Section 3.3 introduces the mathematical definition of the method. Section 3.6 displays the results of the forecasting using the Non-Stationary inverted Transformer (NSiTransformer). Section 3.7 proposes an analysis of several components and mechanisms of the model. Section 3.8 and 3.9 are the conclusion and future work of the chapter.

3.2 Literature Review

The transformer architecture [29] has become a cornerstone of deep learning, particularly in natural language processing tasks. The self-attention mechanism allows the model to weight the importance of different tokens in a sequence relative to one another. This architecture is the foundation behind most of the mainstream models, such as ChatGPT [82], Claude [83], Mistral [84], and Llama [85].

The surge in research has provided fast improvement in parallelization [86] and diverse optimizations [87].

Various modification paradigms have been proposed to improve the accuracy of Transformer-based forecasters. Autoformer [88] and Informer [89] propose to replace the attention component respectively with an autocorrelation and sparse attention mechanisms. Crossformer [90] focuses on modeling the cross-time and cross-dimension dependency using a two-stage attention and modified hierarchical encoder-decoder architecture. Finally, PatchTST [91] and Non-Stationary Transformer (NSTransformer) [92] focused on the processing of timeseries using patching and stationarization respectively.

The inverted transformer (iTransformer) introduces no modification to the original Transformer components [29]. By inverting the duties of the attention mechanism and the feed-forward network, this architecture aims to reduce performance degradation and computation explosion in larger lookback windows. This recent model has been succesfully used to predict the useful life of Lithium-Ion batteries [93], earthquake detection [94] and predict sea surface temperature [95]

Interpretability methods for Transformers are sparse in the literature. Exploiting the raw attention weights to draw attention maps has been criticized for a limited contribution to the interpretability [96] [97]. The ConceptTransformer [98] proposed to modify the architecture for better explainability. Vision Transformer is the most prolific source of interpretability attempts, with neural tree decoder [99] and interpretability-aware training objectives [100]. The method used to get insights in this study is centered around an adaptation of a general interpretability technique to iTransformer [42]. This technique consists of building a relevancy maps of the different tokens using the gradients of the feed forward networks.

In this chapter, we build upon these ideas by integrating the de-stationary attention mechanism and variable projector from the NSTransformer into the iTransformer framework. We further enhance computational efficiency by incorporating a sparse attention mechanism that computes scaling and shifting factors only for the top-k most relevant variable pairs. This approach aims to capture non-stationary inter-variable relationships more effectively while maintaining scalability for large-scale timeseries forecasting tasks.

3.3 Non-Stationary iTransformer with Time2Vec Embedding

3.3.1 Preliminaries

iTransformer: Given historical observations $X = \{x_1, x_2, \dots, x_T\} \in \mathbb{R}^{T \times N}$ with T time steps and N variables, the goal is to predict the future S time steps $Y =$

$\{x_{T+1}, x_{T+2}, \dots, x_{T+S}\} \in \mathbb{R}^{S \times N}$. In the iTransformer, each timeseries of a variable is embedded into variable tokens, which are then utilized by the attention mechanism to capture multivariable correlations. The feed-forward network is applied to each variable token to learn nonlinear representations, and the final output is generated by projecting these representations back to the timeseries domain.

The process can be formulated as follows:

$$h_0^n = \text{Embedding}(X_{:,n}) \tag{3.1}$$

$$H^{l+1} = \text{TrmBlock}(H^l), \quad l = 0, \dots, L-1 \tag{3.2}$$

$$\hat{Y}_{:,n} = \text{Projection}(h_L^n) \tag{3.3}$$

where $H = \{h_1, h_2, \dots, h_N\} \in \mathbb{R}^{N \times D}$ contains N embedded tokens of dimension D. The functions Embedding $: \mathbb{R}^T \to \mathbb{R}^D$ and Projection $: \mathbb{R}^D \to \mathbb{R}^S$ are implemented by multi-layer perceptrons (MLP). The self-attention and feed-forward network operations in each Transformer block (TrmBlock) enable the model to learn complex dependencies across variables and time steps.

NSTransformer: addresses the challenges posed by non-stationary timeseries data, where statistical properties such as mean and variance change over time. It introduces a de-stationary attention mechanism that adjusts the attention computations to account for these changes, enhancing the model's ability to capture evolving patterns in the data.

In the NSTransformer, per-time-step scaling (τ_t) and shifting (δ_t) factors are learned to adjust the input:

$$\tilde{\mathbf{x}}_t = \tau_t \odot \mathbf{x}_t + \delta_t, \tag{3.4}$$

where $\mathbf{x}_t$ is the input at time step t, $\odot$ the Hadarmard product (element-wise multiplication) and $\tilde{\mathbf{x}}_t$ is the adjusted input.

NSiTransformer: The inverted architecture combined with the de-stationary factors proposed lead us to the name of Non-stationary Inverted Transformer, or NSiTransformer.

3.3.2 Components

This section details the components of the NSiTransformer.

Overall Model Architecture: Figure 3.1 presents the architecture of the model. The overall process of the proposed model can be summarized as follows:

1. **Normalization**: Normalize each variable timeseries using its mean and standard deviation.

2. **Embedding**:

- Concatenate the normalized variable timeseries with the Time2Vec embeddings at each time step.
- Apply a linear transformation to project the concatenated vectors into the model dimension D.

3. **Attention with De-Stationary Factors**:
   - Compute preliminary attention scores between variable tokens.
   - Select the top-k variable pairs for each variable based on these scores.
   - Compute scaling and shifting factors $\tau_{i,j}$ and $\delta_{i,j}$ using the variable projector network for the selected pair of variables (i, j) for each of the top-k pairs.
   - Adjust the attention scores using $\tau_{i,j}$ and $\delta_{i,j}$.
   - Apply the attention mechanism to update variable tokens.
4. **Feed-Forward Network**: Apply position-wise feed-forward networks to the updated variable tokens.
5. **Projection**: Project back from the embedded dimension back to the projection length.
6. **De-Normalization**: Reintroduce the original scale and mean to the variable tokens to obtain the final output.

Normalization: To stabilize training and improve convergence, the input timeseries is normalized before being fed into the model. For each variable i, the mean μ_i and standard deviation σ_i are computed over the sequence length T:

$$\mu_i = \frac{1}{T}\sum_{t=1}^{T} X_{t,i}, \quad \sigma_i^2 = \frac{1}{T}\sum_{t=1}^{T}(X_{t,i} - \mu_i)^2 + \epsilon, \tag{3.5}$$

where $X_{t,i}$ is the $i-th$ variable at time t, and ϵ is a small constant to prevent division by zero. The normalized input $\tilde{\mathbf{X}}_{:,i}$ is then obtained by:

$$\tilde{\mathbf{X}}_{:,i} = \frac{\mathbf{X}_{:,i} - \mu_i}{\sigma_i}. \tag{3.6}$$

This normalization ensures that each variable has zero mean and unit variance, reducing the impact of scale differences between variables.

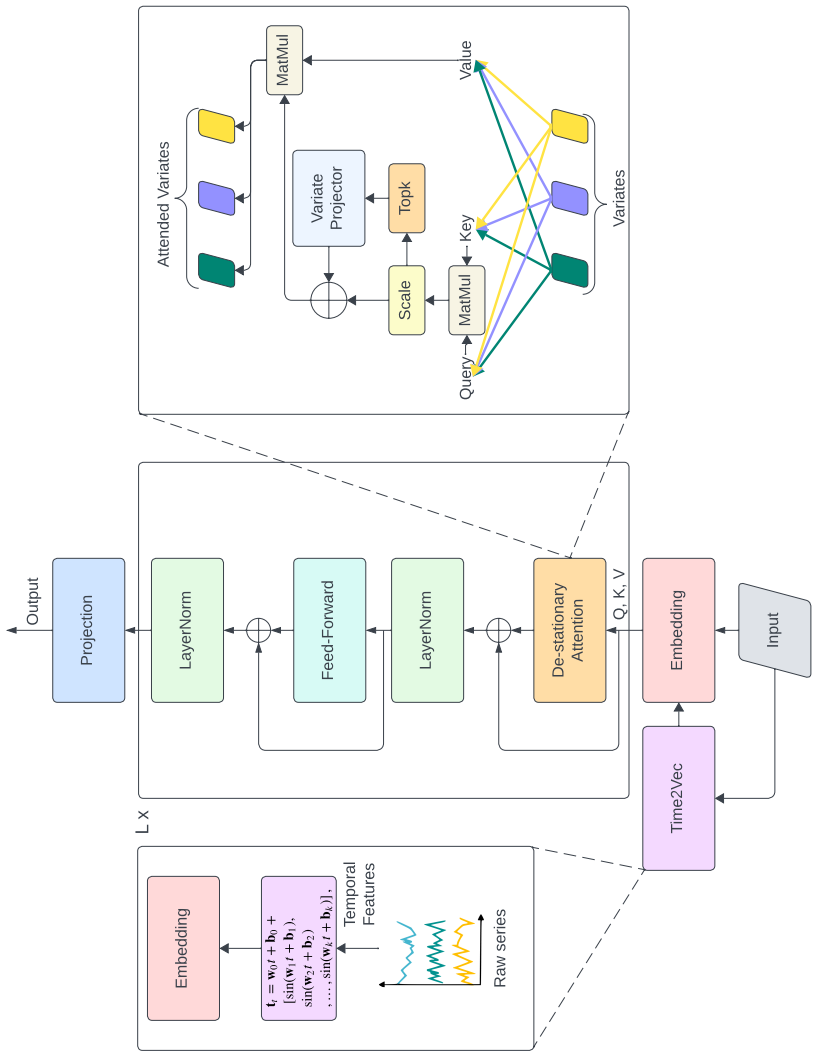


Figure 3.1: Architecture of the proposed model. MatMul is the matrix multiplication. The temporal features are embedded using Time2vec, and the series is embedded. De-stationary attention is then applied. We then apply Layer Normalization and the Feed-Forward Network. The result is then projected to the prediction length.

Embedding: Time2Vec [101] extends the concept of positional encoding by learning a vector representation of time that captures both linear and periodic patterns.

Given an hyperparameter d_{time}, for each time step t, the Time2Vec embedding $\mathbf{t}_t \in \mathbb{R}^{d_{\text{time}}}$ is defined as:

$$\mathbf{t}_t = \mathbf{w}_0 t + \mathbf{b}_0 + [\sin(\mathbf{w}_1 t + \mathbf{b}_1), \sin(\mathbf{w}_2 t + \mathbf{b}_2), \dots, \sin(\mathbf{w}_k t + \mathbf{b}_k)] \tag{3.7}$$

where $\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_k \in \mathbb{R}$ and $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_k \in \mathbb{R}$ are learnable parameters, and $k = d_{\text{time}} - 1$ is the number of sine components.

The Time2Vec embedding captures both linear trends and periodic patterns, enhancing the model's ability to learn temporal dynamics.

Modeling Inter-variable Non-Stationary Relationships using τ and δ: To capture non-stationary relationships between variables, we introduce learned scaling ($\tau_{i,j}$) and shifting ($\delta_{i,j}$) factors for each pair of variables (i, j). These factors adjust the attention scores between variable tokens, allowing the model to adapt to changes in inter-variable relationships over time.

Attention Mechanism with De-Stationary Factors: The attention scores between variable tokens are computed as:

$$\text{scores}_{i,j} = \frac{\mathbf{q}_i \mathbf{k}_j^\top}{\sqrt{d_k}} \times \tau_{i,j} + \delta_{i,j}, \tag{3.8}$$

where $\mathbf{q}_i, \mathbf{k}_j \in \mathbb{R}^{d_k}$ are the query and key vectors for variable tokens i and j, respectively, and d_k is the dimension of the key vectors.

Variable Projector Network The scaling and shifting factors $\tau_{i,j}$ and $\delta_{i,j}$ are computed using a single linear transformation of the concatenated embeddings of variable tokens i and j:

$$[\tau_{i,j}, \delta_{i,j}] = \mathbf{W}[\mathbf{h}_i; \mathbf{h}_j] + \mathbf{b}, \tag{3.9}$$

where $\mathbf{h}_i, \mathbf{h}_j \in \mathbb{R}^D$ are the embeddings of variable tokens i and j, $[\mathbf{h}_i; \mathbf{h}_j] \in \mathbb{R}^{2D}$ denotes their concatenation, $\mathbf{W} \in \mathbb{R}^{2 \times 2D}$ is a learnable weight matrix, and $\mathbf{b} \in \mathbb{R}^2$ is a bias vector.

Sparse Computation with Top-k Selection: To reduce computational complexity from $O(N^2)$ to $O(Nk)$, we compute $\tau_{i,j}$ and $\delta_{i,j}$ only for the top-k most relevant variable pairs for each variable. The top-k variables are selected based on the preliminary attention scores:

$$\text{scores}_{i,j}^{\text{pre}} = \sum_{h=1}^{H} \mathbf{q}_i^{(h)} \left(\mathbf{k}_j^{(h)}\right)^\top, \tag{3.10}$$

where H is the number of attention heads, and $\mathbf{q}_i^{(h)}, \mathbf{k}_j^{(h)}$ are the query and key vectors for head h. For each variable i, we select the indices of the top-k variables j with the highest scores $\text{score}_{i,j}^{\text{pre}}$.

De-Normalization: After the attention mechanism and updates to the variable tokens, we reintroduce the original scale and mean to obtain the final output. This de-normalization step ensures that the model's predictions are in the same scale as the original data and are interpretable.

The de-normalization is performed as:

$$\mathbf{h}_i^{\text{final}} = \mathbf{h}_i' \odot \sigma_i + \mu_i, \tag{3.11}$$

where $\mathbf{h}_i' \in \mathbb{R}^D$ is the updated variable token after the attention and feed-forward layers, and $\odot$ denotes element-wise multiplication.

The following pseudo-code outlines the implementation of the NSiTransformer:

Algorithm 1 Non-Stationary Inverted Transformer (NSiTransformer)

Require: Normalized data $\mathbf{X}$, Time2Vec embeddings $\{\mathbf{t}_t\}$, Model parameters
Ensure: Forecasted values $\hat{\mathbf{Y}}$

Embedding Layer:
for $t = 1$ to T **do**
  $\mathbf{z}_t \leftarrow [\tilde{\mathbf{X}}_{t,:}, \mathbf{t}_t]$
  $\mathbf{h}_t \leftarrow \mathbf{W}_{\text{embed}}\mathbf{z}_t + \mathbf{b}_{\text{embed}}$
Transpose embeddings to get variable tokens $\{\mathbf{h}_i^0\}_{i=1}^N$
for layer $l = 1$ to L **do**
  for each variable i **do**
    Compute queries $\mathbf{q}_i$, keys $\mathbf{k}_i$, values $\mathbf{v}_i$
    Compute preliminary scores $\text{score}_{i,:}^{\text{pre}}$
    Select top-k indices S_i
    for $j \in S_i$ **do**
      $[\tau_{i,j}, \delta_{i,j}] \leftarrow f_{\text{proj}}(\mathbf{h}_i^{l-1}, \mathbf{h}_j^{l-1})$
      Adjusted score: $\text{score}_{i,j} \leftarrow \dfrac{\mathbf{q}_i^\top \mathbf{k}_j}{\sqrt{d_k}}\tau_{i,j} + \delta_{i,j}$
    Attention weights: $\alpha_{i,:} \leftarrow \text{softmax}(\text{score}_{i,S_i})$
    Update embedding: $\mathbf{h}_i^l \leftarrow \sum_{j \in S_i} \alpha_{i,j}\mathbf{v}_j$
  Apply FFN to $\{\mathbf{h}_i^l\}$
for each variable i **do**
  De-normalize: $\hat{\mathbf{Y}}_{:,i} \leftarrow \mathbf{h}_i^L \odot \sigma_i + \mu_i$
return $\hat{\mathbf{Y}}$

3.4 Interpretability in Inverted Transformers

3.4.1 Relevancy Initialization

We initialize the relevancy maps as follows:

$$R_{vd} = I_{v \times d} \tag{3.12}$$
$$R_{vt} = 0_{v \times t} \tag{3.13}$$
$$R_{vv} = Concat(R_{vd}, R_{vt}) \tag{3.14}$$

Here, R_{dv} represents the self-attention relevancy map for variate tokens, where d is the number of variate tokens. R_{vt} represents the interaction between all tokens and time-related tokens, where t is the number of time tokens and $v = d + t$. The identity matrix $I_{v \times d}$ ensures that each variate token initially has a relevance score focused on itself, while the zero matrix $0_{v \times d}$ indicates no initial interaction between variate and time tokens. R_{vv} is the concatenation of R_{vd} and R_{vt} alongside the dimension 0 (v)

3.4.2 Self-Attention Relevancy Update

In the iTransformer, self-attention is applied to variate tokens. The relevancy maps are updated using attention weights:

$$R_{vv} \leftarrow R_{vv} + \overline{A}_{vv} \odot R_{vv} \tag{3.15}$$

where $\overline{A}_{vv}$ represents the averaged attention weights for variate tokens, computed as:

$$\overline{A}_{vv} = \frac{1}{H} \sum_{h=1}^{H} \text{ReLU}\left(\nabla A_{vv}^{h} \odot A_{vv}^{h}\right) \tag{3.16}$$

With

$$\text{ReLU}(x) = x^{+} = \max(0, x) = \frac{x + |x|}{2} = \begin{cases} x & \text{if } x > 0, \\ 0 & x \leq 0 \end{cases} \tag{3.17}$$

Here, H denotes the number of attention heads, ∇A_{vv}^{h} represents the gradients of the attention weights, and $\odot$ denotes element-wise multiplication. The use of ReLU ensures that only positive contributions are considered, by zeroing out inhibitory effects to highlight the relationship between variables.

3.4.3 Feed-Forward Network Relevancy Update

The feed-forward network relevancy update is applied independently to each variate token:

$$R_{\text{ff}} = Expand(\text{ReLU}(\nabla y \odot y)) \tag{3.18}$$

where y represents the outputs of the feed-forward network, and ∇y denotes the gradients of these outputs with respect to the loss.

$$R_{vv} \leftarrow R_{vv} + R_{\text{ff}} \tag{3.19}$$

Equation 3.19 refines the relevancy scores by applying learned transformations, emphasizing the most influential tokens.

The following pseudo-code outlines the implementation of the adapted Chefer method for the iTransformer:

Algorithm 2 Relevancy Mapping for iTransformer

Input: Number of variate tokens v, number of time tokens t
Output: Relevancy maps R_{output}
Initialize $R_{vd} \leftarrow I_{v \times v}$ ▷ Self-attention for variate tokens
Initialize $R_{vt} \leftarrow 0_{v \times t}$ ▷ Interaction of variate tokens with time-related tokens, if applicable
Initialize $R_{vv} \leftarrow Concat(R_{vd}, R_{vt})$ ▷ Concatenation alongside dimension 0 (v)
for each layer in iTransformer **do**
    $A_{vv} \leftarrow$ layer.variate_attention_map()
    $\overline{A_{vv}} \leftarrow \frac{1}{H} \sum_{h=1}^{H} \text{ReLU}\left(\nabla A_{vv}^{h} \odot A_{vv}^{h}\right)$ ▷ Averaged across heads
    $R_{vv} \leftarrow R_{vv} + \overline{A_{vv}} \cdot R_{vv}$
    $R_{vv} \leftarrow R_{vv} + Expand(\text{ReLU}(\nabla y \odot y))$ ▷ Feed-Forward update
Return R_{output}

3.4.4 Visualization and Analysis

The relevancy maps for each variate token can be visualized similarly to attention maps. For interpretability, we focus on how each variate token contributes to the final regression output. All relevance scores are normalized between $[-1, 1]$ for readability. Each cell at position (i, j) shows the relevance of token i (on the y-axis) to token j (on the x-axis).

3.5 Token Analysis

In this section, we provide a detailed analysis of the embedding process used by the iTransformer architecture. This process is crucial for transforming the input timeseries data into a set of tokens that the model can effectively process to capture important multivariate correlations.

3.5.1 Mathematical Representation of the Embedding Process

The iTransformer architecture inverts the traditional roles of the attention mechanism and the feed-forward network found in conventional transformers. Instead of using tokens to represent time steps, the iTransformer generates tokens that correspond to the variates (or features) of the timeseries. This inversion allows the attention mechanism to focus on capturing correlations between different variates, which is essential for multivariate timeseries forecasting.

Let $\mathbf{X} \in \mathbb{R}^{B \times T \times V}$ represent the input timeseries data, where B is the batch size, T is the sequence length, and V is the number of variates (features). Each element $\mathbf{X}_{b,t,v}$ corresponds to the value of variate v at time step t in batch b.

Additionally, let $\mathbf{M} \in \mathbb{R}^{B \times T \times F}$ represent the temporal features, where F is the number of temporal features (e.g., time of day, day of the week, seasonality). Each element $\mathbf{M}_{b,t,f}$ corresponds to the temporal feature f at time step t in batch b.

The embedding process involves the following steps:

3.5.1.1 Concatenation of Features and Temporal Information

The variates and temporal features are first concatenated along the feature dimension:

$$\mathbf{Z} = [\mathbf{X}, \mathbf{M}] \in \mathbb{R}^{B \times T \times (V+F)}$$

Here, $\mathbf{Z} \in \mathbb{R}^{B \times T \times (V+F)}$ is the concatenated representation where each time step t in the sequence now has $V + F$ dimensions, accounting for both the variates and temporal features.

3.5.1.2 Linear Transformation to Token Space

Next, a linear transformation is applied to map each combined feature vector at time step t to a d-dimensional token space:

$$\mathbf{E}_t = \mathbf{W}_e \mathbf{Z}_t + \mathbf{b}_e, \quad \mathbf{E} \in \mathbb{R}^{B \times T \times d}$$

where $\mathbf{W}_e \in \mathbb{R}^{(V+F) \times d}$ is the weight matrix, $\mathbf{b}_e \in \mathbb{R}^d$ is the bias vector, and $\mathbf{E}_t \in \mathbb{R}^{B \times d}$ is the embedded token for time step t. This transformation projects

each time step's concatenated features into a d-dimensional space, producing T tokens for each sequence.

3.5.1.3 Generation of Variate Tokens

In the iTransformer, instead of focusing on individual time steps, we invert the focus to variates. The attention mechanism processes the sequence of tokens $\mathbf{E}$ to capture the relationships between different variates across the entire timeseries. Each variate token $\mathbf{e}_v$ can be represented as:

$$\mathbf{e}_v = \sum_{t=1}^{T} \alpha_{v,t} \mathbf{E}_t, \quad v = 1, \ldots, (V+F)$$

where $\alpha_{v,t}$ are the attention weights that determine the contribution of each time step t to the variate token $\mathbf{e}_v$. The resulting tokens $\mathbf{e}_v$ encapsulate the interactions between different variates, as well as their temporal contexts.

3.6 Benchmarks

Benchmarking the NSiTransformer against other state-of-the-art models is essential to assess the effectiveness of our method.

Models: We harness the Time-Series-Library [102] and propose seven of the best performing models as our benchmark: iTransformer [78], PatchTST [91], Crossformer [90], TimesNet [103], DLinear [77], NSTransformer (NST) [92].

Datasets: We use the ETT, Weather, ECL and Traffic datasets included in Autoformer for long term forecasting. Table 3.1 details the features of the datasets.

Table 3.1: Detailed dataset descriptions.

Task	Dataset	Dim	Dataset Size (Train, Val, Test)	Frequency
Forecasting (long-term)	ETT	7	(8545, 2881, 2881)	15min
	Weather	21	(36792, 5271, 10540)	10min
	ECL	321	(18317, 2633, 5261)	Hourly
	Traffic	862	(12185, 1757, 3509)	Hourly

Forecasting: Table 3.2 presents the full results in long-term forecasting of the NSiTransformer against the six benchmark models. NSiTransformer achieves state-of-the-art or near state-of-the-art performance in all 4 benchmark datasets.

Table 3.2: Full results for the long-term forecasting task. The input sequence length is set to 96 for all baselines, and T is the prediction length. Avg is the average result of all four prediction lengths. MSE stands for Mean Squared Error and MAE for Mean Absolute Error.

Models	T	Our		iTransformer		PatchTST		Crossformer		TimesNet		DLinear		NST	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETT	96	**0.292**	**0.347**	0.297	0.349	0.302	0.348	0.745	0.584	0.340	0.374	0.333	0.387	0.476	0.458
	192	**0.374**	**0.400**	0.380	0.400	0.388	0.400	0.877	0.656	0.402	0.414	0.477	0.476	0.512	0.493
	336	**0.426**	**0.432**	0.428	**0.432**	0.426	0.433	1.043	0.731	0.452	0.452	0.594	0.541	0.552	0.551
	720	**0.420**	**0.440**	0.427	0.445	0.431	0.446	1.104	0.763	0.462	0.468	0.831	0.657	0.562	0.560
	Avg	**0.378**	**0.404**	0.383	0.407	0.387	0.407	0.942	0.684	0.414	0.427	0.559	0.515	0.526	0.516
ECL	96	**0.147**	**0.238**	0.148	0.240	0.195	0.285	0.219	0.314	0.168	0.272	0.197	0.282	0.169	0.273
	192	**0.162**	**0.253**	**0.162**	**0.253**	0.199	0.289	0.231	0.322	0.184	0.289	0.196	0.285	0.182	0.286
	336	**0.175**	**0.268**	0.178	0.269	0.215	0.305	0.246	0.337	0.198	0.300	0.209	0.301	0.200	0.304
	720	**0.208**	**0.298**	0.225	0.317	0.256	0.337	0.280	0.363	0.220	0.320	0.245	0.333	0.222	0.321
	Avg	**0.173**	**0.264**	0.178	0.270	0.216	0.304	0.244	0.334	0.192	0.295	0.212	0.300	0.193	0.296
Traffic	96	**0.393**	**0.267**	0.395	0.268	0.544	0.359	0.522	0.290	0.593	0.321	0.650	0.396	0.612	0.338
	192	**0.414**	**0.275**	0.417	0.276	0.540	0.354	0.530	0.293	0.617	0.336	0.598	0.370	0.613	0.340
	336	**0.428**	**0.281**	0.433	0.283	0.551	0.358	0.558	0.305	0.629	0.336	0.605	0.373	0.618	0.328
	720	**0.459**	**0.300**	0.467	0.302	0.586	0.375	0.589	0.328	0.640	0.350	0.645	0.394	0.653	0.355
	Avg	**0.423**	**0.280**	0.428	0.282	0.555	0.362	0.550	0.304	0.620	0.336	0.625	0.383	0.624	0.340
Weather	96	**0.171**	**0.211**	0.174	0.214	0.177	0.218	0.158	0.230	0.172	0.220	0.196	0.255	0.173	0.223
	192	0.224	0.256	**0.221**	**0.254**	0.225	0.259	0.206	0.277	0.219	0.261	0.237	0.296	0.245	0.285
	336	0.281	0.297	**0.278**	**0.296**	0.278	0.297	0.272	0.335	0.280	0.306	0.283	0.335	0.321	0.338
	720	**0.356**	**0.348**	0.358	0.349	0.354	0.348	0.398	0.418	0.365	0.359	0.345	0.381	0.414	0.410
	Avg	**0.258**	**0.278**	**0.258**	0.279	0.259	0.281	0.259	0.315	0.259	0.287	0.265	0.317	0.288	0.314

3.6.1 Correspondence Between Tokens and Variates

The dataset used is the Electricity Transformer Temperature (ETT2) dataset, which is widely considered as standard and the benchmark in timeseries forecasting. The embedding process results in a set of tokens, each corresponding to a specific variate or temporal feature. The iTransformer, and its modified counterpart, generate a total of $V + F = 11$ tokens in the experiments conducted, where:

- **Tokens 0-6**: Represent the original variates of the timeseries (e.g., HUFL, HULL, MUFL, MULL, LUFL, LULL, OT).
- **Tokens 7-10**: Correspond to the additional temporal features (e.g., hour of the day, day of week, day of month, day of year).

Token	Feature	Description	Units
0	`HUFL`	High UseFul Load	kW
1	`HULL`	High UseLess Load	kW
2	`MUFL`	Middle UseFul Load	kW
3	`MULL`	Middle UseLess Load	kW
4	`LUFL`	Low UseFull Load	kW
5	`LULL`	Low UseLess Load	kW
6	`OT`	Oil temperature of the transformer	°C
7	`hourDay`	hour of the day	-
8	`dayWeek`	day of the week	-
9	`dayMonth`	day of the month	-
10	`dayYear`	day of the year	-

Table 3.3: Tokenized features of the ETT2 dataset.

Table 3.3 presents the correspondence between tokens and features in the experiments. The temporal features are transformed as value between $[-0.5, 0.5]$ before being embedded into tokens 7 to 10. The direct embedding of a variate as a token allows for a much better understanding of the relationship between features. For instance, token 6 is expected to have a high relevancy, as this token represents the past values of the predicted variate, `OT`.

3.7 Analysis

We propose supplemental considerations regarding efficiency of the different modules, computation, hyperparameters sensitivity and interpretability.

3.7.1 Ablation

We propose an ablation study to determine the contributing mechanisms in the NSiTransformer. Table 3.4 displays the result of this ablation study. We first remove Time2vec, then remove the de-stationary attention mechanism.

Table 3.4: Ablation results for the NSiTransformer. The input sequence length is set to 96, and T is the prediction length. Avg is the average result of all four prediction lengths.

Models	T	NSiTransformer	W/o Time2Vec	W/o DSAttention
Metric		MSE	MSE	MSE
ETT	96	**0.292**	0.296	0.299
	192	**0.374**	0.382	0.378
	336	0.426	**0.420**	0.427
	720	0.420	0.423	**0.410**
	Avg	0.378	0.380	**0.377**
ECL	96	**0.147**	0.148	0.147
	192	**0.162**	0.162	0.162
	336	**0.175**	0.179	0.176
	720	**0.208**	0.213	0.208
	Avg	**0.173**	0.175	0.173
Weather	96	0.171	0.174	**0.171**
	192	**0.224**	0.225	**0.224**
	336	0.281	0.282	**0.280**
	720	**0.356**	0.359	0.360
	Avg	**0.258**	0.259	**0.258**

The ablation study highlights the mechanism that contributes the most in each experiment. In the ETT dataset, both Time2Vec and the De-stationary attention contribute depending on the prediction length. At $T = 96$ and $T = 336$, the de-stationary attention is driving the MSE down, while at $T = 192$ the combination of both mechanisms is best performing. The ECL dataset benefits the most from Time2Vec embedding, as the default self-attention performs as well as the NSi-Transformer at most prediction length. The Weather dataset performs best when using the combination of both mechanisms at all prediction length, highlighting the relevancy of de-stationary attention and Time2Vec for predicting this dataset.

3.7.2 Mixed Floating Point Precision

Due to the possibly heavy computation at large k, we use mixed floating point computation for the ECL and Traffic dataset. Mixed floating point truncates Float16 to Float8 unless the supplementary precision is necessary.

Table 3.5: Difference in performance for ETT and Weather with and without mixed precision.

Models Metric	T	FP16 MSE	Mixed MSE
ETT	96	**0.292**	0.293
	192	**0.374**	0.384
	336	**0.426**	0.427
	720	**0.420**	0.423
	Avg	**0.378**	0.381
Weather	96	**0.172**	**0.172**
	192	**0.222**	**0.222**
	336	**0.278**	0.282
	720	**0.356**	0.359
	Avg	**0.257**	0.259

Table 3.5 displays the difference in performance for ETT and Weather with and without mixed precision. The MSE is equal or slightly higher in Mixed precision, indicating that mixed floating point is a valid option for larger datasets when computation can be a bottleneck.

3.7.3 Hyperparameters Sensitivity

We experiment with different values of d_{time} and top-k. The maximum value of topk for a given dataset is equal to the number of features, plus the embedded time dimensions, 35 in Weather and 21 in ETT. Figure 3.2 presents the influence of top-k on the MSE of the Weather dataset. As the top-k grows, the MSE diminishes.

Figure 3.2: Influence of top-k hyperparameter on MSE for Weather dataset.

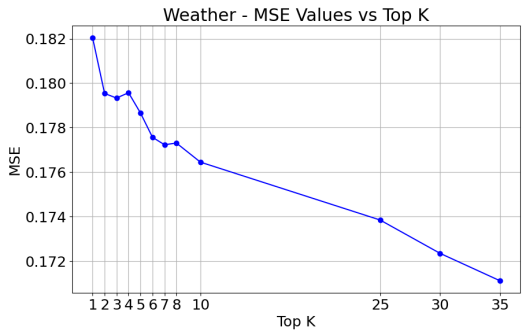


Figure 3.3 presents the influence of top-k on the MSE of the Weather dataset. There is a local minima at $k = 6$, indicating that a top-k value too high can also be detrimental to the performance of the model.

Figure 3.3: Influence of top-k hyperparameter on MSE for ETT dataset.

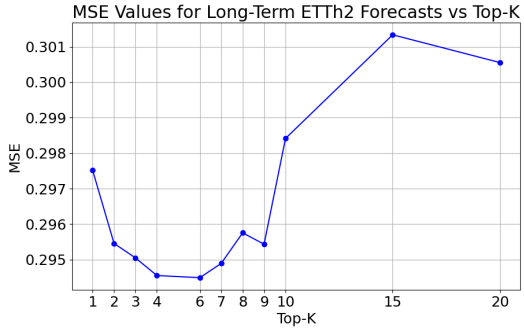


Figure 3.4 presents the influence of the hyperparameter d_{time} on the MSE for the dataset ECL. It appears that the model experiences an initial loss in performance, before reaching a its best performance at 32 dimensions. As the number of dimensions increases, the model experiences worse performance.

Figure 3.4: Influence of d_{time} hyperparameter on MSE for ECL dataset.

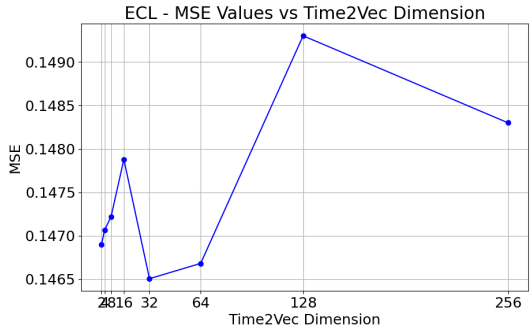
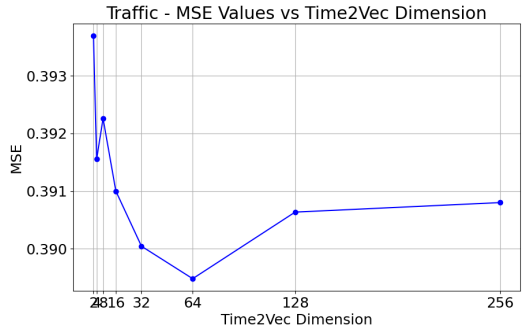


Figure 3.5 presents the influence of the hyperparameter d_{time} on the MSE for the dataset Traffic. The trend is more pronounced on the Traffic dataset, as increasing the number of Time2vec dimensions dimishes the loss up to 64 dimensions, but further increasing it leads to diminishing returns and a worse MSE.

Figure 3.5: Influence of d_{time} hyperparameter on MSE for Traffic dataset.



These results lead to believe that efficient tuning of d_{time} is related to the number of features in the dataset. As the number of features increases, the best value for d_{time} increases.

3.7.4 Depth of the variate projector

We experiment with different depths for the variable projector network. Table 3.6 displays the forecasting result when using a simple linear projector network versus a deeper network with 128 hidden layers. We find that using a high number of hidden dimensions considerably increases computational overhead but can be

beneficial, especially at higher prediction length. Numerous hidden layers can also lead the model to overfit the training dataset.

Table 3.6: variable projector depth. The input sequence length is set to 96, and T is the prediction length. Avg is the average result of all four prediction lengths.

Models	T	NSiTransformer No hidden layers		NSiTransformer 128 Hidden Layers	
Datasets		MSE	MAE	MSE	MAE
ETT	96	**0.292**	**0.345**	0.294	0.347
	192	**0.374**	**0.396**	0.382	0.400
	336	0.426	0.435	**0.419**	**0.432**
	720	0.420	0.443	**0.415**	**0.440**
	Avg	0.378	**0.404**	**0.377**	**0.404**
Weather	96	**0.171**	**0.211**	0.172	**0.211**
	192	0.224	0.256	**0.222**	**0.255**
	336	0.281	0.297	**0.278**	**0.296**
	720	**0.356**	**0.348**	0.357	0.349
	Avg	**0.258**	**0.278**	0.257	**0.278**

3.7.5 De-stationary Factors

We sample the tensors of τ and δ during testing and represent it as heatmaps. The de-stationary factors represent the evolving relationship between the features.

Figure 3.6: De-stationary factors for ETT dataset in testing.

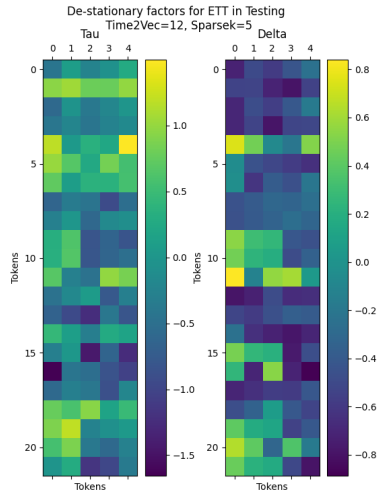


Figure 3.6 presents the de-stationary factors sampled during testing for the ETT dataset. The Tau shows the scaling of the attention scores, and the Delta the shifting of the attention scores. Token 1 is noteworthy as it is scaled up but shifted down. However, token 4 is both scaled and shifted up, indicating that the variable projector believes this token to have a strong relationship relative to other tokens.

Figure 3.7: De-stationary factors for Weather dataset in testing.

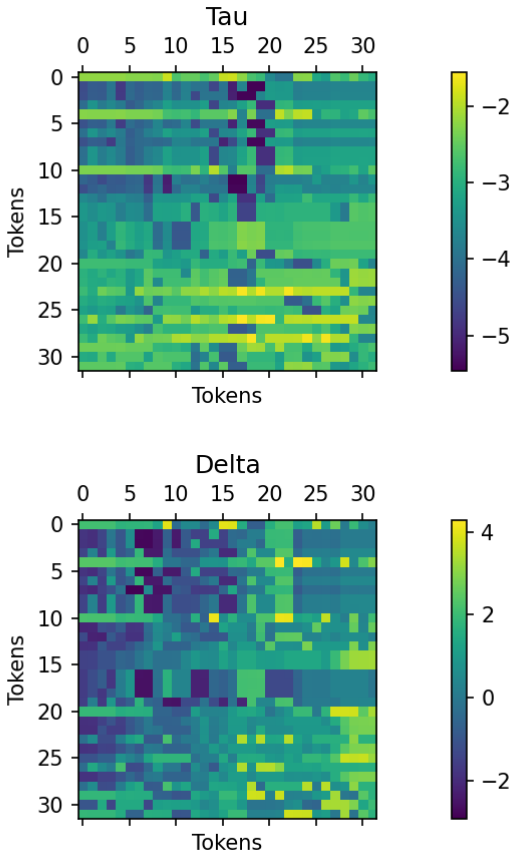


Figure 3.7 presents the de-stationary factors sampled during testing for the Weather dataset. Tokens 0, 4, 10, 20, 23 and 29 exhibit the same pattern, with a higher tau and delta than most. In both figures, the distribution of tau and delta appears to be similar, as evidenced by the different zones in the heatmaps.

3.7.6 Relevance maps

We provide supplemental interpretability by calculating the relevance of each token using Chefer et al [81] general technique adapted to the iTransformer [42].

Figure 3.8: Total relevance of tokens for Weather Dataset. Dataset features in blue, Time2vec features in red.

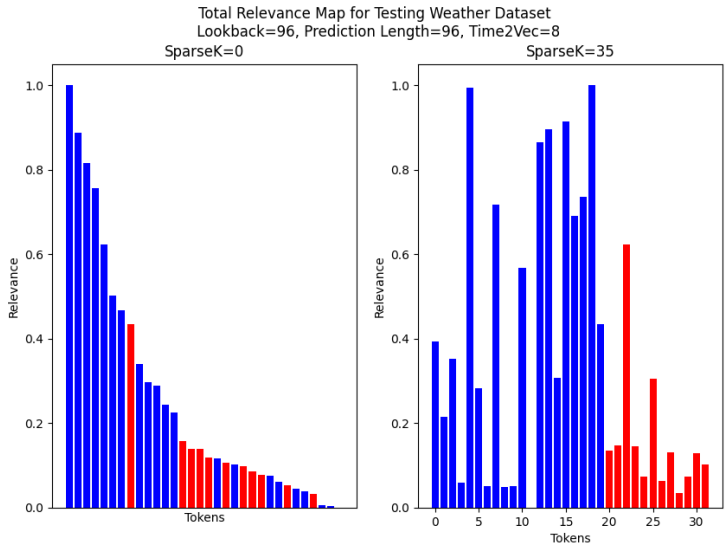


Figure 3.8 presents the total relevance of tokens for the Weather dataset. The relevance of the different tokens is significantly scattered, and 9 tokens reach a relevancy score ≥ 0.5. Notably, 2 of the embedded time features are prevalent, tokens 22 and 25. Those tokens are sine components representing the temporality dependencies of the dataset. The high relevancy of multiple tokens also corroborate the use of a higher k. In the ETT dataset, only a few tokens are relevant, and the model performs best at $k = 5$. On the other hand, the Weather dataset performs best at a significant $k = 35$, with only about 10 more features after Time2Vec. These results demonstrate that by observing the relevancy of the tokens we can determine experimentally the local best k for a given dataset. Essentially, sparseK acts as a magnifying glass for the most important interactions. With a first run at $k = 0$, we plot the relevance maps to visualize the most important tokens. By recognizing the tokens that are most relevant and setting a threshold, we can match the sparsek to that threshold, especially in high dimensions datasets such as ECL where high k can be costly.

Figure 3.9: Total relevance of tokens for ECL Dataset. Dataset features in blue, Time2vec features in red.

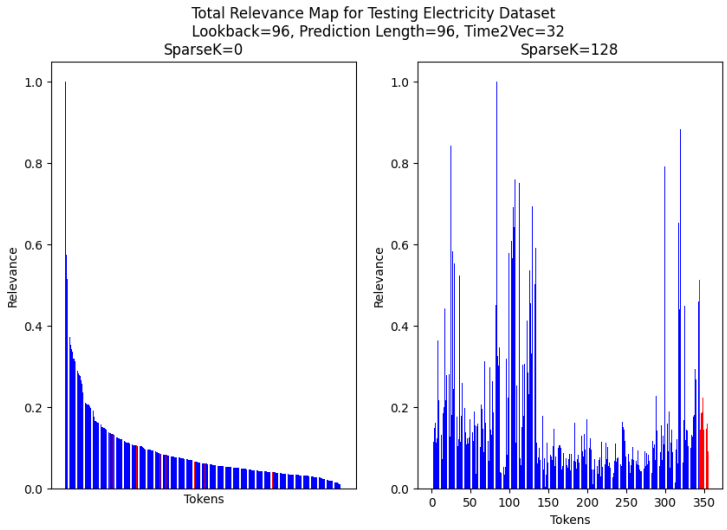


Figure 3.9 presents the total relevance of tokens for the ECL dataset. The large number of features of this dataset smooths out the distribution of relevance. The Time2Vec features are particularly prevalent in this dataset, as corroborated by Figure 3.4. We chose $k = 128$ as a compromise between highlighting the relevant features and computation time.

3.8 Conclusion

This chapter proposes the NSiTransformer, an alternative architecture that places itself in the inverted transformer framework, and implements a custom attention mechanism and time embedding. The model performs at state-of-the-art level on the dataset benchmarks for long-term forecasting. The experiments highlight the efficiency of the attention mechanism and Time2vec in the different datasets and proposes a relevant use-case for each. The ETT, ECL and Traffic datasets proposed the largest gain in performance, with an average gain of 0.005 over the current state-of-the-art (vanilla iTransformer). The performance on the Weather dataset was equivalent on average to the iTransformer.

Specific interpretability techniques are also implemented to gain insights in the inner workings of the model. De-stationary factors are sampled during testing to keep track of the most important relationships between variables. Supplemen-

tal interpretability is provided through the use of token relevance, which helps determine which tokens are the most influential in the prediction. This information allows for an effective tuning strategy for the new hyperparameters d_time and k. The depths of the variable projector was also treated as an hyperparameter. The gains in longer prediction length with a deeper network indicate an increased flexibility for the NSiTransformer.

3.9 Future work

Recent studies proposed alternative architectures for foundational models such as the multi-layer perceptrons used in this paper. Kolmogorov-Arnold networks (KAN) [104] in particular stand out as an ideal candidate for better interpretability and possibly better performance. A potential avenue for work would be to modify the original TrmBlock from the iTransformer and replace it with KANs. It is not well determined how this changed would alter the computational requirements. An initial idea could be to replace the variable projector linear layer with a KAN. Another approach could be to approximate the frozen MLP layers using KAN layers and try to deduce a closed-form expression of the model.

Other types of data augmentation are also a promising avenue for work.Fundamentally, Time2Vec is a form of data augmentation designed for temporal features. Domain-specific techniques combined with autoencoders [105] could further refine the model. Other tasks could also be explored: The original non-stationary transformer remains second in the short-term forecasting, classification, and imputation benchmarks of the Timeseries library. It is likely that the inverted version can perform well in those tasks too.

Finally, the high interpretability that comes with the inverted framework is extremely valuable. Future work could use this model to demonstrate the contribution of a variable to the prediction, analogous to a variable to variable correlation. This transparency also broadens the field of applications to more critical industries. Law and finance, for instance, might value the accountability offered by more interpretable models while maintaining state-of-the-art performance. Other fields that suffer from the curse of dimensionality could use the innate interpretability to remove the variables that are not relevant enough, similar to a principal components analysis.

Chapter 4

Centralized Multi-Agent Reinforcement Learning

This chapter presents a framework for centralized multi-agent reinforcement learning. Although this framework can be applied to any RL algorithm, we focus on proximal policy optimization. The downstream task for this model is fine-tuning of timeseries predictors, which is detailed in Chapter 7.

4.1 Introduction

In the current landscape of machine learning, the complexity and volume of data require innovative solutions to harness the most out of a dataset. This chapter aims to lay down the foundations of a multi-agent theoretical framework, specifically a centralized multi-agent proximal policy optimization approach. Proximal policy optimization (PPO) [106] is presently considered state of the art in reinforcement learning, a subset of machine learning. The algorithm is detailed in section 4.3. The framework presented here harnesses the decision-making of several subagents tuned to each prioritize certain aspects of the dataset. These decisions are then ultimately processed by a superagent, tasked with synthesizing the opinion of each subagent and reaching the final action. The superagent is equipped with an attention module that dynamically balances between environment variables and subagent input. From this decision-making process comes the term *centralized*, there is no communication between the subagents and the resulting subagent actions (subactions) are part of the superagent observation space.

The core research questions developed in this chapter are the following: how can one build a resilient and versatile artificial intelligence framework using a centralized multi-agent approach? What are the pros and cons of this approach in terms of performance, sample-efficiency and interpretability?

The chapter is structured as follows: Section 4.2 includes a contextualization and a review of the existing literature. Section 4.3 introduces the mathematical definition of the framework. Section 4.4 presents the results of the framework against standard strategies and alternative models. Section 4.5 proposes an ablation study. Section 4.6 discusses training time. Section 4.7 develops interpretability tools for the model. Section 4.8 is the conclusion to this study.

4.2 Literature Review

PPO is a policy gradient method developed by John Schulman et al. in 2017 [106]. The key innovation of this algorithm over older methods such as TRPO [107] or ACER [108] is the clip function that constrains policy updates of the agent. PPO has been used in a wide variety of applications: Atari games [109], track racing games [110], suspension monitoring for cars [111], and image captioning [112]. A number of articles have proposed innovations to the base algorithm, for instance an alternative minimization target [113], [114] introduced policy feedback; specifically improving early learning stages, which are recognized as a potential weak point of PPO [115]. Recently proposed improvements include a shift in learning to offline policy optimization [116] and including conservatism [117].

Multi-agent methods have gained significant attention in the field of reinforcement learning, particularly for their capability to simulate complex systems involving interactive agents. A notable early work in multi-agent systems is [118] which explored the dynamics of cooperative and competitive agents in a shared environment. Recent advancements have integrated PPO into multi-agent applications: [119] applied multi-agent PPO to competitive and cooperative tasks, [120] successfully employed multi-agent reinforcement learning in the complex environment of the Dota 2 game. The integration of PPO into multi-agent systems has also been explored in real-world scenarios such as traffic light control [121], and collaborative robotics [122]. Innovations specific to multi-agent PPO include [123], which introduced a meta-learning approach to enhance adaptability across different tasks and agent configurations and [124], which presented the concept of leniency in multi-agent learning, mitigating the non-stationary issue commonly faced in such environments.

Attention is a machine learning mechanism designed to imitate human awareness. Attention was brought to the forefront of the field with the transformer architecture, a self-attention-based architecture that enabled the recent breakthroughs in large language models [29]. It has since seen many implementations including in recurrent neural networks for search results customization [125], missing data imputation [126], and in computer vision [127]. In reinforcement learning, attention models have been developed within theoretical frameworks [128] and diverse applications, such as source code summarizing [129], dynamic graph problems

[130], and road networks management [131].

The novelty of this model lies in the combination of reinforcement learning concepts. Multi-agents models have been explored in adversarial and cooperative settings, but to our knowledge not in an independent centralized manner. The addition of the attention module to the superagent provides avenues of interpretability and fine-tuning for the model that were not previously studied. The method-agnostic nature of this design also increases its potential for future studies, as this study only explores its application with PPO. This study also creates an opportunity for further applications of the design in simulated environment encompassing diverse fields.

4.3 Centralized Multi-Agent Proximal Policy Optimization

As mentioned in [132], implementation is key in deep policy gradient algorithms. As such, the framework below is implemented using the clean-rl library [133].

4.3.1 Proximal Policy Optimization (PPO)

- Policy Function: For an agent x, its policy at time t is a probability density function denoted as $\pi_\theta(a_t|o_t)$, where θ are the parameters of the policy, o_t is the observation for agent x at time t, and a_t are the actions that can be taken. The policy is then sampled to obtain the action taken $\alpha_t \sim \pi_\theta(a_t|o_t)$.

- Objective Function: The PPO objective function is defined as:

$$L^{PPO}(\theta) = \mathbb{E}_t \left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t) \right]$$

  where $r_t(\theta) = \frac{\pi_\theta(a_t|o_t)}{\pi_{\theta_{\text{old}}}(a_t|o_t)}$ is the probability ratio, ϵ an hyperparameter and $\hat{A}_t$ is an estimator of the advantage at time t, typically computed using Generalized Advantage Estimation (GAE).

- Advantage Estimation: The advantage $\hat{A}_t$ is computed as:

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \ldots + (\gamma\lambda)^{T-t+1}\delta_{T-1} \tag{4.1}$$

  with $\delta_t = r_t + \gamma V(o_{t+1}) - V(o_t)$ and V a learned state-value function.

- **Training Process:** The agent is trained by iteratively updating its policy parameters. This involves:

  1. Collecting trajectories by interacting with the environment using the current policy.
  2. Estimating the advantages using GAE.

3. Calculating the surrogate objective function.
4. Optimizing the surrogate objective function using gradient ascent while ensuring the updates stay within a specified clipping range to maintain policy stability.

4.3.2 Centralized Multi-Agent Model

- Centralization: Each subagent is trained independently on its own environment. The action taken by a subagent on its given environment does not influence the other environments, and as such there is no communication between the subagents. The local observation for agent x_i at time t is represented by $o_{t,i}$.
- Policy Representation: The policy of an agent i is $\pi_{\theta_i}(a_{t,i}|o_{t,i})$.
- Sampling of the Policy: $\alpha_t, i \sim \pi_{\theta_i}(a_{t,i}|o_{t,i})$ where α_t, i is the action taken by agent i at time t.
- Reward Function: Each agent x_i has its own reward function $R_i(o_t, a_{t,i})$.
- Training Process: Agents are trained iteratively, updating their policy parameters using the PPO objective function.

4.3.3 Superagent Decision-Making Model

- Superagent's role: The superagent x_f makes the overarching decision, influenced by the decisions of the subagents $\{x_1, x_2, \dots, x_n\}$ and the current state of the environment o_t.
- Aggregation Function: the aggregation function $\mathcal{F}$ is a linear or non-linear function that combines the outputs of the subagents and the current state of the environment:
$$s_t^f = \mathcal{F}(\alpha_{t,1}, \alpha_{t,2}, \dots, \alpha_{t,n}, o_t; \phi) \tag{4.2}$$
where s_t^f is the state at time t, and ϕ are the parameters of the aggregation function.
- Final Decision-Making Policy: The superagent's policy $\pi_{\theta_f}(a_t^f, s_t^f)$ is then sampled to produce the final action α_t^f.

4.3.4 Attention Mechanism in Decision-Making

To enhance the decision-making process, an attention mechanism is integrated into the superagent's framework. This mechanism is designed to dynamically prioritize the influence of subagent actions and the environmental state on the final decision-making process.

- **Attention Module Construction**: The attention module consists of two main components:
  - Linear transformations that compute the attention scores for environmental states and subagent actions respectively, denoted as f_{env} and f_{sub}.
  - A softmax layer that normalizes these scores to form attention weights.
- **Input Representation**: Let e_t represent the encoded environmental state and $\alpha_{t,1}, \alpha_{t,2}, \ldots, \alpha_{t,n}$ represent the actions taken by the subagents at time t. z_{env} and z_{sub} are the linearly transformed environmental state and actions. These are processed through their respective linear layers:

$$z_{\text{env}} = f_{\text{env}}(e_t; \theta_{\text{env}}), \tag{4.3}$$

$$z_{\text{sub}} = f_{\text{sub}}([\alpha_{t,1}, \alpha_{t,2}, \ldots, \alpha_{t,n}]; \theta_{\text{sub}}), \tag{4.4}$$

  where θ_{env} and θ_{sub} are the parameters of the linear transformations for the environment and subagent actions, respectively.
- **Attention Weights Calculation**: The attention weights w_{env} and w_{sub} are computed as follows:

$$[w_{\text{env}}, w_{\text{sub}}] = \text{softmax}\left([z_{\text{env}}, z_{\text{sub}}]\right). \tag{4.5}$$

  These weights determine the relative influence of the environmental states and the subagent actions on the decision-making process of the superagent.
- **Feature Aggregation**: The weighted sum of features, influenced by the calculated attention weights, forms the input to the decision-making layers of the superagent:

$$d_t^f = w_{\text{env}} \cdot e_t + w_{\text{sub}} \cdot [\alpha_{t,1}, \alpha_{t,2}, \ldots, \alpha_{t,n}] \tag{4.6}$$

  where d_t is the aggregated decision input for the superagent at time t.
- **Policy Decision**: The superagent uses d_t along with the state s_t to determine the appropriate action α_t^f through its policy network:

$$\alpha_t^f \sim \pi_{\theta_f}(a_t^f | d_t^f) \tag{4.7}$$

  where θ_f are the parameters of the superagent's policy network.

- Finally, we can express the action taken by the superagent relative to the subagents' policy as:

$$\begin{aligned} \alpha_t^f \sim \pi_{\theta_f}(a_t^f | w_{\text{env}} \cdot e_t + w_{\text{sub}} \cdot [\alpha_{t,1} \sim \pi_{\theta_1}(a_{t,1}|o_{t,1}) \\ , \dots, \alpha_{t,2} \sim \pi_{\theta_2}(a_{t,2}|o_{t,2}), \alpha_{t,n} \sim \pi_{\theta_n}(a_{t,n}|o_{t,n})]) \end{aligned} \tag{4.8}$$

This attention mechanism allows the superagent to adaptively focus more on either the subagent actions or the environmental state based on the current scenario, enhancing the flexibility and effectiveness of the decision-making process.

4.4 Benchmarks

Multi-Joint dynamics with Contact, commonly called MuJoCo [134], proposes several standard environments to train and benchmark models on. Three MuJoCo environments were selected as experimental settings. The three environments are: Hopper-v4, Half-Cheetah-v4 and Humanoid-v4. Each environment creates a shape with the goal to learn the most efficient way to move forward. Hopper-v4 is a single jumping leg, Half-Cheetah-v4 is a 2-legged feline, and Humanoid-v4 is anthropomorphic. In these environments, the reward (R) is calculated using several factors: the forward reward (F_r) and control cost ($Ctrl_c$) are common to all tasks. The forward reward is the movement alongside the x-axis, while the control cost is a penalty for each action taken. The Hopper and Humanoid implement a healthy reward (H_r) that determines whether the action is damaging. The Humanoid also implements a contact cost ($Ctct_c$) that penalizes the agent if the contact force with the ground is too high. The subagents are then tested on the base environment over 10 epochs, and ranked by the average cumulative reward. Based on the ranking, eight superagents are then trained with an increasing number of subagents contributing to the observation space. All agents are trained over four million timesteps.

Table 4.1: Reward function formulas

Environment	Reward formula
HalfCheetah-v4	$R = w_f \cdot F - w_{ctrl} \cdot Ctrl$
Hopper-v4	$R = w_f \cdot F + w_h \cdot H - w_{ctrl} \cdot Ctrl$
Humanoid-v4	$R = w_f \cdot F + w_h \cdot H - w_{ctrl} \cdot Ctrl - w_{ctct} \cdot Ctct$

Table 4.1 presents the reward formula for each environment. The term w_i is the weight for a reward term i.

4.4.1 Subagents Performance - Same Reward

In this experiment, eight agents were trained with identical reward functions. Each environment uses the default configuration, but a different random seed.

Table 4.2: Subagents performance across HalfCheetah-v4, Hopper-v4, and Humanoid-v4

Subagent	Average Reward (HalfCheetah-v4)	Average Reward (Hopper-v4)	Average Reward (Humanoid-v4)
1	-719.19	1207.45	**3187.72**
2	-287.17	**1023.79**	2857.51
3	-388.11	1203.78	3050.35
4	-342.99	1172.62	2971.13
5	-667.03	1152.54	2988.99
6	**-145.95**	1212.92	**2801.29**
7	-180.96	1142.96	2984.01
8	**-898.91**	**1237.77**	3018.10

Table 4.2 shows the performance of subagents across the three test environments. Highest and lowest performing agents for each environment in bold. The average cumulative reward varies by environment, which indicates that the environment state in certain random seeds is more suited for policy gradient learning.

4.4.2 Superagents Performance - Same Reward

Table 4.3: Superagents performance across environments

Number of Subagent(s)	Average Reward (HalfCheetah-v4)	Average Reward (Hopper-v4)	Average Reward (Humanoid-v4)
1	-239.99	1204.68	**2873.40**
2	-214.03	1267.98	2993.25
3	-323.44	**1336.50**	3057.08
4	**-180.76**	1207.91	2999.07
5	-380.27	1259.87	2931.13
6	-414.66	**969.09**	2974.15
7	-440.89	1211.66	2888.64
8	**-498.05**	1266.79	**3178.57**

Table 4.3 displays the average cumulative reward across the three environments. When using subagents with the same reward function, the performance slowly increases for HalfCheetah-v4, until four subagents. When adding more subagents, the model drops in performance. A similar pattern can be observed with Hopper-v4, which peaks at three subagents. The Humanoid-v4 sees no major gain or loss and remains stable across the board, despite a slight boost in performance at eight subagents. A possible explanation could be that using the same reward function, the subagents are unlikely to explore new behaviours that could then be passed to the superagent.

4.4.3 Subagents Performance - Mixed Reward

In this experiment, the reward function coefficients of the subagents were altered to promote emergent behaviours and exploration. In each subagent configurations table, the configurations in bold are the default settings of the environment.

Table 4.4: Subagent configurations for HalfCheetah-v4

(Forward reward weight, Control cost weight)	Average reward
(0.5, 0.5)	105.06
(1.0, 0.5)	-132.32
(1.0, 0.1)	-297.40
(0.5, 0.1)	-650.33
(1.0, 0.01)	-659.02
(1.0, 0)	-680.37
(1.0, 0.1)	**-707.54**
(2.0, 0.001)	-734.57
(3.0, 0.001)	-765.30

As shown in Table 4.4, the HalfCheetah-v4 altered configurations had a wide range of performance on the original environment. The average cumulative reward degraded significantly when the forward reward weight and control cost weight were changed. One explanation could be that techniques to move forward with a very high control cost might have been learnt by the last two subagents, which hindered their performance on the base environment.

Table 4.5: Subagent configurations for Hopper-v4

(Forward reward weight, Control cost weight, Healthy reward weight, Healthy state range, End when unhealthy)	Average reward
(3.0, 0.0001, 0.0, [-100, 100], N)	1468.63
(0.5, 0.01, 2.0, [-100, 100], Y)	1402.84
(0.5, 0.0005, 1.0, [-100, 100], Y)	1267.66
(1.0, 0.001, 0.5, [-100, 100], Y)	1233.06
(1.0, 0.05, 1.0, [-100, 100], Y)	1227.56
(1.5, 0.01, 0.5, [-150, 150], Y)	1192.80
(2.0, 0.001, 0.5, [-100, 100], Y)	1161.34
(3.0, 0.0, 0.5, [-300, 300], Y)	1157.77
(1.0, 0.001, 1.0, [-100, 100], Y)	**894.40**

Table 4.5 presents the configurations for the Hopper-v4 environment. The best performer was surprisingly one of the most altered configuration. This configuration prioritized heavily forward reward by discounting the control costs and health penalties, encouraging risky behaviours. This strategy fell apart when the health penalty was reintroduced, despite completely removing the control cost and increasing the accepted healthy range. The average rewards are however much closer from one configuration to the other, indicating that the environment could be less sensitive to extreme configurations.

Table 4.6: Subagent configurations for Humanoid-v4

(Forward reward weight, Control cost weight, Contact force cost weight, Healthy reward weight, Terminate when unhealthy)	Average reward
(0.5, 1.0, 1.5, 1.0, Y)	4768.19
(1.0, 0.5, 2.0, 1.0, Y)	3524.77
(1.0, 0.5, 1.0, 1.0, Y)	3457.94
(1.5, 0.5, 0.5, 0.5, Y)	3449.48
(0.5, 0.5, 0.5, 2.0, Y)	3265.30
(3.0, 0.4, 0.5, 0.5, N)	3251.51
(1.25, 0.1, 5e-7, 5.0, Y)	**3175.08**
(0.8, 0.1, 0.5, 1.0, Y)	3135.93
(5.0, 0.01, 0.5, 0.0, N)	2881.68

In the Humanoid-v4 environment, the best performer was the complete opposite of the Hopper-v4 top performer, as shown in Table 4.6. The best perform-

ing configuration was tweaked to have an increased control cost weight, and a discounted forward reward weight, promoting a safer and minimalist approach. The configuration with increased forward reward and discounted control cost performed poorly, with two of the bottom configurations disregarding the healthy reward completely.

4.4.4 Superagent Performance - Mixed Reward

Table 4.7: Superagents performance across environments

Number of Subagent(s)	Average Reward (HalfCheetah-v4)	Average Reward (Hopper-v4)	Average Reward (Humanoid-v4)
1	**-638.57**	1211.08	2947.44
2	-580.10	1218.86	3006.55
3	-370.44	**1070.40**	2924.49
4	-509.26	1239.58	**2847.21**
5	**-166.40**	1240.30	**3013.42**
6	-411.29	**1960.75**	2895.03
7	-269.72	1124.24	2924.15
8	-201.02	1317.25	3003.72

Table 4.7 presents the superagent performance across the three environments according to the number of subagents used. In the Hopper-v4 environment, where the action space is smaller, the addition of subagents seems to directly contribute to performance enhancement. The performance consistently improves with the number of subagents up to five, achieving the highest average reward. This trend suggests that the lower dimensionality of the action space allows for effective integration and utilization of the diverse strategies provided by multiple subagents. As the number of possible actions and subagents grow, the number of dimensions handled by the superagent increases. A very low dimensional action space can allow for many subagents without overly complexifying the observation space of the superagent. Beyond five subagents, the benefits stabilize, indicating a potential optimal number of subagents for balancing decision complexity and performance gain in this environment.

4.5 Ablation

We perform an ablation study by removing the attention module from the model. The observation state is an aggregation of the environment state and the subagent actions.

Table 4.8: Superagents performance across environments - Same reward

Number of Subagent(s)	Average Reward (HalfCheetah-v4)	Average Reward (Hopper-v4)	Average Reward (Humanoid-v4)
1	-816.96	750.50	**3201.09**
2	-774.47	895.89	**3134.73**
3	**-845.92**	**1007.24**	3190.50
4	-743.34	916.88	3198.28
5	-831.56	959.84	3192.95
6	-685.47	785.57	3194.82
7	-822.90	907.81	3191.17
8	**-638.94**	**838.87**	3193.77

As shown in Table 4.8, introducing subagent actions to the environment space without the attention module provides a reduction of the variance of average cumulative reward, especially in the Humanoid-v4 environment. The HalfCheetah-v4 environment presents a lower average cumulative reward with eight subagents, which could be due to the higher number of dimension in the observation space not being counterbalanced by the attention mechanism.

Table 4.9: Superagents performance across environments - Mixed reward

Number of Subagent(s)	Average Reward (HalfCheetah-v4)	Average Reward (Hopper-v4)	Average Reward (Humanoid-v4)
1	-403.16	1086.12	3046.32
2	-411.80	1181.87	**2909.05**
3	-398.84	**1228.52**	2999.23
4	-240.12	1186.45	2992.99
5	-389.94	1254.67	3061.87
6	-354.67	**992.95**	**3097.98**
7	**-736.08**	1158.10	3093.66
8	**-111.10**	1175.43	3072.68

The absence of the attention module is further illustrated in Table 4.9 when using mixed rewards. The HalfCheetah-v4 environment shows a pattern of increasing the average reward as the number of subagents grows, up to five where the performance goes down. However, the best performer is surprisingly the one using eight subagents, indicating that more subagents could potentially enhance the performance regardless. In the Hopper-v4 environment, the lack of attention is the most evident, as the best performer with attention becomes the worst performer without it. This indicates a failure to capture the potential emerging be-

haviours brought up by the subagents. Finally the Humanoid-v4 environment remains consistent across the board, with no discernable pattern that can be linked to the number of subagents or the presence of the attention mechanism. The ablation study shows that the attention module has at best positive impact on the average reward for the Hopper-v4 and at worst no impact for the Humanoid-v4. The integration of the attention module also allows for further interpretability, as demonstrated in section 4.7.1.

4.6 Training Time

The subagents need to be trained before suggesting relevant subactions. This means that the superagent can only be trained after the subagents have completed their own learning. In section 4.4, all agents were trained using four million timesteps as a balance between computational time and performance. This means that a superagent with 2 subagents would have been trained effectively a total of 12 million timesteps. The following section will compare the performance of the centralized multi-agent model with attention with different numbers of subagents versus the performance of baseline PPO at different timesteps. The training time of each subagent and superagent remains 4M timesteps. Since the superagent can only be trained sequentially after the subagents, the total timesteps required to train can be approximated in two different ways:

- The subagents can be trained in parallel in separate environments and learn their respective policies independently. The total training time is then $t_{total} = t_{subagent} + t_{superagent}$, which in the Results section adds up to eight million timesteps.
- The subagents are trained in parallel, but the total training time of the model is a function of the number of subagents: $t_{total} = n_{subagent} * t_{subagent} + t_{superagent}$.

With t denoting the number of timesteps in training and $n_{subagent}$ the number of subagents.

Tables 4.10, 4.11 and 4.12 compare the average reward of the superagent versus vanilla PPO depending of the number of subagents/timesteps and their respective environment.

Table 4.10: Superagent performance versus PPO at different number of subagents/timesteps in the HalfCheetah-v4 environment

Number of Subagent(s)/Timesteps (millions)	Average Reward (Superagent)	Average Reward (Baseline PPO)
1 / 8	**-638.57**	-266.09
2 / 12	-580.10	-527.12
3 / 16	-370.44	-453.93
4 / 20	-509.26	-628.43
5 / 24	**-166.40**	**-150.54**
6 / 28	-411.29	-328.68
7 / 32	-269.72	**-628.64**
8 / 36	-201.02	-496.64

In Table 4.10, the 24 million timesteps baseline PPO performs the best. The model experiences diminishing returns at a higher number of timesteps. At ≤ 6 subagents, the superagent gets outperformed by baseline PPO. But as the number of subagents increases, the superagent beats out baseline PPO.

Table 4.11: Superagent performance versus PPO at different number of subagents/timesteps in the Hopper-v4 environment

Number of Subagent(s)/Timesteps (millions)	Average Reward (Superagent)	Average Reward (Baseline PPO)
1 / 8	1211.08	1036.82
2 / 12	1218.86	802.06
3 / 16	**1070.40**	820.14
4 / 20	1239.58	909.84
5 / 24	1240.30	756.75
6 / 28	**1960.75**	**1185.06**
7 / 32	1124.24	**294.01**
8 / 36	1317.25	735.11

In Table 4.11, the superagent beats out baseline PPO. According to the second approach, the best performer in vanilla PPO has an equivalent training time to the best performer of the superagents, but a much lower reward. This result further indicates the adequacy of the centralized multi-agent model with attention for this environment.

Table 4.12: Superagent performance versus PPO at different number of subagents/timesteps in the Humanoid-v4 environment

Number of Subagent(s)/Timesteps (millions)	Average Reward (Superagent)	Average Reward (Baseline PPO)
1 / 8	2947.44	2818.45
2 / 12	3006.55	2224.58
3 / 16	2924.49	2058.88
4 / 20	**2847.21**	2178.31
5 / 24	**3013.42**	**1854.18**
6 / 28	2895.03	2108.89
7 / 32	2924.15	2283.85
8 / 36	3003.72	**2897.81**

In Table 4.12, the baseline PPO experiences heavy diminishing return as the number of timesteps increases, before increasing again. This could be due to an overfit in training in larger timesteps. The superagent remains stable with the number of subagents increasing, while not significantly improving upon the result of baseline 8M PPO. A possible avenue for improvement could be to further explore the optimal configurations for the subagents that cover a targeted range of useful behaviours.

Table 4.13: Superagents performance across environments - Mixed reward at 4M timesteps

Number of Subagent(s)	Average Reward (HalfCheetah-v4)	Average Reward (Hopper-v4)	Average Reward (Humanoid-v4)
1	-434.83	1211.15	**3068.60**
2	-400.35	1220.97	3152.63
3	-489.26	1178.29	3116.24
4	-232.20	**1250.41**	3143.65
5	**-188.02**	1225.38	3143.34
6	-214.85	1200.25	3149.51
7	-315.33	1213.10	3182.25
8	-247.26	1198.95	3149.07
Baseline PPO	**-579.76**	**895.19**	**3190.66**

Table 4.13 presents the subagents performance across environments. The subagents and superagents were trained two million timesteps each, and the baseline PPO 4M timesteps. In HalfCheetah-v4, the top performer remains the superagent with five subagents. There is little variance in average reward for the Hopper-v4

environment, as all values except baseline are within ±60. In both HalfCheetah-v4 and Hopper-v4, the superagents significantly outperform baseline. In Humanoid-v4, baseline is the best performer and outperforms all superagents. With seven subagents, the superagent comes close to outperforming benchmark PPO.

4.7 Interpretability and Scalability in Reinforcement Learning

4.7.1 Attention weights

Extracting the weights attributed to each component of the superagent state can help us interpret the model's decision making. We record and plot the attention weight for the three superagents with the best cumulative average reward.

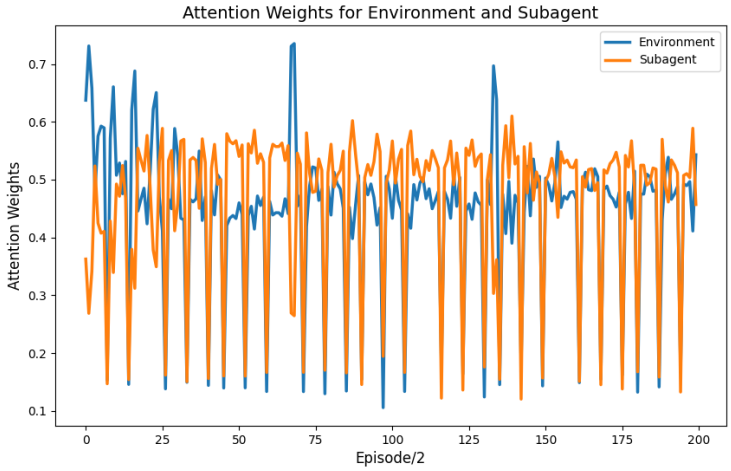


Figure 4.1: Attention weights per episode/2 - HalfCheetah-v4

In Figure 4.1, The HalfCheetah-v4 strikes a balance of attention between the subactions and the environment state. A possible explanation for this distribution of attention could be that further training is needed to reach more stable attention weights.

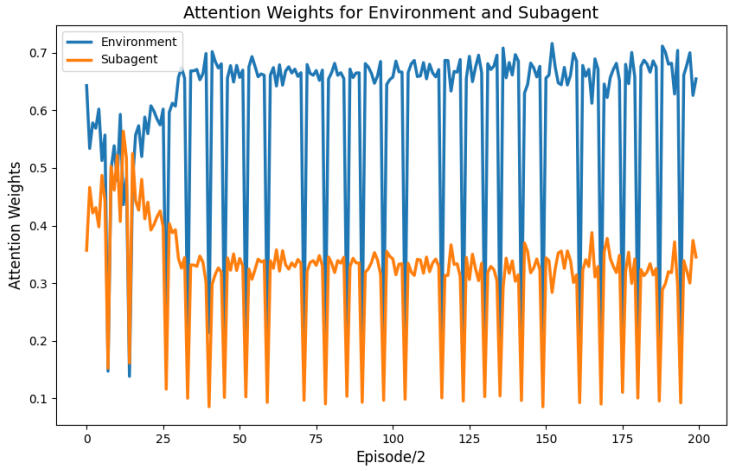


Figure 4.2: Attention weights per episode/2 - Hopper-v4

The Hopper-v4 rapidly stabilizes at a 70-30 split between the environment state and the subactions, as shown in Figure 4.2. Since this model outperformed the baseline considerably, a possible tool to tune the number of subagents and their configurations could be the distribution of attention weights. A quick convergence to a stable split of attention between subactions and environment state could indicate efficient prioritization from the superagent.

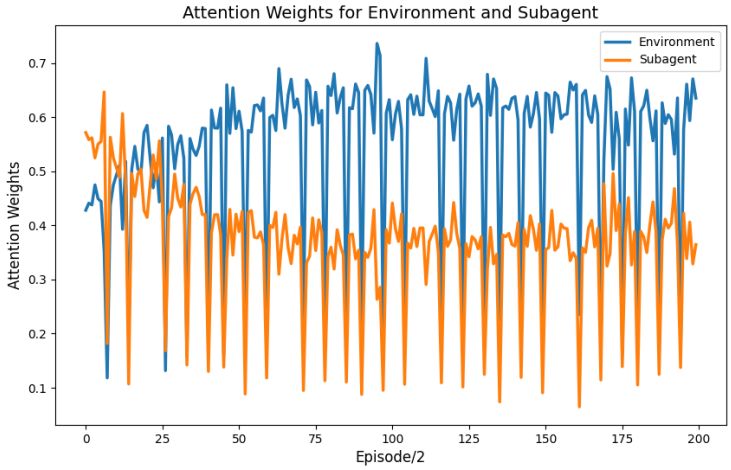


Figure 4.3: Attention weights per episode/2 - Humanoid-v4

In Figure 4.3, the Humanoid-v4 is shown to follow a similar yet slower trend than the Hopper-v4. This could be due to needing a longer training time, or the difference in dimensions in the action space. The latter seems more likely, as the action space of the Hopper-v4 is only 2-dimensional and the action space of the Humanoid-v4 has 64 dimensions.

4.7.2 Cosine distance between actions of the superagent and subagents

In order to evaluate how close the action of the superagent are from the action of its subagents, we calculate the cosine distance [135] between the action vectors given by the subagents, and the one calculated by the superagent. We recorded this data over 10 epochs on a randomly seeded environment for the three superagents with the best cumulative average reward.

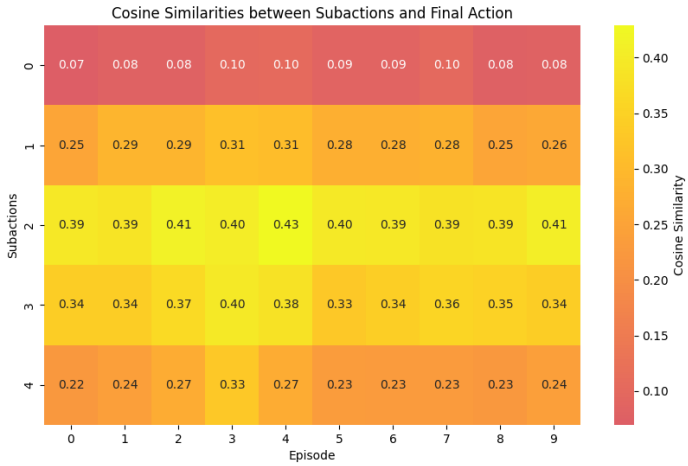


Figure 4.4: Heatmap of cosine similarities between the subactions and the superagent actions - HalfCheetah-v4

In Figure 4.4, the superagent actions show very little cosine similarity with the actions taken by the subagent 0, despite this subagent earning the highest reward out of all the altered configurations. Instead, the superagent actions have a high cosine similarity with the actions of subagent 3, which used the base parameters. This could mean that the attention module failed to recognize behaviours that could potentially earn a higher reward. This could also mean that the behaviour proposed by the subagents were not sustainable long term strategies and instead were shortcuts to a local optimum.

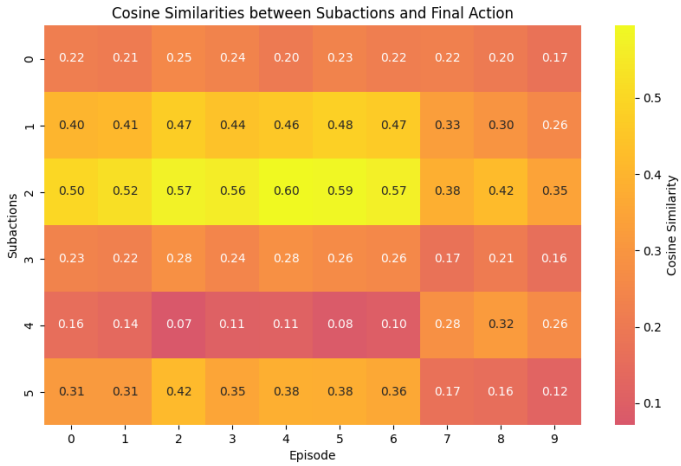


Figure 4.5: Heatmap of cosine similarities between the subactions and the superagent actions - Hopper-v4

The Hopper-v4 environment presents the clearest trend, as shown in Figure 4.5: the three first subagents, which are the best performers, have a higher cosine similarity between the superagent actions and the subactions. This means that in this environment the attention managed to capture the relevancy of the actions advised by the subagent. The superagent's actions also demonstrate fluctuations across episodes in the cosine similarity with subactions, indicating that different strategies are prioritized depending on the environment state and the attention given to the subactions.

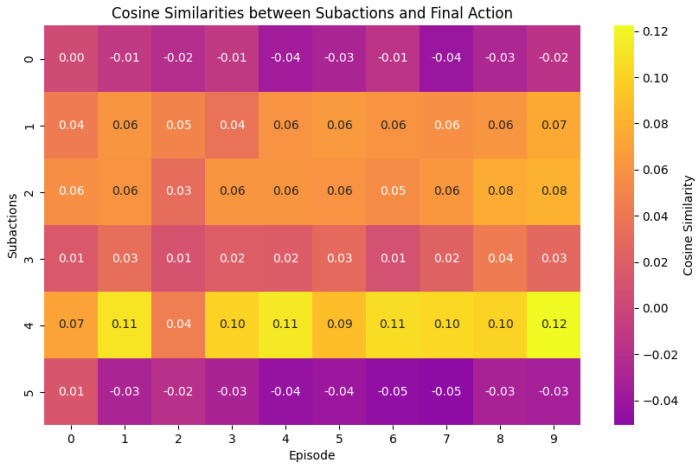


Figure 4.6: Heatmap of cosine similarities between the subactions and the superagent actions - Humanoid-v4

In Figure 4.6, the Humanoid-v4 environment presents no similarity between the actions of the superagent and the subactions. The subagent 0, which performed outstandingly on the base environment, has a nearly 0 cosine similarity in its action with the superagent. The conclusions are similar to the HalfCheetah-v4 environment, and the model fails to capture the value of the subactions despite the weighting of the aggregation function through the attention module.

4.8 Conclusion

This study presents a novel multi-agent architecture for Proximal Policy Optimization which harnesses attention to prioritize behaviours from subagents depending on the environment state. The model outperforms the baseline in the Hopper-v4 and HalfCheetah-v4 environments in mixed reward, and performs at baseline level in the Humanoid-v4 environment. Using mixed reward functions, the framework is versatile and applicable to real-world problems.

The model performs best in a continous action space with few dimensions, as the benefits of augmenting the environment state with the suggested actions of the subagents fall off as the number of dimensions in the action space increases. This approach also provides additional interpretability tools by studying the attention weights and the cosine similarity between actions and subactions.

Future research could focus on implementing this model in real-life reinforcement learning problems. The attention module could also be extended into multi-

headed attention, in order to have separate channels for each subagent. Another possible improvement could be to apply this method to another algorithm than PPO, as the principle behind the centralized multi-agent approach is model agnostic.

Part IV

Application: Machine Learning Models on ESG Financial Datasets

Chapter 5

A Financial Dataset with ESG Ratings

This chapter introduces the design of the dataset used in the applicative side of this thesis. The following sections discuss the initial construction of the financial dataset, the addition of ESG ratings from different providers, and the data augmentation tools applied. Similarly to Chapter 2, we focus on the S&P500 for the abundance of data available.

5.1 Dataset Providers

The construction of the dataset starts with an assessment of the options for data providers. In finance, the reliability and quality of the data is fundamental to any robust analysis. The main providers used in this thesis are the following:

- **Refinitiv**, more recently known as **LSEG Data & Analytics** is a global leader in financial data and analytics, and serves as one of the primary sources for this study. Through the Eikon terminal Refinitiv provides an extensive real-time customisable news feed, as well as powerful charting tools, data extraction, cross-asset calculators, portfolio analytics and a seamless integration in Excel. Refinitiv covers a large array of use cases, from corporate treasury, sales and trading and investment research. Refinitiv is also a significant ESG providers, covering over 80% of the global market capitalization with over 450 different ESG metrics. Refinitiv Eikon was extensively used throughout this study for both financial and ESG data. In particular, the Python proxy combined with the formula builder provided reliable and extensive data. The rigorous validation process ensures little to no discrepancy or missing value, and Refinitiv has earned the general trust of the industry, especially in the context of industry research.

- **Sustainalytics** is a company that rates the sustainability of listed companies, based on their ESG performance. Sustainalytics was bought out in 2020 by Morningstar, Inc and started being included with Yahoo Finance in 2018 for over 2000 companies. Their ESG ratings are accessible on demand through their Global Access and API. Their research universe, covers over 16,000 companies across public equity, fixed income, and private sectors [10]. Their ratings are used by a variety of asset managers, asset owners and banks to define a sustainable investment strategy and create portfolios with strong ESG performers. They also provide monitoring and reporting to maintain active ownership and regular updates on the evolution of companies' engagement towards a sustainable future.
- **Sustainability Accounting Standards Board (SASB)** identify the most critical sustainability-related issues to investor in a wide array of industries. Since August 2022, the SASB standards have taken an important role by becoming the basis for the first two International Financial Reporting Standards (IFRS) dedicated to sustainability, IFRS S1 *General requirements for Sustainability-related Disclosures* [136] and S2 *Sustainability-related Disclosures*. The first publication of the SASB standards date back to 2018, using a project-based model. The standards are based on a combination of technical knowledge and public comments to build and amend the standards for specific companies.

Other providers that were considered or used:

- **Bloomberg** is the primary data provider in finance and widely considered as the gold standard for integration of analytics tools and real-time data. Perhaps even more important than the data provided, access to the proprietary Bloomberg Terminal is the production swiss knife of any institutional investor. The large customer base and chat integration foster an ease of access to other Bloomberg users, filtered geographically or by interest. Although Bloomberg is the primary data provider in a production environment and extremely efficient when it comes to real-time data, this study focused on historical data. Financial actors want to act quick to react to minuscule market changes, and depend on this data to maintain a high sensitivity. The considerable financial strain of a Bloomberg license is often a bottleneck for companies and academia, and was an unwarranted constraint for the time granularity of data required in this study.
- **Yahoo Finance** complements Refinitiv in this study by providing international market, free stock quotes, and up-to-date news for free. The unofficial *yfinance* package was invaluable in exploratory stages to prototype the dataset. There are limitations to the quantity of data extractable for a given day, which limited the use cases long term.

5.2 Financial features and data augmentation

Raw data can be enriched through calculated features and adjustments to better model financial dynamics and support robust predictions. In this study, similarly to Chapter 2, data augmentation involves calculating log returns, controlling the returns using Fama-French 5, and deriving technical indicators such as **RSI**, **MACD**, and **Bollinger Bands**. Technical indicators are derived from historical price and volume data to capture patterns, momentum, and volatility. These indicators are vital for machine learning models, providing features that encapsulate complex financial behaviors.

5.2.1 Raw data

Based on Chapter 2, we determined that sectors and materiality were driving factors that influenced the correlation between the controlled logreturns and the ESG ratings. As such, we decided to focus on 3 GICS sectors that presented different levels of correlation: Technology, Financial Services and Industrial. Table 5.1 presents the ticker symbols included in each sector.

Category (Count)	Tickers
Technology (69)	ACN, ADBE, AMD, AKAM, APH, ADI, ANSS, AAPL, AMAT, ANET, ADSK, AVGO, CDNS, CDW, CSCO, CTSH, GLW, CRWD, DELL, ENPH, EPAM, FFIV, FICO, FSLR, FTNT, IT, GEN, GDDY, HPE, HPQ, IBM, INTC, INTU, JBL, JNPR, KEYS, KLAC, LRCX, MCHP, MU, MSFT, MPWR, MSI, NTAP, NVDA, NXPI, ON, ORCL, PLTR, PANW, PTC, QRVO, QCOM, ROP, CRM, STX, NOW, SWKS, SMCI, SNPS, TEL, TDY, TER, TXN, TRMB, TYL, VRSN, WDC, ZBRA
Financial Services (72)	AFL, ALL, AXP, AIG, AMP, AON, ACGL, AJG, AIZ, BAC, BRK.B, BLK, BX, BK, BRO, COF, CBOE, SCHW, CB, CINF, C, CFG, CME, CPAY, DFS, ERIE, EG, FDS, FIS, FITB, FI, BEN, GPN, GL, GS, HIG, HBAN, ICE, IVZ, JKHY, JPM, KEY, KKR, L, MTB, MKTX, MMC, MA, MET, MCO, MS, MSCI, NDAQ, NTRS, PYPL, PNC, PFG, PGR, PRU, RJF, RF, SPGI, STT, SYF, TROW, TRV, TFC, USB, V, WRB, WFC, WTW
Industrial (78)	MMM, AOS, ALLE, AMTM, AME, ADP, AXON, BA, BR, BLDR, CHRW, CARR, CAT, CTAS, CPRT, CSX, CMI, DAY, DE, DAL, DOV, ETN, EMR, EFX, EXPD, FAST, FDX, FTV, GE, GEV, GNRC, GD, HON, HWM, HUBB, HII, IEX, ITW, IR, JBHT, J, JCI, LHX, LDOS, LMT, MAS, NDSN, NSC, NOC, ODFL, OTIS, PCAR, PH, PAYX, PAYC, PNR, PWR, RTX, RSG, ROK, ROL, SNA, LUV, SWK, TXT, TT, TDG, UBER, UNP, UAL, UPS, URI, VLTO, VRSK, GWW, WAB, WM, XYL

Table 5.1: Tickers listed by category in a single row per category.

Table 7.1 shows a sample of the financial data extracted for Apple from 2005-12-05 to 2005-12-13. The initial dataset consists of financial data dating from **2005-12-05** to **2024-08-07**.

Date	Open	Low	High	Close	Volume
2005-12-05	2.17	2.15	2.19	2.16	5.84e8
2005-12-06	2.23	2.21	2.25	2.23	8.57e8
2005-12-07	2.24	2.20	2.24	2.23	6.79e8
2005-12-08	2.21	2.19	2.23	2.23	7.90e8
2005-12-09	2.24	2.21	2.25	2.24	5.55e8
2005-12-12	2.26	2.25	2.27	2.26	5.25e8
2005-12-13	2.25	2.24	2.27	2.26	4.94e8

Table 5.2: Sample financial data for AAPL

5.2.2 Log Returns: Capturing Price Movements Logarithmically

Logarithmic returns (log returns) are a fundamental metric for time-series analysis in finance, offering a symmetric and scale-invariant measure of asset price movements. The log return r_t for a given time t is calculated as:

$$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right) \tag{5.1}$$

where P_t and P_{t-1} represent the closing prices at time t and $t-1$, respectively.

Log returns present several advantages:

- **Symmetry:** Log returns ensure that equal percentage increases and decreases yield consistent values.
- **Additivity:** Overlapping periods can be aggregated, simplifying cumulative return calculations, as we did with monthly returns in Chapter 2.
- **Scale Independence:** Useful for comparing returns across assets with varying price levels.

We calculate the autocorrelation function for log returns to evaluate the usefulness of past samples to predict future values. The autocorrelation function can also indicate whether or not a series is stationary. A stationary series has the same joint probability distribution across time, and is often a necessary assumption in time-series analysis. This assumption is not necessary when using the NSiTransformer, which attempts to model the non-stationary series as the process progresses.

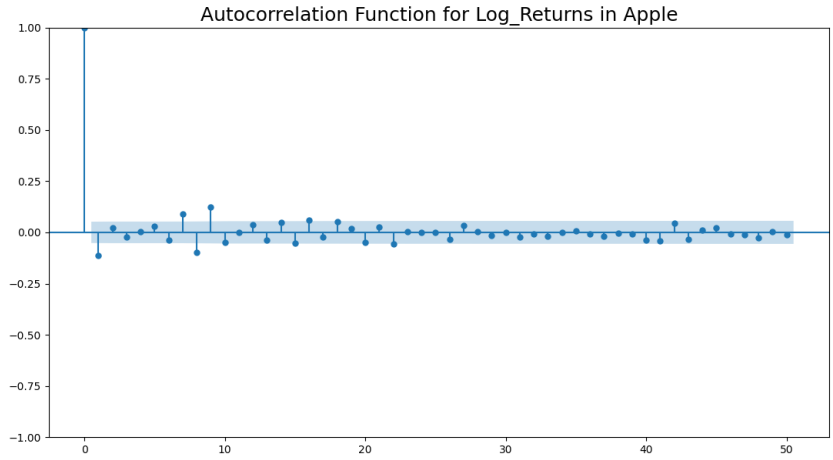


Figure 5.1: Autocorrelation of log returns for AAPL with 50 lags. First data point is self autocorrelation and always 1.

Figure 5.1 presents the autocorrelation of log returns for AAPL with 50 lags. We found a small autocorrelation at certain lags, but the data remains largely not autocorrelated.

5.2.3 Controlled Returns: Adjusting with the Fama-French Five-Factor Model

To isolate firm-specific characteristics and remove broader market effects, log returns are adjusted using the Fama-French five-factor (FF5) model. This model accounts for market-wide influences and fundamental financial drivers.

The FF5 model decomposes returns as:

$$R_{it} - R_f = \alpha_i + \beta_m(R_m - R_f) + \beta_s \text{SMB} + \beta_v \text{HML} + \beta_r \text{RMW} + \beta_c \text{CMA} + \epsilon_{it} \quad (5.2)$$

where:

- R_{it}: Return of asset i at time t.
- R_f: Risk-free rate.
- R_m: Market return.
- **SMB:** Size factor (small minus big).
- **HML:** Value factor (high minus low).

- **RMW:** Profitability factor (robust minus weak).
- **CMA:** Investment factor (conservative minus aggressive).

By regressing the log returns on these factors, the residual component (ϵ_{it}) represents firm-specific, market-neutral returns. This adjustment is crucial for reducing noise and enhancing signal clarity in predictive modeling. More details on the Fama-French model are available in Section 2.4.2.

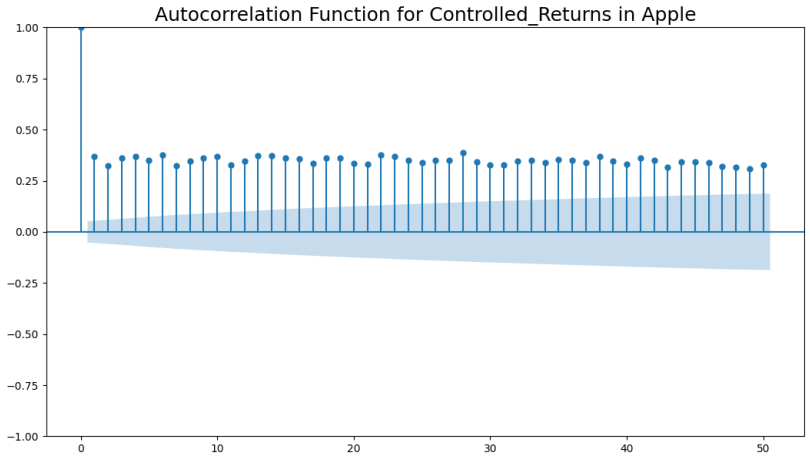


Figure 5.2: Autocorrelation of controlled returns for AAPL with 50 lags. First data point is self autocorrelation and always 1.

Figure 5.2 presents the autocorrelation of the controlled returns with 50 lags. There is a clear positive autocorrelation, notably at lag 1, which indicates that future values tend to move in the same direction as previous lags. The shaded region represents the 95% confidence bounds around zero autocorrelation, and all lags are above the upper band. This slow decay of the autocorrelation function (ACF) can also be indicative that the series is non-stationary. In order to verify that the series is non-stationary, we perform an Augmented Dickey-Fuller (ADF) test [137].

Table 5.3 presents the results of the Augmented Dickey-Fuller statistic test. Since the ADF test statistic (-1.7759) is higher (less negative) than the 5% critical value (-2.8640), and the p-value (0.3925) exceeds 0.05, we *fail to reject* the null hypothesis of a unit root. Therefore, the series is likely non-stationary.

Table 5.3: Augmented Dickey-Fuller Test Results for `Controlled_Returns`

Statistic	-1.7759
p-value	0.3925
Used Lag Order	22
Number of Observations	1396
Critical Values	
1%	-3.4350
5%	-2.8640
10%	-2.5680

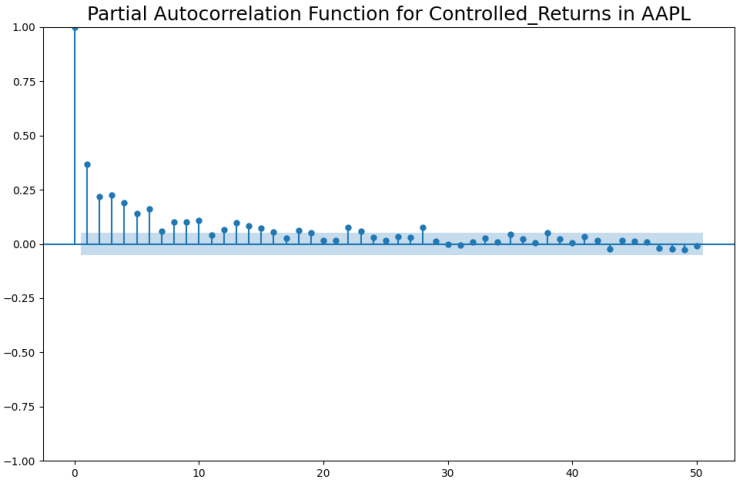


Figure 5.3: PACF for controlled returns in AAPL.

Figure 5.3 displays the Partial Autocorrelation Function (PACF) plot for lags 1 through 50. Significant spikes beyond the confidence bounds indicate direct lagged relationships after accounting for intermediate lags. This pattern suggests a very persistent process consistent with non-stationarity. These signs of the series being non-stationary confirm that the NSiTransformer is adequate to predict controlled returns.

5.2.3.1 Relative Strength Index (RSI)

RSI measures the strength and speed of price movements over a fixed period, indicating overbought (RSI $>$ 70) or oversold (RSI $<$ 30) conditions. Figure 5.4 shows

the RSI of Apple from 2005 to 2023.

$$\text{RSI} = 100 - \frac{100}{1 + RS}, \quad \text{where } RS = \frac{\text{Average Gain over 14 days}}{\text{Average Loss over 14 days}} \tag{5.3}$$

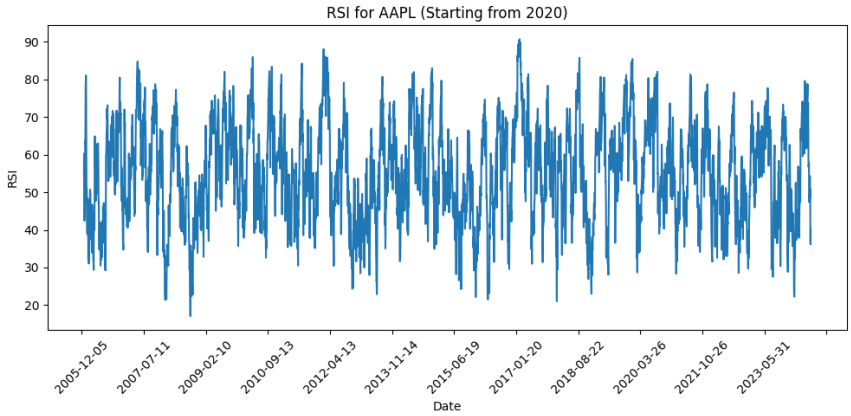


Figure 5.4: RSI of AAPL

5.2.3.2 Moving Average Convergence Divergence (MACD)

MACD is a trend-following momentum indicator that shows the relationship between two moving averages of a security's price. The role of this indicator is to identify trend changes and momentum shifts.

MACD uses the following indicators:

- **Exponential Moving Average (EMA):** A type of weighted moving average that gives more importance to recent prices.
- **Short-Term EMA ($\text{EMA}_{\text{short}}$):** Typically calculated over 12 periods, representing recent momentum.
- **Long-Term EMA (EMA_{long}):** Typically calculated over 26 periods, providing a broader view of the trend.
- **Period:** The number of data points (e.g., days) considered for the moving average calculation.

MACD uses three components that interact with each other:

- **MACD Line:** The difference between a short-term exponential moving average (EMA) and a longer-term EMA. This line reflects the momentum of price changes.

$$\text{MACD Line} = \text{EMA}_{\text{short}} - \text{EMA}_{\text{long}} \tag{5.4}$$

  Common values are 12 days for the short-term EMA and 26 days for the long-term EMA. We used 12 and 26 in this study.

- **Signal Line:** A 9-day EMA of the MACD Line. This serves as a smoother representation of the MACD Line, helping to identify potential crossovers that indicate trend shifts.

$$\text{Signal Line} = \text{EMA}_9(\text{MACD Line}) \tag{5.5}$$

- **MACD Histogram:** The difference between the MACD Line and the Signal Line. The histogram visually represents the strength and direction of momentum.

$$\text{Histogram} = \text{MACD Line} - \text{Signal Line} \tag{5.6}$$

MACD provides three ways to interpret market behavior. First, crossover signals occur when the MACD line crosses the signal line. When the MACD line crosses the signal line, a crossover signal is sent. This signal is bullish if the MACD line is crossing above the signal line, and bearish if the MACD line is below. The histogram provides insights into the strength and direction of the momentum. If the histogram is positive, i.e. the MACD line is above the signal line, the indicator suggest increasing bullish momentum. A negative histogram suggests increasing bearish momentum, with the scale of the histogram representing the magnitude of the shift. The last signal is the zero line crossing which suggests a change in trend direction. This happens when the short-term EMA is equal to the long-term EMA (see 5.4).

MACD captures both trend direction and momentum, which are essential for machine learning. It also defines actionable signals that are used to define an investment strategy called MACD crossover. It is however important to acknowledge that reliance on moving averages introduces lag to the signal generation, and is susceptible to false signals in choppy markets. By adding the three MAACD components to the dataset, we provide supplementary data for the model to understand the momentum shifts of the asset. Figure 5.5 presents the MACD of Apple from 2020 to 2023.

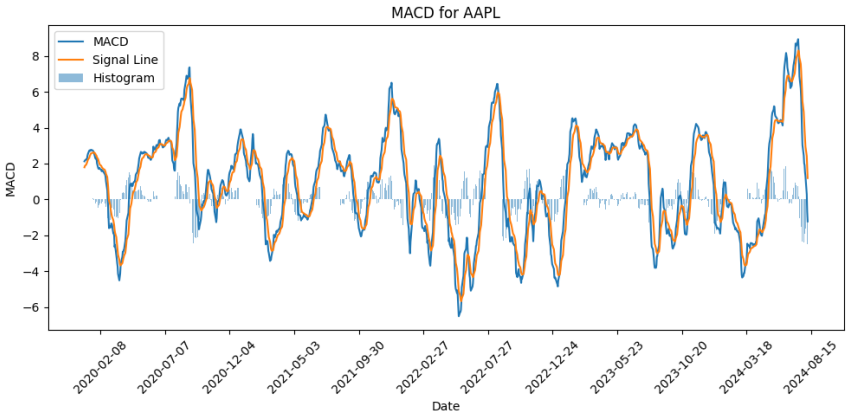


Figure 5.5: MACD of AAPL

Bollinger Bands consist of three lines: a moving average of a security's price and two standard deviation bands above and below it:

- $BBL_{5,2.0}$: Lower band (5-day moving average - 2 standard deviations).
- $BBM_{5,2.0}$: Middle band (5-day moving average).
- $BBU_{5,2.0}$: Upper band (5-day moving average + 2 standard deviations).

This captures price volatility and overbought/oversold conditions:

- Prices touching the upper band indicate overbought conditions.
- Prices touching the lower band indicate oversold conditions.

Figure 5.6 presents the Bollinger Bands of Apple from 2020 to 2023.

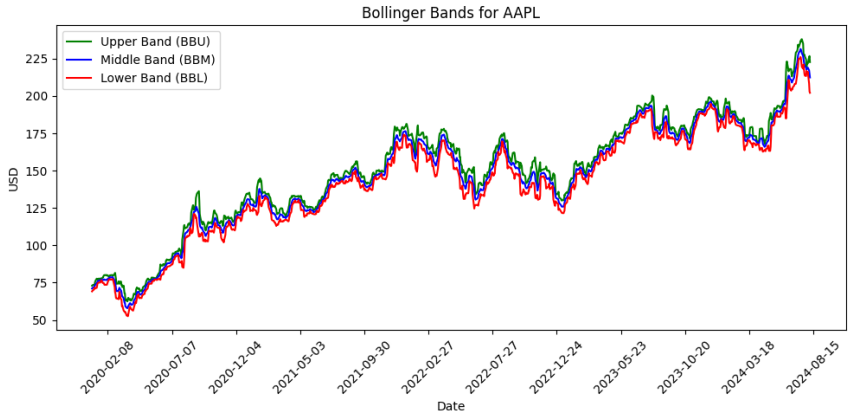


Figure 5.6: Bollinger Bands of AAPL

The augmentation of raw financial data with log returns, controlled returns, and technical indicators creates a robust feature set for predictive modeling. Each feature captures distinct aspects of market behavior, providing machine learning models with the tools to identify patterns, trends, and anomalies effectively.

5.3 Integration of ESG and SASB Materiality

In this section, we detail the implementation of the ESG metrics and SASB materiality flags in the dataset. We also provide details about the methodology used by the two data providers to calculate the ratings.

5.3.1 ESG Metrics

We integrate the ESG ratings and SASB materiality issues to the dataset. A fundamental topic of discussion in ESG ratings comes from the different methodology in rating agencies. As such, we used ratings from 3 different providers in this study:

- **Thomson Reuters**, which is **Refintiv**'s parent company. Reuters offers one of the largest ESG content collection in the market. There are over 400 ESG measures available through their API. The data is refreshed yearly in accordance to the end of the fiscal year and company disclosure. Scandals and controversies are however reflected in the scores and can induce a refresh. Reuters' coverage is extremely wide, consisting of over 7000 companies globally. The S&P500 was among the first index of companies that entered the Thomson Reuters ESG universe in 2003. Refinitv first assesses the 178 most

relevant fields. These fields are then grouped into 10 categories, which constitute the three pillar scores: Environmental, Social and Governance. Table 5.4 summarizes the categories and how they are rolled up in pillar scores. A set of 23 controversy measures are also computed to assess the controversy score for a given year. The goal of the controversy score is to discount ESG performance if a scandal occurs. Ongoing investigations or disputes are taken into account and the metrics are updated accordingly. As Reuters original business was company reporting and journalism, the coverage of scandal and controversies is exhaustive. Finally the ESG score is computed, which incorporates the Environmental, Social and Governance scores as detailed in Table 5.4. This score is then discounted depending on the controversy score, to reach the ESG Combined score, which can be equal to the ESG score should a company not be involved in any scandal that year.

Pillar	Category	Indicators in Rating	Weight
Environmental (33%)	Resource Use	19	11%
	Emissions	22	12%
	Innovation	20	11%
Social (33.5%)	Workforce	29	16%
	Human Rights	8	4.5%
	Community	14	8%
	Product Responsibility	12	7%
Governance (33.5%)	Management	34	19%
	Shareholders	12	7%
	CSR Strategy	8	4.5%
TOTAL		**178**	**100%**

Table 5.4: Pillar Categories, Indicators, and Weights

- **Sustainalytics** ESG Risk Ratings are their flagship product and what we implemented in the dataset. Their methodology is based on three building blocks. First, the Material ESG Issues are 22 criteria that are further filtered for a given subindustry. As Sustainlytics admits [138], using this building block assumes that ESG issues of a company in a given subindustry can influence the economic value in a fairly predictable way. Table 5.5 presents the material ESG issues that Sustainalytics uses. The second building block consists of an assessment of corporate and stakeholder governance. This steps ensures the interests of owners and managers align with a growth mindset, and stakeholder do not pose a reputational or financial risk to the company. In opposition to the first building block, these risks are considered independent from a company's subindustry. The third building blocks captures the Systemic and Idiosyncratic ESG issues. Systemic issues are linked to global

events, such as the raise of the sea level or global geopolitical changes. For instance, Sustainalytics offers a "Business Resilience Risk Due To Ukraine Conflict" that assesses the resilience of a business in relation to the percentage of assets owned in each of the country engaged in war. Idiosyncratic issues include company-specific events that are directly tied to a company's business model. Heavy Machinery, for instance, is statistically bound to be exposed to a human rights scandal risk. These three building blocks are then assessed through two dimensions: exposure and management. Exposure is first defined at the subindustry level, and refined at a company level using the Beta score, which is calculated using the Material ESG issues. The second dimension is management, and is defined as the set of commitments, initiatives and actions offered by a given company to tackle a given Material ESG issue. This dimension is calculated based on the involvement of companies in recent controversies and how the crisis was managed, as well as the implementation of best practices.

Material ESG Issues
Corporate Governance
Stakeholder Governance
Access to Basic Services
Business Ethics
Community Relations
Data Privacy and Cybersecurity
Emissions, Effluents and Waste
Carbon - Own Operations
Carbon - Products and Services
E&S Impact of Products and Services
Human Rights
Human Rights - Supply Chain
Human Capital
Land Use and Biodiversity
Land Use and Biodiversity - Supply Chain
Occupational Health and Safety
ESG Integration - Financials
Product Governance
Resilience
Raw Material Use
Water Use - Own Operations
Water Use - Supply Chain

Table 5.5: Material ESG Issues, Morningstar Sustainlytics

- **Sustainability Accounting Standards Board (SASB)** issues are each treated as a binary flag, with 1 being the issue recognized as material. In order to reduce the number of features, we one-hot encode each unique instance of material issues. In total, there are 68 unique combinations of material issues across all the S&P500 companies.

Discrepancy in ESG ratings is an extremely prominent topic in ESG ratings research [25], and it was essential for the credibility of this research to obtain data from different sources. The combination of data from Reuters and Sustainalytics allows for a more holistic dataset, and will serve as a point of comparison in Chapters 6 and 7. The integration of SASB materiality issues is in continuity with Chapter 2. Table 5.6 summarizes all the features available in the dataset.

Table 5.6: Complete list of features in the dataset.

Feature	Description
Open	Opening price
Low	Lowest price of the period
High	Highest price of the period
Close	Closing price
Volume	Trading volume
Log_Returns	Logarithmic returns
Controlled_Returns	Returns with control adjustments
RSI	Relative Strength Index
MACD	Moving Average Convergence Divergence
MACDs	MACD signal line
MACDh	MACD histogram
BBL_5_2.0	Lower Bollinger Band (5-day, 2.0 std)
BBM_5_2.0	Middle Bollinger Band (5-day, 2.0 std)
BBU_5_2.0	Upper Bollinger Band (5-day, 2.0 std)
Ticker	Stock ticker one-hot encoded
ESG Risk score	ESG risk score
Overall Management Score	Overall management score
Overall Exposure Score	Overall exposure score
Overall Manageable Risk Score	Overall manageable risk score
Overall Unmanageable Risk Score	Overall unmanageable risk score
Overall Managed Risk Score	Overall managed risk score
ESG Score	Overall ESG score
ESG Combined Score	Combined ESG score
ESG Controversies Score	ESG controversies score
Social Pillar Score	Social pillar score
Governance Pillar Score	Governance pillar score
Environmental Pillar Score	Environmental pillar score
SASB	SASB rating
dayWeek	Day of the week
dayMonth	Day of the month
dayYear	Day of the year

5.4 Temporality and Granularity of Data

A central issue that needs to be addressed is the fundamental difference in granularity between financial data and ESG ratings. Financial data can easily be extracted down to the minute, and varies between the span of two queries. ESG ratings, on the other hand, are refreshed annually by Reuters [9] and, ”regularly”

by Sustainalytics [10]. The scores can occasionally be refreshed more often if a scandal comes out, but the yearly financial disclosure remains the main source of information for rating agencies. The slower cycle of updates in ESG ratings is also inherent to the nature of sustainability initiatives, which are often plans involving years of transformative changes to take effect. We explored four different options to deal with this issue:

- **Regression** fits a statistical model to predict the missing ESG scores based on the financial data or the available ESG scores. It can allow for an estimation of the ESG scores during the missing periods and fill the gaps in the data with artificial ratings. It is however sensitive to overfitting and outliers, both of which can cause unrealistic data points. It also does not reflect the methodology of the rating agencies, and may also not have enough data to fit a statistical model for all companies.

- **Interpolation** estimates ESG scores based on adjacent data points. There are several interpolation algorithms, mainly linear and splines. It is computationally efficient and reduces the impact of abrupt changes that can happen in ESG data. But this last quality becomes the main disqualifying factor of this method, as the assumption that the transitions between changes in ratings are smooth and linear is extremely hard to justify. One can easily find counter examples where this assumption does not hold true, as a scandal for instance would trigger a sudden demotion in ESG ratings, or new legislation could tip a company from compliant to the sustainability requirements to below the threshold.

- **Autoencoders for imputation** are a modern way to handle missing data points. A neural network learns the pattern of the dataset and produces the missing features based on the available data for a given point in time. Repeat this process for all the dataset, and you can fill all the temporal gaps. Using a neural network allows us to capture complex and non-linear patterns in the data, and multidimensionality helps to achieve a better prediction of the missing features. However, autoencoders highly depend on the completeness of the training data to generate realistic predictions. The update cycle of ESG ratings simply does not provide enough data to train this kind of model without a high rate of error. This error would then compound within the predictive model and ultimately introduce extremely constraining assumptions for any result.

- **Forward Fill** propagates the most recent ESG score until a new score is available. This is the solution we picked, as it avoids introducing artificial trends in the dataset that could skew the predictive model. It does assume that the ESG ratings of a company remain constant until updated, which

could underestimate on-going changes in a company. Scandals are however dealt by the providers themselves, as mentioned in the previous section. Forward filling also respects the methodology of the providers the most, as a query about the ESG ratings of a given company a quarter after financial disclosure, that has not been affiliated to any scandal meaningful enough to warrant a change, would return the same value as the start of the financial year. It is worth noting that we picked forward fill, and not backward fill, as benefiting from the hindsight of future ratings was not realistic.

The necessity for more granular data foreshadows a recurring problem with ESG ratings. The abundance of financial data is in stark contrast with the scarcity of ESG ratings. While sustainability is a slow-moving target, a higher refresh rate based on external factors would be beneficial for studies such as this thesis. We found that although research on ESG ratings has been flourishing in the past decade [139], there have been no definitive solution in the literature to handle data scarcity. As such, we reviewed the most sensible options and picked forward fill as the most respectful of the methodology used by the providers, and the only option that did not introduce artificial data (regression, autoencoders), assume trends in the data (interpolation) or benefited from hindsight (backward fill).

Chapter 6

NSiTransformer in Financial Predictions with ESG

This chapter introduces the application of the NSiTransformer developed in Chapter 3 using the dataset detailed in Chapter 5. This analysis compares the performance of the NSiTransformer with other state-of-the-art models and harnesses interpretability techniques for insights into the dataset.

6.1 Introduction

The finance sector is familiar with the implementation of emerging technologies to gain an edge over the rest of the market. High frequency trading (HFT) systems for example leveraged progress in computation and lower latency to edge the market through sheer speed. Blockchain technology caught worldwide attention by proposing decentralized transactions and tokenized assets. Machine learning and artificial intelligence are no exceptions, and have been implemented in finance as far back as the 1980s with projects such as the Fifth Generation Computer System in Japan [140]. In more recent years, artificial intelligence (AI) has seen a widespread adoption in virtually every domain. The integration of AI models and agents covers a broad number of use cases, including but not limited to: invoice generation, customer service, automated trading, portfolio balancing. Time-series prediction is one of the most direct application that consists of training a model to learn past patterns in the data to infer future values based on a given number of features. The Non-stationary inverted Transformer (NSiTransformer) positions itself ideally for this task, as financial data can often be noisy and non-stationary.

Environmental, Social, and Governance (ESG) ratings have been at the center of modern investment strategies, as new regulations are increasing the pressure on companies for sustainable long-term strategies. Their integration in predictive

frameworks has been so far focused as a predicted variable, especially using natural language processing to measure the sentiment toward a company. Through the lens of interpretability, using the methods developed in previous chapters, we integrate the ESG ratings with key financial data and indicators to create a competitive model that also reveals insight about the predictive power of ESG ratings. Using the relevance maps, de-stationary factors and Shapley values [37] we can estimate if the integration of ESG ratings is beneficial for the prediction, and to what scale. We also compare the performance to a model that does not integrate any extra-financial data to include a baseline.

The research questions developed in this chapter are: Can the NSiTransformer provide accurate market prediction based on the dataset developed in Chapter 5? Are ESG ratings beneficial for the predictive power of the model? Can interpretability highlight the role of the financial and extra financial parts of the dataset?

This chapter is structured as follows: Section 6.2 presents a literature review of financial and ESG time-series forecasting, Section 6.3 develops the methodology used to conduct the experiments, Section 6.4 evaluates the model in an exclusively financial dataset and sets the baseline, Section 6.5 compares the results when integrating the ESG ratings to the dataset, Section 6.7 uses the interpretability techniques developed in previous chapters to harness insights from the results, and Section 6.8 is the conclusion of this chapter.

6.2 Literature Review

Machine learning based approaches for financial timeseries forecasting have been extensively studied [141]. Artificial neural networks are the most dominant machine learning technique in this field. Support Vector Machines (SVM) also emerges as a popular algorithm, both in predicting future direction of stock price index [142] and future contracts evaluation [143]. More recent applications have been harnessing popular deep learning methods with great success [144]: Recurrent Neural Networks [145] [146] [147], Convolutional Neural Networks [148] [149] [150], Long-Short Term Memory Neural Networks [151] [152] [153] and Deep Reinforcement Learning [154] [155] [156]. As mentioned in Section 3.2, the transformer architecture [29] has become the forefront of deep learning research. In finance, transformers have mostly been used for natural language processing and sentiment analysis using BERT models [157] [158] [159]. Other studies model stock volatility using modified transformer layers to improve forecasting models [160], use iterative dropout tests and batch size optimization [161], or harness multiplexed attention to increase the inference speed of the model [162].

ESG ratings have been integrated in machine learning and deep learning models in diverse studies in the literature [163]. On one hand, studies attempted to

predict the ESG performance of a given company using random forests [164] [165], deep neural networks [166] [167], regression [168], and ensemble methods [169]. On the other hand, studies have tried to evaluate financial data based on the ESG ratings using natural language processing for volatility prediction [170], deep learning with ESG and technical indicators [171], and ensemble methods [172] [173]. The main application consists of processing ESG news pipeline in data suitable for machine learning, effectively circumventing the need for data providers. The use of transformers was centered around natural language processing [174] and sentiment analysis [175]. To the best of our knowledge, no transformer-based timeseries prediction model uses ESG ratings as a feature.

6.3 Methodology

This section presents the methodology used to evaluate the NSiTransformer. Details on the components of the model are available in Chapter 3.3.

6.3.1 Predicted Variables

We use three predicted variables, each with different properties:

- **Close price** is a standard equity metric and serves as a baseline for prediction. In the sector analysis, we prefer other metrics as the capitalization of a given company implies large scale discrepancies.
- **Log returns** are symmetric, additive and scale independent. This metric will be particularly useful in sector analysis when data of companies of various size will be used.
- **Controlled log returns** are an alternative to log returns that remove a portion of the market influence to focus on the individual company performance. This metric has the benefits of symmetry, additivity and scale independence of log returns, but aims to remove the market noise from the log returns, which can improve performance in both individual company and sector analysis.

6.3.2 Walk-Forward Time-Series Evaluation

One of the first challenge we faced when evaluating the models was the large discrepancy between older and the latest market data. In a classic timeseries prediction task, a proportion of the dataset (usually 70%) is used for training, another one for validation (10%) and the rest becomes unseen testing data (20%). This division respects the temporality of the timeseries, meaning that the 30% dedicated to validation and testing are the latest observations in the timeseries.

This is a fundamental problem in financial data, as significant events such as the Covid-19 pandemic or the rise of artificial intelligence fall into the validation or testing dataset, leading to worse performances in choppy markets. The first solution that was developed to counter this issue was to randomly sample the segments of data in the training, validation and testing datasets [176]. This solution improved the performance of the model considerably but posed a major problem when the sequence and prediction length grew, as either data from multiple segments would leak onto each other and bias the model, or the safeguards put in place would significantly reduce the number of samples available. In order to conserve a high number of samples and take into account more recent data, we decided to put in place a rolling timeseries evaluation mechanism in the training loop [177].

Consider a timeseries:

$$\{x_t\}_{t=1}^{T},$$

where T is the total number of time steps.

Let $L_{\text{train}}, L_{\text{val}}, L_{\text{test}}$ denote the (fixed) lengths of the training, validation, and test windows, respectively. We also define a *step size* Δ by which we will slide the window for each new fold. For fold $k = 1, 2, \dots, K$, we define $s_k$as start index of fold k. The training, validation, and test windows for fold k are given by:

$$W_{\text{train}}^{(k)} = \{\, t \mid s_k \le t < s_k + L_{\text{train}} \},$$

$$W_{\text{val}}^{(k)} = \{\, t \mid s_k + L_{\text{train}} \le t < s_k + L_{\text{train}} + L_{\text{val}} \},$$

$$W_{\text{test}}^{(k)} = \{\, t \mid s_k + L_{\text{train}} + L_{\text{val}} \le t < s_k + L_{\text{train}} + L_{\text{val}} + L_{\text{test}} \}.$$

After processing fold k, we shift the start index by Δ, i.e. In each fold k:

1. **Training:** Fit the model using data $\{x_t \mid t \in W_{\text{train}}^{(k)}\}$.
2. **Validation:** Tune hyperparameters or perform early stopping using $\{x_t \mid t \in W_{\text{val}}^{(k)}\}$.
3. **Testing:** Evaluate final performance on $\{x_t \mid t \in W_{\text{test}}^{(k)}\}$.

We then collect the test metrics across folds $k = 1, \dots, K$ to obtain an estimate of the model's performance under various temporal regimes:

$$\text{Score} = \frac{1}{K} \sum_{k=1}^{K} \text{Metric}\big(\text{predictions on } W_{\text{test}}^{(k)},\ \text{ground truth on } W_{\text{test}}^{(k)}\big).$$

6.3.3 Integration of multiple stocks

Another benefit of the Walk-Forward method is the capacity to bundle several stocks in the training sets . Let us assume we have M distinct stocks, each represented by a timeseries $\{x_t^{(m)}\}_{t=1}^{T_m}$ for $m = 1, 2, \ldots, M$. When performing a walk-forward evaluation across multiple stocks, we examined two main options to integrate multiple stocks:

1. **Combine as Features.** Concatenate or merge each stock's measurements into a single multivariate timeseries, effectively treating the stock identifier m or its price series $\{x_t^{(m)}\}$ as additional features. Concretely, we form a single dataset $\left\{\left(\mathbf{x}_t, y_t\right)\right\}_{t=1}^{T}$, where $\mathbf{x}_t$ may include features from all M stocks at time t (e.g., price, volume), and y_t is the target variable (e.g., a future return) for one or multiple stocks. The walk-forward splits (*training*, *validation*, *test*) then proceed as described in Section X, but now on the merged dataset.

2. **Per-Stock Walk-Forward.** If each stock is to be modeled independently, we start by adding to the dataset a one-hot encoded column representing the stocks' ticker. We then perform the walk-forward procedure on each stock's timeseries separately. For stock m, we define $W_{\text{train}}^{(k,m)}, W_{\text{val}}^{(k,m)}, W_{\text{test}}^{(k,m)}$ as the respective train, validation, and test windows for fold k. We repeat the rolling-window approach from $k = 1$ to K for each stock m, creating $M \times K$ sets of evaluation results. A final performance measure can be computed by averaging (or otherwise aggregating) across all stocks and all folds:

$$\text{Score} = \frac{1}{MK} \sum_{m=1}^{M} \sum_{k=1}^{K} \text{Metric}\Big(\hat{y}_t^{(k,m)},\ y_t^{(k,m)};\ t \in W_{\text{test}}^{(k,m)}\Big).$$

   Alternatively, the model can be tested on any given stock's test segment for performance evaluation or comparison with a single stock model.

Table 6.1: Comparison between Walk-Forward methods on {AAPL, MSFT}, predicting AAPL

Metric	MSE	MAE
Combine as features		
Close price	16.72	2.28
Log returns	16.65	2.27
Controlled Log Returns	16.73	2.28
Per-Stock Walk-Forward		
Close price	1.11	0.62
Log returns	1.18	0.68
Controlled Log Returns	1.11	0.61

Table 6.1 compares the performance of the two methods explored on a dataset combining AAPL and MSFT, predicting AAPL. The Per-stock Walk-Forward method outperforms the other option, and was retained as the main method throughout the rest of this chapter.

6.3.4 Benchmark Models

We harness the Time-Series-Library [102] and propose seven of the best performing models as our benchmark: iTransformer [78], PatchTST [91], Crossformer [90], TimesNet [103], DLinear [77], NSTransformer [92].

6.4 Predictive Performance on Financial Timeseries

We start by evaluating the performance of the NSiTransformer on the financial dataset, and compare the results to state-of-the-art benchmark models. The preliminary results are then used as a baseline referenced when introducing the ESG-enhanced timeseries in Section 6.5.

6.4.1 Individual Stocks

We start by training several models on a single stock to single out which predicted variable performs best and hyperparameters tuning without the heavy computational cost of training on multiple stocks. In each Sector, we selected two benchmark tickers and predicted the Close Price, Log Returns and Controlled Returns at three different time horizons: day-after prediction, 7 days after, and 14 days after. We trained the model with progressively deeper networks, as shown in Figure 6.1 and 6.2 on AAPL. We use depths of the power of 2 to fully take advantage of bit-wise operations and AVX-512 computing [178]. We want to use the depths that yields the lowest MSE and MAE while also minimizing the computation overhead.

Depths from 16 to 256 have similar results, with a small gain in performance at 32 hidden layers. At 512, the model massively overfits and the MSE/MAE increase dramatically. Consequently, the results of this preliminary depth tuning indicate that the optimal dimension of the model is 32, with a good compromise between fitting the model and reasonable computational overhead.

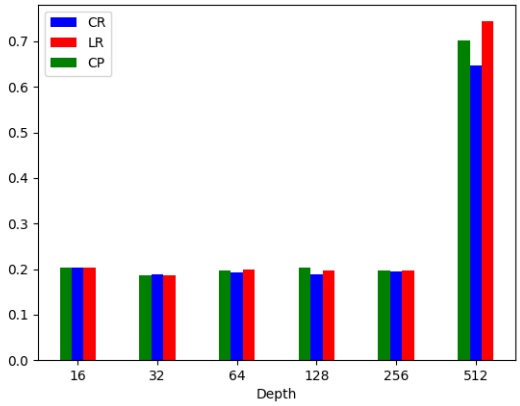


Figure 6.1: MSE per depth and per predicted variable on AAPL (Lower is better).

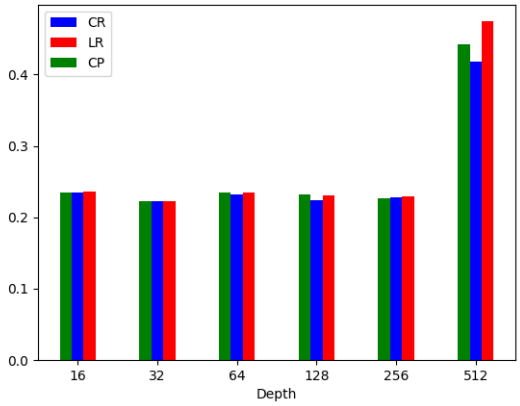


Figure 6.2: MAE per depth and per predicted variable on AAPL (Lower is better).

Table 6.2 presents the results of this experiment. The Mean Squared Error and

Mean Average Error presented in this table represents the mean of the metrics obtained when testing on the two Walk-forward data segments. All three metrics are close in performance in these examples, and we will be focusing on predicting controlled returns in the next sections. Controlled returns present the advantage of removing the influence of the market from the data and reducing the auto-correlation of the dataset, which can be beneficial once trained on a large number of stocks.

Table 6.2: Full results for the long-term forecasting task. The input sequence length is set to 96 for all baselines, and P is the prediction length. Avg is the average result of all four prediction lengths. MSE stands for Mean Squared Error and MAE for Mean Absolute Error.

Models	P	Close Price		Log Returns		Controlled Returns	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
Morgan Stanley (MSCI)	1	.188	0.219	0.188	0.220	0.188	0.220
	7	0.336	0.319	0.335	0.319	0.334	0.319
	14	0.482	0.398	0.492	0.403	0.482	0.398
GE Aerospace (GE)	1	0.099	0.151	0.100	0.153	0.100	0.151
	7	0.152	0.208	0.154	0.21	0.152	0.209
	14	0.208	0.266	0.214	0.264	0.209	0.259
Apple (AAPL)	1	0.185	0.222	0.186	0.223	0.187	0.222
	7	0.424	0.347	0.429	0.350	0.429	0.349
	14	0.668	0.455	0.671	0.458	0.670	0.456

Table 6.5 summarizes the long-term forecasting results for three representative stocks (MSCI, GE, and AAPL) across three prediction horizons (P=1, 7, and 14). The table compares the NSiTransformer against the five benchmark models: iTransformer, PatchTST, DLinear and Non-Stationary Transformer (Stationary in the table). The input sequence is fixed at 96. The NSiTransformer consistently outperforms the iTransformer, notable in MSCI at $T = 1$ where the NSiTransformer's MSE of 0.188 outclasses the 0.323 from the iTransformer. At higher prediction length, the iTransformer catches up, with similar MSE and MAE or outright beating the NSiTransformer, for instance in AAPL at $P = 7$. Out of all the benchmark models, PatchTST performs the best and exhibits the lowest error metrics, especially for short-term predictions. For instance, PatchTST attains an MSE of 0.085 for GE at $P = 1$. PatchTST uses patching to batch together similar trends in a given timeseries, an approach that has shown to be incredibly efficient. The performance gap appears to narrow as the prediction horizon increases. The orig-

inal Stationary performed extremely well in this experiment, beating the inverted transformers in at least one prediction length for each stock. While the NSiTransformer does not always demonstrate the lowest errors, the model performs either at state-of-the-art or near state-of-the-art level.

Table 6.3: Full results for the long-term forecasting task. The input sequence length is set to 96 for all baselines, and P is the prediction length. Avg is the average result of all four prediction lengths. MSE stands for Mean Squared Error and MAE for Mean Absolute Error.

Models	P	NSiTransformer		iTransformer		PatchTST		TimesNet		DLinear		Stationary	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
	1	0.188	0.223	0.323	0.316	**0.150**	**0.185**	0.172	0.218	0.219	0.265	0.185	0.232
	7	0.340	0.321	0.323	0.316	**0.290**	**0.293**	0.315	0.327	0.392	0.382	0.342	0.340
MSCI	14	0.486	0.399	0.466	0.394	**0.442**	**0.378**	0.421	0.387	0.570	0.479	0.436	0.400
	1	0.100	0.151	0.099	0.151	**0.085**	**0.138**	0.100	0.163	0.108	0.176	0.097	0.166
	7	0.155	0.210	0.153	0.211	**0.144**	**0.207**	0.171	0.232	0.168	0.244	0.174	0.241
GE	14	0.213	0.262	0.214	0.261	**0.199**	**0.251**	0.256	0.291	0.227	0.294	0.220	0.291
	1	0.186	0.222	0.192	0.227	**0.146**	**0.193**	0.173	0.226	0.216	0.255	0.174	0.227
	7	0.427	0.348	0.415	0.343	**0.365**	**0.317**	0.384	0.341	0.463	0.388	0.381	0.334
AAPL	14	0.660	0.441	0.658	0.442	**0.602**	**0.413**	0.552	0.409	0.728	0.526	0.563	0.413

6.4.2 Sectors

Using the walk forward, we train the models on all stocks for a given sector and compare the MSE of the fully trained model to the intermediary results. Table **??** presents the results of the forecasting task for three distinct sectors (Finance, Industrials, and Technology) using the walk-forward approach. The MSE and MAE shown are the average of testing results across all stocks **after** the model has completed pre-training. In the Finance and Industrials sectors, the NSiTransformer and its competitors achieve similar error levels. For instance at $P = 1$ in the Finance sector, the top three models are within ± 0.002 of MSE suggesting that the models all efficiently capture the dynamics present in the dataset. DLinear and Stationary on the other hand perform significantly worse than on a single stock compared to the other models. Stationary remains competitive at $P = 1$, but experiences a massive drop in performance at higher prediction length, being the worst performer for the Industrials and Technology sectors at $P = 14$ by a significant margin. A clear pattern that emerges is the uneven difficulty of sectors: Finance and Industrials appear to be significantly easier than Technology, especially at higher prediction length. This could indicate that this sector does not benefit from pre-training as much as the two others, or that predicting the erratic movement of Tech stocks at longer horizons is particularly difficult. But neither of these explanations align with the APPL results in Table 6.2, that are higher than GE or MSCI but not by the orders of magnitude seen in the pre-trained results. We found that the median MSE for the NSiTransformer at $P = 14$ was 0.407, indicating very strong outliers on the tail end of the dataset. Two main outliers were identified: Nvidia's (NVDA) 1,789.12% growth over the past five years was especially hard to predict by all the models, with MSEs around 8.001 (NSiTransformer) or 10.53 (TimesNet), but the biggest culprit was Super Micro Computer Inc (SMCI), which hit an all time high four months before the dataset cut-off, as shown on Figure 6.3.

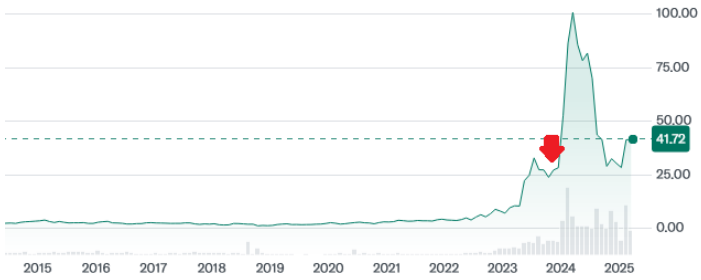


Figure 6.3: Share price of Super Micro Computer Inc over time. The red arrow shows the cut-off of the dataset.

Table 6.4: Results of benchmark models on Super Micro Computer Inc. On day-after prediction, the model maintains a decent understanding of the upward pattern, but when the prediction length rises, the MSE increases drastically.

Models	P	SMCI	
Metric		MSE	MAE
NSiTransformer	1	23.74	2.17
	7	145.20	5.19
	14	257.84	7.14
iTransformer	1	23.83	2.20
	7	**118.75**	**4.73**
	14	**254.88**	7.17
PatchTST	1	**23.39**	**0.19**
	7	125.50	4.81
	14	271.90	**7.06**
TimesNet	1	25.33	2.35
	7	120.40	4.77
	14	307.52	7.41
DLinear	1	65.74	3.96
	7	208.36	6.71
	14	387.67	9.38
Stationary	1	31.82	2.55
	7	492.87	9.17
	14	520.49	10.35

Table 6.4 presents the MSE and MAE of all models when predicting SMCI. The MSEs and MAEs are exceedingly high and too polarizing to be included and as such SMCI will be excluded in the next tables referencing the Technology sector.

Table 6.5: Full results for the long-term forecasting task. The input sequence length is set to 96 for all baselines, and P is the prediction length. Avg is the average result of all four prediction lengths. MSE stands for Mean Squared Error and MAE for Mean Absolute Error.

Models	P	NSiTransformer		iTransformer		PatchTST		TimesNet		DLinear		Stationary	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
MSCI	1	0.188	0.223	0.323	0.316	**0.150**	**0.185**	0.172	0.218	0.219	0.265	0.185	0.232
	7	0.340	0.321	0.323	0.316	**0.290**	**0.293**	0.315	0.327	0.392	0.382	0.342	0.340
	14	0.486	0.399	0.466	0.394	0.442	**0.378**	**0.421**	0.387	0.570	0.479	0.436	0.400
GE	1	0.100	0.151	0.099	0.151	**0.085**	**0.138**	0.100	0.163	0.108	0.176	0.097	0.166
	7	0.155	0.210	0.153	0.211	**0.144**	**0.207**	0.171	0.232	0.168	0.244	0.174	0.241
	14	0.213	0.262	0.214	0.261	**0.199**	**0.251**	0.256	0.291	0.227	0.294	0.220	0.291
AAPL	1	0.186	0.222	0.192	0.227	**0.146**	**0.193**	0.173	0.226	0.216	0.255	0.174	0.227
	7	0.427	0.348	0.415	0.343	**0.365**	**0.317**	0.384	0.341	0.463	0.388	0.381	0.334
	14	0.660	0.441	0.658	0.442	0.602	0.413	**0.552**	**0.409**	0.728	0.526	0.563	0.413

6.5 Predictive Performance on ESG-Enhanced Timeseries

Based on the performance of the NSiTransformer developed in 6.4 on the exclusively financial dataset, we add the ESG data from Sustainalytics and Reuters. We determine that adding ESG data, regardless of the provider, improves the predictive performance of the models. Table 3.1 presents the number of ticker at each granularity levels of the dataset.

Table 6.6

Granularity	Financial	Industrial	Technology	Total
Financial	71	77	68	216
Sustainalytics	62	72	63	197
Reuters	67	76	65	208

6.5.1 Sustainalytics

While including ESG ratings shrinks the dataset size, the results reveal clear patterns in how different models handle this expanded set of features across three sectors: Finance, Industrials, and Technology. Table 6.7 shows how adding Sustainalytics ESG data affects forecasting performance across financial models.

The NSiTransformer performs strongest in the Finance sector, with MSE scores of 0.253 ($P = 1$), 0.384 ($P = 7$), and 0.512 ($P = 14$). Both iTransformer and PatchTST deliver nearly identical results, while TimesNet and DLinear show slightly higher errors. The Stationary model keeps pace on short-term forecasts but loses accuracy as predictions extend beyond a week.

All models show higher error rates on the Industrials sector compared to Finance. The NSiTransformer achieves MSEs of 1.18 ($P = 1$), 1.47 ($P = 7$), and 1.86 ($P = 14$). While the NSiTransformer, iTransformer, PatchTST and TimesNet cluster closely in performance, the Stationary model struggles, as its MSE jumps to 3.90 for $P = 14$ forecasts. The DLinear model places itself between the leading pack and Stationary, getting close to the top performer in Finace at $P = 1$ but never beating out any of the top 4 models. This pattern remains in the Technology sector, that appears to be simpler to predict at $P = 1$ but sharply increasing in difficulty as P rises.

Incorporating Sustainalytics ESG data creates a mixed picture: The initial top models like NSiTransformer, iTransformer, PatchTST and TimesNet maintain strong performance despite a smaller datasets, while DLinear and Stationary suffer, especially at higher P. These results indicate a predominant problem in using ESG ratings: depending on the provider, the history of data available might be more beneficial than the added dimensions.

Table 6.7: Full results for the long-term forecasting task. The input sequence length is set to 96 for all baselines, and T is the prediction length. MSE stands for Mean Squared Error and MAE for Mean Absolute Error.

Models	T	NSiTransformer		iTransformer		PatchTST		TimesNet		DLinear		Stationary	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Finance	1	**0.253**	0.158	0.268	0.158	0.272	**0.157**	0.267	0.186	0.303	0.200	0.260	0.175
	7	**0.384**	0.245	0.382	0.245	0.405	**0.240**	0.404	0.271	0.429	0.274	0.435	0.290
	14	**0.512**	**0.312**	0.526	0.324	0.524	0.333	0.549	0.346	0.580	0.382	0.610	0.377
Industrials	1	1.18	0.185	1.21	0.185	**1.13**	**0.180**	1.21	0.216	1.28	0.230	1.56	0.213
	7	1.47	0.281	1.47	0.294	**1.40**	**0.277**	1.56	0.332	1.62	0.355	1.73	0.392
	14	1.86	0.375	1.98	0.375	**1.76**	**0.349**	2.06	0.400	2.22	0.505	3.90	0.496
Tech (No SMCI)	1	0.427	0.194	0.399	0.187	**0.398**	**0.183**	0.566	0.220	0.629	0.257	1.021	0.240
	7	1.240	0.306	1.222	0.306	**1.196**	**0.296**	1.600	0.360	1.544	0.382	1.989	0.350
	14	**2.010**	0.398	2.234	0.405	2.016	**0.384**	3.018	0.482	2.505	0.451	5.610	0.610

6.5.2 Reuters

Table 6.8 compares long-term forecasting performance using Reuters ESG ratings alongside financial data. Unlike the earlier Sustainalytics analysis, Reuters' ESG integration causes less severe data reduction as this smaller trimming of the dataset helps preserve more training examples while still adding sustainability metrics.

In Finance, NSiTransformer, iTransformer, and PatchTST dominate short-term predictions: their 1-day forecasts show nearly identical MSE scores (0.151–0.152) and MAE values (0.148–0.153). TimesNet performs slightly worse, only beating the NSiTransformer on Industrials at $P = 7$ and $P = 14$. While DLinear trails with a 0.265 MSE at this horizon, the Stationary model holds its own initially. Predictably, errors grow for all models as forecasts stretch to 7 and 14 days.

The Industrials sector tells a similar story. All models start strong with 1-day MSEs between 0.159–0.181. As predictions extend to $P = 7$, the Stationary model surprisingly closes the gap slightly at the $P = 14$, despite the previous trend of massive compounding error in longer time predictions.

In Technology, the first four leading models achieve $P = 1$ MSEs of 0.201–0.233, matching the Stationary model's short-term performance. But beyond a week, their lead widens dramatically: by the $P = 14$ horizon, the top models clearly outpace both DLinear and Stationary models. This divergence implies these newer architectures handle Technology inherent instability better.

Reuters' ESG integration appears to support steadier performance across sectors and timeframes compared to Sustainalytics. The larger quantity of available training data is a definite benefiting factor, which also makes the results between the two models incomparable in this state since the test segments differ.

Table 6.8: Full results for the long-term forecasting task. The input sequence length is set to 96 for all baselines, and T is the prediction length. (All numbers are rounded to 3 s.f.) MSE stands for Mean Squared Error and MAE for Mean Absolute Error.

Models	P	NSiTransformer		iTransformer		PatchTST		TimesNet		DLinear		Stationary	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Finance	1	0.151	**0.148**	**0.151**	0.153	0.152	0.153	0.163	0.168	0.265	0.233	**0.149**	0.154
	7	0.458	0.780	0.439	0.274	**0.425**	**0.267**	0.519	0.302	0.655	0.392	0.532	0.303
	14	0.780	0.365	0.779	0.365	**0.773**	**0.361**	0.918	0.410	0.995	0.437	0.959	0.409
Industrials	1	0.168	0.168	0.162	0.161	**0.159**	**0.159**	0.181	0.187	0.291	0.242	0.171	0.168
	7	0.506	0.305	0.481	0.300	**0.455**	**0.290**	0.484	0.303	0.657	0.371	0.566	0.313
	14	0.895	0.411	0.813	0.403	**0.811**	**0.392**	0.878	0.419	1.03	0.479	0.927	0.421
Tech (No SMCI)	1	**0.201**	0.179	0.202	0.181	0.212	0.191	0.233	0.202	0.510	0.297	0.226	0.191
	7	1.004	0.369	0.908	0.355	**0.911**	**0.355**	0.943	0.363	1.403	0.453	1.212	0.372
	14	1.69	0.482	**1.663**	0.485	1.732	**0.480**	1.762	0.497	2.206	0.563	2.732	0.594

6.6 Towards more comparable metrics

Although the results are pointing towards ESG ratings improving the performance of the model, directly comparing the metrics of the exclusively financial and ESG-enhanced models is not possible. In order to provide this comparison, we narrow down the datasets to the largest comparable subset through a temporal and stock cut-off.

6.6.1 Temporal cut-off

Table 6.9 compares forecasting performance across ESG datasets (financial-only, Sustainalytics, and Reuters) using identical temporal ranges. The evaluation spans across the three sectors: Finance, Industrials, and Technology at 1-, 7-, and 14-day prediction horizons ($P = 1, 7, 14$). Three sector-specific patterns emerge when controlling for temporal coverage.

Finance demonstrates consistent ESG benefits. At P=1, the financial-only baseline achieves an MSE of 0.331, while Sustainalytics (0.253) and Reuters (0.238) show progressive improvement. This advantage holds across extended horizons: Reuters maintains the lowest errors (MSE/MAE) at P=7 and P=14, outperforming both Sustainalytics and the ESG-free dataset.

Industrials reveals stark provider divergence. The financial-only baseline starts at $P = 1$ with an MSE of 0.211, but Sustainalytics integration paradoxically increases errors (MSE=1.18), a 460% surge suggesting that the ESG metrics introduce detrimental noise to the dataset. The other possibility is that the stocks with available Sustainalytics data are especially difficult to predict, an idea that will be dealt with in the next section. Reuters achieves superior $P = 1$ performance (MSE=0.126), sustaining this lead through longer horizons.

Technology exhibits Reuters' most pronounced advantage. While the ESG-free model records a $P = 1$ MSE of 0.258, Sustainalytics degrades performance (MSE=0.427), whereas Reuters cuts errors by 43% (MSE=0.147). Horizon extensions further magnify these disparities, with Reuters' error margins remaining notably tighter at $P = 7$ and $P = 14$ compared to both alternatives.

There is however a possible explanation for the stark degradation in performance using Sustainalytics. In this experiment, all the stocks with available data for a given provider were considered, and the data cut-off was based on the latest data available. There is no guarantee that the stocks considered for No ESG, Sustainalytics and Reuters are the same. In order to provide the most directly comparable metrics, the next subsection proposes to examine the intersection of available stocks on top of the temporal cut-off.

Table 6.9: Results of forecasting of NSiTransformer for Finance, Industrials and Technology sector with a temporal cut-off in 2018 at P=1, 7 and 14.

Models	P	No ESG		Sustainalytics		Reuters	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
Finance	1	0.331	0.189	0.253	0.158	**0.238**	**0.134**
	7	0.446	0.291	0.384	0.245	**0.325**	**0.207**
	14	0.571	0.370	0.512	0.312	**0.410**	**0.264**
Industrial	1	0.211	0.210	1.18	0.185	**0.126**	**0.138**
	7	0.346	0.317	1.47	0.281	**0.216**	**0.217**
	14	0.504	0.416	1.86	0.375	**0.311**	**0.278**
Tech (No SMCI)	1	0.258	0.223	0.427	0.194	**0.147**	**0.147**
	7	0.479	0.360	1.240	0.306	**0.270**	**0.235**
	14	0.704	0.454	2.010	0.398	**0.394**	**0.299**

6.6.2 Stock intersection

To propose the fairest possible comparison between No ESG and the two providers, we experiment on the intersection of available stocks for each sectors. This process reduces the total number of stocks from 216 down to 184. Table 6.10 shows the number of stocks removed for each sector. The proportion of stocks removed is close to even for each sector.

Table 6.10: Proportion of stocks removed due to missing or incomplete data in at least one other dataset.

Sector	Pre-cut	Post-cut
Finance	71	59
Industrials	77	68
Tech	68	57
Total	216	184

Table 6.11 evaluates the NSiTransformer's performance using a consistent subset of 184 stocks (down from 216) across three ESG datasets (financial-only, Sustainalytics, and Reuters) in the Finance, Industrials, and Technology sectors (excluding SMCI). By harmonizing the sample, this comparison isolates the impact of ESG data quality on forecasting accuracy. We also train a model (Sustainalytics + Reuters, S+R in Table 6.11) using both providers' metrics, since the harmonization of the sample allows for no missing data.

The analysis reveals consistent forecasting gains when ESG data supplements financial metrics. Across all sectors and prediction windows, ESG integration reduces errors relative to financial-only baselines. In Finance, for instance, $P = 1$

forecasts improve from an MSE of 0.322 (no ESG) to 0.255 with Sustainalytics and 0.226 with Reuters, all the way down to 0.203 with S+R. This hierarchy also scales at longer horizons: at $P = 14$, S+R forecasts achieve an MSE of 0.396, outperforming Reuters (0.401) , Sustainalytics (0.463) and the baseline (0.550).

In Industrials, Reuters emerges as the superior ESG source. its $P = 1$ MSE (0.123) undercuts both S+R (0.128), Sustainalytics (0.159) and the baseline (0.175). This pattern maintains at $P = 7$ and $P = 14$, where the Reuters trained model maintains the top spot. This results echoes with Table 6.9 where Sustainalytics struggled with Industrials: there was sample bias, as shown by the dramatically lower MSE at all P, but this sector remains challenging in Sustainalytics framework.

In Technology, the same pattern as Finance emerges: No ESG serves as the baseline with the highest MSE and MAE. Individually, Sustainalytics and Reuters both help reduce the MSE, despite Reuters clearly promoting the model more, but S+R emerges as the clear top performer in all sectors, for all P. These results indicate that the contributions of each ESG provider are separated, as there is a compounding effect when used in conjunction.

The results imply Sustainalytics and Reuters capture distinct but overlapping facets of ESG risk. Their combination appears to filter out provider-specific biases: for instance, Sustainalytics' noisier Industrials metrics are counterbalanced by Reuters' cleaner signal, while Reuters' potential blind spots in Technology volatility are mitigated by Sustainalytics' complementary data. The differences in methodology exposed in 5.3 could be a determining factor to the compounding effect observed. This synergy elevates the NSiTransformer's predictive capacity, particularly in longer-horizon forecasts where the modeling challenges are amplified.

Collectively, the experiment demonstrates that ESG integration strategies transcend simple additive benefits. By merging providers, models like NSiTransformer can exploit inter-dataset correlations to dampen noise and amplify sectorally relevant signals which is a critical advantage in multi-horizon forecasting where isolated data sources may inadequately capture complex market interdependencies.

In Chapter 7, we trained the models on the complete set of stocks, using both providers. Table 7.5 presents the results of this experiment (line "Original"), which further improve upon the results of Table 6.11.

Table 6.11: Results with forecasting of NSiTransformer for Finance, Industrials and Technology sector with a temporal cut-off in 2018 and stock cut-off at P=1, 7 and 14. S+R stands for Sustainalytics+Reuters and was trained with both ESG metrics available.

Models	P	No ESG		Sustainalytics		Reuters		S+R	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Finance	1	0.322	0.189	0.255	0.163	0.226	0.136	**0.203**	**0.118**
	7	0.449	0.292	0.359	0.220	0.332	0.200	**0.317**	**0.182**
	14	0.550	0.357	0.463	0.280	0.401	0.252	**0.396**	**0.228**
Industrial	1	0.175	0.187	0.159	0.155	0.123	0.132	**0.128**	**0.121**
	7	0.311	0.305	0.268	0.243	**0.210**	0.211	0.214	**0.192**
	14	0.457	0.396	0.410	0.320	**0.311**	0.276	0.326	**0.255**
Tech (No SMCI)	1	0.207	0.207	0.164	0.150	0.146	0.142	**0.131**	**0.119**
	7	0.391	0.335	0.328	0.257	0.282	0.237	**0.260**	**0.204**
	14	0.567	0.420	0.489	0.327	0.405	0.299	**0.400**	**0.266**

6.7 Insights from Interpretability Techniques

This section harnesses the interpretability techniques developed in Section 3.4 to provide insights on the quality and relevance of ESG ratings. We evaluate the differences between data providers, and highlight the key features contributing to the prediction accross several examples.

6.7.1 Correspondence Between Tokens And Variables

In the next sections, we will be referring to the tokens used by the model to encode the features of the dataset. This section serves as a reference for the subsequent figures in subsection 6.7.2 and 6.7.3. Table 6.12 describes the relationship between tokens and features. This dataset serves as the base for the ESG-enhanced dataset, which all use the same first 14 tokens that encode the financial features. Tokens 15 to 26 are used to encode the temporal context of the time-series.

Table 6.12: Description of tokens for the encoded network without ESG ratings.

Token	Feature	Description
0	Open	Opening price
1	Low	Lowest price of the period
2	High	Highest price of the period
3	Close	Closing price
4	Volume	Trading volume
5	Log_Returns	Logarithmic returns
6	Controlled_Returns	Returns with control adjustments **(Target variable)**
7	RSI	Relative Strength Index
8	MACD	Moving Average Convergence Divergence
9	MACDs	MACD signal line
10	MACDh	MACD histogram
11	BBL_5_2.0	Lower Bollinger Band (5-day, 2.0 std)
12	BBM_5_2.0	Middle Bollinger Band (5-day, 2.0 std)
13	BBU_5_2.0	Upper Bollinger Band (5-day, 2.0 std)
14	Ticker	Stock ticker one-hot encoded
15	dayWeek	Day of the week
16	dayMonth	Day of the month
17	dayYear	Day of the year
18 - 25	date	Time2Vec Embedded Features

Table 6.13 presents the tokens corresponding to the encoding of the Sustainalytics ESG ratings. Due to the Time2Vec embedding happening after the dataset features, the last 11 tokens always are the temporal tokens.

Table 6.13: Description of tokens for the encoded network with Sustainalytics ESG ratings. Tokens 15-20 are the embedded ESG ratings from Sustainalytics.

Token	Feature	Description
0 - 14	Financial features	Same features as Table 6.12
15	ESG Risk score	ESG risk score
16	Overall Management Score	Overall management score
17	Overall Exposure Score	Overall exposure score
18	Overall Manageable Risk Score	Overall manageable risk score
19	Overall Unmanageable Risk Score	Overall unmanageable risk score
20	Overall Managed Risk Score	Overall managed risk score
21	SASB	SASB rating
22	dayWeek	Day of the week
23	dayMonth	Day of the month
24	dayYear	Day of the year
25 - 32	date	Time2Vec Embedded Feature

Table 6.14 presents the tokens corresponding to the encoding of the Reuters ESG ratings. The Reuters scores correspond more directly to the idea of ESG ratings by directly integrating the values as pillar score.

Table 6.14: Description of tokens for the encoded network with Reuters ESG ratings. Tokens 15-20 are the embedded ESG ratings from Reuters.

Token	Feature	Description
0 - 14	Financial features	Same features as Table 6.12
15	ESG Score	Overall ESG score
16	ESG Combined Score	Combined ESG score
17	ESG Controversies Score	ESG controversies score
18	Social Pillar Score	Social pillar score
19	Governance Pillar Score	Governance pillar score
20	Environmental Pillar Score	Environmental pillar score
21	SASB	SASB rating
22	dayWeek	Day of the week
23	dayMonth	Day of the month
24	dayYear	Day of the year
25 - 32	date	Time2Vec Embedded Features

Table 6.15 presents the tokens corresponding to the encoding of the ESG ratings from both providers. This experiment alongside the relevance maps and de-

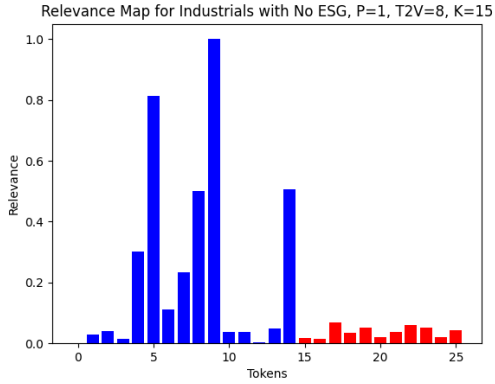
stationary factors aims at determining the influence of each token in the prediction.

Table 6.15: Description of tokens for the encoded network with both providers of ESG ratings. Tokens 15-20 are the embedded ESG ratings from Sustainalytics, tokens 21-26 are the embedded ESG ratings from Reuters

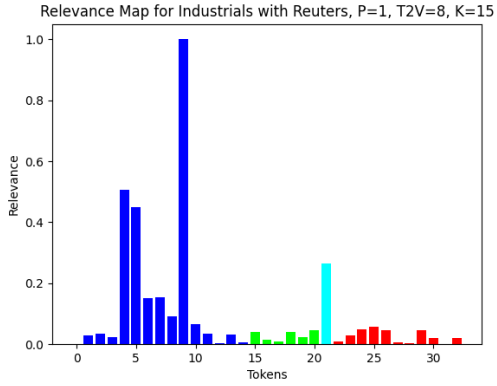
Token	Feature	Description
0 - 14	Financial features	Same features as Table 6.12
15	ESG Risk score	ESG risk score
16	Overall Management Score	Overall management score
17	Overall Exposure Score	Overall exposure score
18	Overall Manageable Risk Score	Overall manageable risk score
19	Overall Unmanageable Risk Score	Overall unmanageable risk score
20	Overall Managed Risk Score	Overall managed risk score
21	ESG Score	Overall ESG score
22	ESG Combined Score	Combined ESG score
23	ESG Controversies Score	ESG controversies score
24	Social Pillar Score	Social pillar score
25	Governance Pillar Score	Governance pillar score
26	Environmental Pillar Score	Environmental pillar score
27	SASB	SASB rating
28	dayWeek	Day of the week
29	dayMonth	Day of the month
30	dayYear	Day of the year
31 - 38	date	Time2Vec Embedded Feature

6.7.2 Relevance Maps

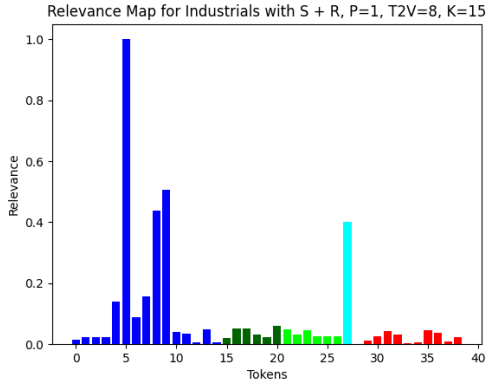
We provide supplemental interpretability by calculating the relevance of each token using Chefer et al [81] general technique adapted to the iTransformer [42].



(a)



(b)



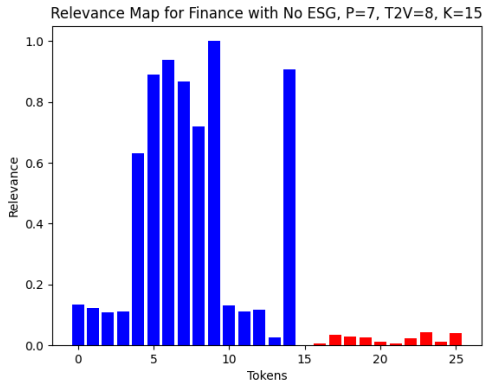
(c)

Figure 6.4: Relevance Maps for Industrials with No ESG (top), Reuters (middle) and both providers (bottom), P=1, T2V=8, K=15. In blue financial features, lime Reuters features, cyan SASB, red temporal features.

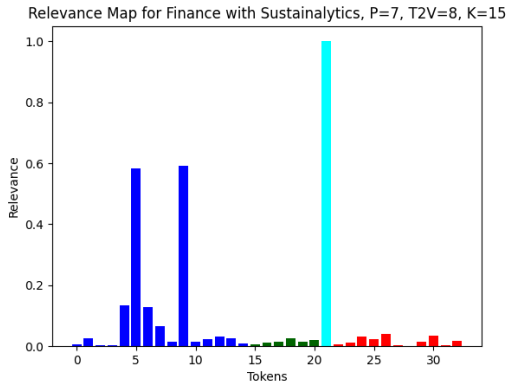
Figure 6.4 presents the relevance maps for the models trained on Industrials at $P = 1$. In the No ESG experiment, the top relevant tokens are 5, 9 and 14, as shown in 6.4a. Token 5 corresponds to the Log returns before controlling, which is surprising considering that this experiments predicts token 6, which appears to be not as relevant. A possible explaining factor is the embedding of market movement in log returns: since the model is trained on multiple stocks, the uncontrolled returns could be useful to gauge the market. Token 9 is the MACD signal line, and paired with token 8 showing a respectable relevance on its own, MACD clearly emerges as the most relevant financial indicator in the dataset. Token 14 is a special case: although it is highly relevant, this token represents the one-hot encoded ticker, and as such never changes throughout the sequence fed into the model. The most likely explanation for this relevance is the identification of a stock for the model. Although there are common patterns, when trained on all the stocks for a given sector (here Industrials) the model has to determine which stock it is predicting. In No ESG, the ticker serves as an anchor for the model to tailor the prediction to a given stock. The temporal tokens, although all slightly relevant, do not present a strong dominance between the raw data and time2vec embeddings.

When adding the Reuters ESG ratings, the relevance of the ticker plummets as shown in 6.4b. The log returns and MACD tokens remain highly relevant, token 5 being even more dominating than in No ESG. Token 4, the volume, also increases in relevance. But the most interesting observation comes from token 20, a newly introduced token that encodes the SASB score. The SASB score is an encoded number that designates which issues are recognized are material for a given company. Similarly to the ticker, this data point remains the same, and is likely to contributing to the recognition of the stock being predicted. This relevance score also means the model buckets SASB similar companies, which is in line with the findings in Table 2.5. The other Reuters tokens are also slightly relevant, especially the Environmental pillar score. Going back to Figure 2.5, Industrials was most correlated with the Environmental sector, which is coherent with what intuition would tell for a sector that includes Caterpillar (mining and construction), Union Pacific (freight hauling) or Boeing (aircraft manufacturer). This result echoes how simpler tools such as correlation can infer behaviors that will be found in far more complicated architectures such as the NSiTransformer.

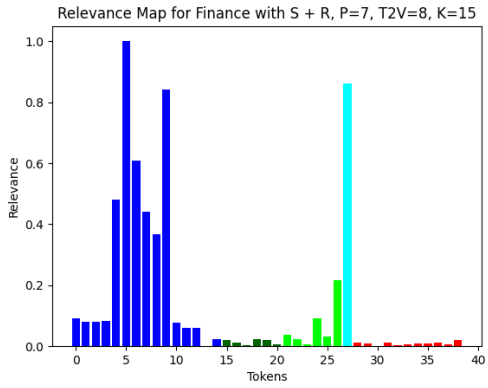
Finally, when both Sustainalytics and Reuters are added to the dataset, the patterns that emerged before are further emphasized, as shown in 6.4c. Token 14, the ticker, is no longer as relevant, and the SASB ticker maintains a high relevance. The leading Reuters score is the ESG score, while the leading Sustainalytics score is the Overall Managed Risk Score. The relevance of the metrics of both providers is in line with financial indicators such as the Bollinger Bands, and are more relevant to the model than the Open (token 0), Low (1), Close (2) or High (3).



(a)



(b)



(c)

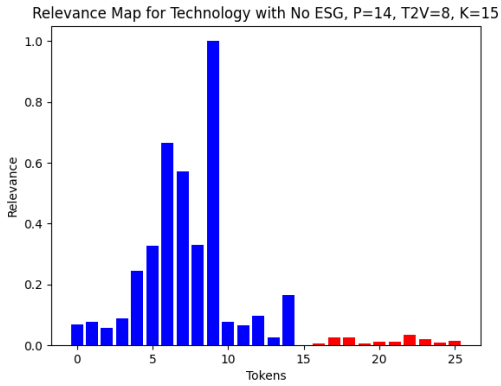
Figure 6.5: Relevance Maps for Finance with No ESG (top), Reuters (middle) and both providers (bottom), P=7, T2V=8, K=15.In blue financial features, lime Reuters features, cyan SASB, red temporal features.

Figure 6.5 presents the relevance maps for the models trained on Finance at $P = 7$. As the prediction length increase, we find similarities and discrepancies compared to the previous maps. In 6.5a (No ESG), the ticker is once again extremely relevant at first. Token 8 and 9, corresponding to the MACD and MACDs, maintain high relevance, with token 9 becoming the most relevant. The controlled returns, encoded in token 6, are however far more relevant than before. As the temporal horizon for prediction increases, it is likely that the model starts favoring historical data of the target value. As the relevance of the uncontrolled returns are still high, the hypotheses that the model uses uncontrolled returns to gauge the market still holds up.

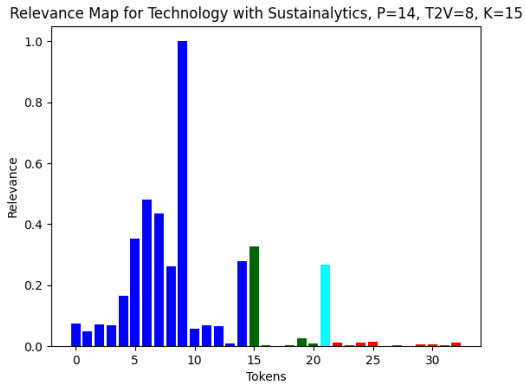
The integration of Sustainalytics to the model shows an extremely polarized model with a very high relevance for the SASB code in 6.5b. This result further strengthen the theory that the SASB issue code comes at a replacement for the ticker. The model also shows high relevance in log returns and MACDs, which is coherent with previous relevance maps. Through the relevance map, we can discern the strategy used by the model: identify which group of stock is being predicted with the SASB code, and use the historical financial data to produce an adapted result.

In 6.5c, The relevance map with both providers shows an enhanced version of the Sustainalytics integration. The SASB code clearly remains pivotal for the predictive power, but the model uses a much broader array of tokens to fuel its prediction. For instance, the historical controlled log returns in token 6 gain relevance, similarly to the other components of MACD and RSI. The Reuters ESG scores also start demonstrating sizable relevance, with the Environmental Pillar Score and Social Pillar Score displaying particularly high relevance. This broader approach could be the reason for the promotion in performance between the Sustainalytics-only and both providers model.

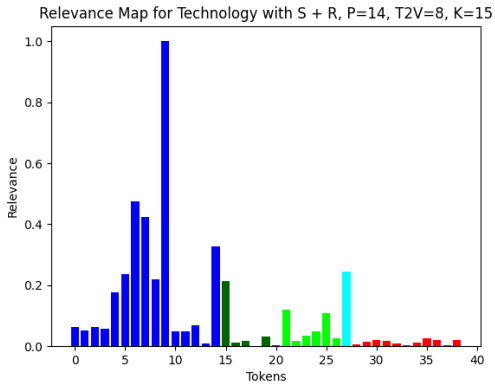
Finally, despite the model's performance being sensitive to the number of time embedded dimension, we found no significant difference in how the temporal tokens were employed in longer prediction horizons. The embedded dimensions are in all cases slightly relevant without a particular dominance. The raw data of the days, which could indicate periodicity in the data, are not particularly relevant either regardless of the prediction horizon



(a)



(b)



(c)

Figure 6.6: Relevance Maps for Technology with No ESG (top), Reuters (middle) and both providers (bottom), P=14, T2V=8, K=15. In blue financial features, lime Reuters features, cyan SASB, red temporal features.

Figure 6.6 presents the relevance maps for the models trained on Technology at $P = 14$. At this prediction length, the ticker is no longer as relevant compared to previous experiments. This could be an indication of a broader strategy that does not tailor the prediction as much as other model, but rather uses a combination of historical data and technical indicators to infer general behavior. This strategy can lead to the model underfitting, and explain why this model performed worse than the ones using ESG data. The historical data of controlled returns have taken over the log returns, but the most relevant token remains the MACD signal line (token 9). This consistent behavior throughout all experiments indicates that the models are extremely receptive to the MACD model, and this technical indicator is likely to yield improvements in other financial predictors.

The Sustainalytics model shows a nearly identical relevance structure to the No ESG model for the financial tokens, with slightly more emphasis on the ticker token. This model also displays the highest relevancy for an ESG metric out of all experiments, with a high relevance on token 15, which is the general ESG Risk score. This result is in line with the discussion from section 2.7, where the importance of ESG score in predicting power increases with the time horizon. Strategy-wise, the promotion of this model compared to No ESG could be explained by an adaptation of the general inference strategy with more targeting. The higher relevance of low periodicity and fix data such as the ESG Risk Score, ticker and SASB indicates that the model is trying to recognize which stock is the input sequence, which performs better on unseen data.

In the same fashion as the Sustainalytics model, the S + R model closely maintains the structure of No ESG for financial data in Figure 6.6c. This further reinforces the thesis that ESG ratings are beneficial for the predictive power of the model, since the structure of financial tokens remains sensibly similar. The same pattern of close to equally relevant tokens for the ticker and SASB indicates that the S + R model also adopts a more targeted strategy. Sustainalytics' ESG Risk Score is slightly less relevant in this model, but it is to the benefit of Reuters' ESG Score and Governance score. The model appears to favor general scores over specific subcategories in the majority of cases, however Governance is an intuitively highly sensitive pillar to Technology companies, as the companies within this sector are under constant scrutinization from the public due to their high profile.

Relevance maps emerge as a key tool to not only identify the most relevant features but also infer the strategy used by the model to make a prediction. Through this technique, we established how the model uses the Ticker token and the SASB code to identify companies. We also shown that the ESG ratings were relevant in the prediction, which combined with the lower MSE and MAE indicate that there is benefit to their integration in the dataset. These strategies can then be assessed based on domain expertise and results on unseen data, in order to ensure that the model is not underfitting or overfitting. This is a promising results for this technique, as it is applicable to a number of transformers-based models and used for

monitoring how a model learns.

6.7.3 De-stationary Factors

We sample the tensors of τ and δ during testing and represent it as heatmaps. The de-stationary factors represent the evolving relationship between the features. We used a top-k of 15, meaning that de-stationary attention is applied to the top 15 features. This visualization allows us to gain insights on how the de-stationary factors from the NSiTransformer are used within the model to represent the changing relationships between the features.

Figure 6.7 presents the de-stationary factors for the model trained on Industrials at $P = 1$, with both providers and only Reuters. Horizontally, the tokens are ordered according to the tables in subsection 6.7.1. Vertically, the tokens are the top 15 with the highest attention at the time of the sampling. In Reuters, a clear horizontal pattern emerges alongside tokens 13 to 20, with both tau and delta being equal for those tokens regarding the top 15 tokens. A possible explanation for this result is the low periodicity of these tokens, which also often change at the same timestamps when the ratings are updated. This idea is further backed up by the S + P model, which demonstrates the same behavior with Sustainalytics and Reuters tokens. Another observation is the relationship between the de-stationary factors and the relevance maps. It appears that the tokens that are highly relevant in previous figures often present lower de-stationary factors. For instance, on the S+R model, token 5 is by far the most relevant on Figure 6.5, but has some of the lowest tau and delta. Token 9 and 27 also show the same behavior, while the ticker (token 14) displays the opposite: extremely low relevance but high de-stationary factors.

Figure 6.8 presents the de-stationary factors for the model trained on Finance at $P = 7$, and the inverted relationship between relevance and de-stationary factors persists. In the S+R model in particular, we can clearly see the groups of tokens 0 to 4 with low relevancy and high factors, as opposed to tokens 5 to 9 with high relevancy and low factors. There are however counter examples, for instance with token 9 in No ESG maintaining both high relevancy and high factors. A likely explanation is that since the de-stationary factors are learned, the model could be compensating high relevancy with low factors to balance out the prediction and avoid overfitting.

Visualizing the de-stationary factors allows us to gain insight about the model and how it tries to balance the evolving relationship between features. While the vertical lecture of the maps are imperfect when using a k inferior to the number of features, the horizontal lecture still gives insight as to which token are given increased or decreased attention at a given time. In this case, the tendency of de-stationary factors to counterbalance the relevance and the low periodicity of

data having an influence on their attention are important signs that the model can generalize well on unseen data.

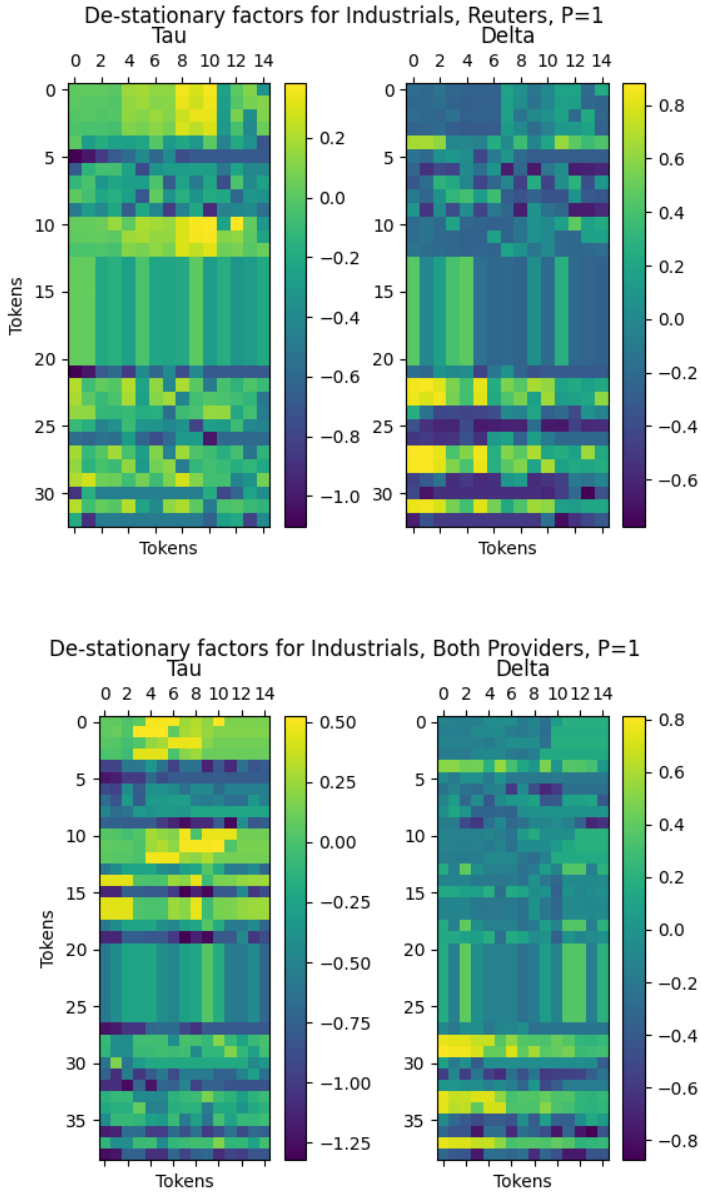


Figure 6.7: De-stationary factors for Industrials, P=1, S + R (top), Reuters (bottom).

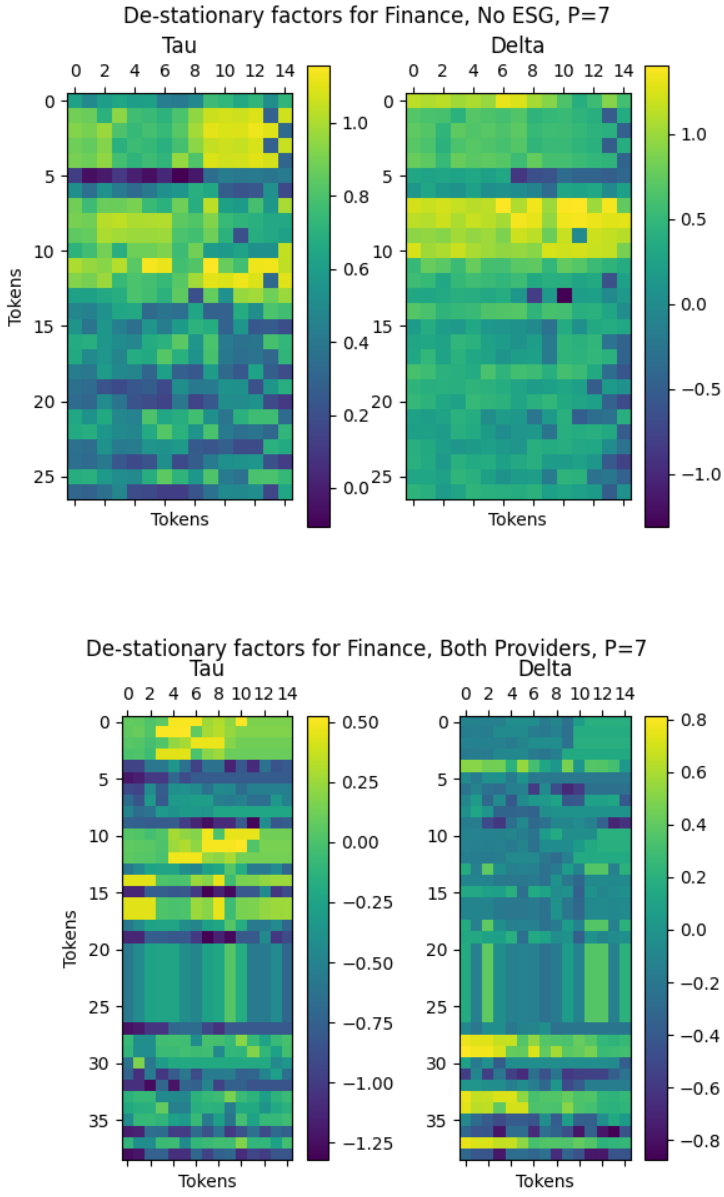


Figure 6.8: De-stationary factors for Finance, P=1, S + R (top), No ESG (bottom).

6.8 Conclusion

In this chapter, we used the dataset established in Chapter 5 in conjunction with the NSiTransformer developed in Chapter 3 to produce a stock predicting model. We used the walk-forward method to include data from multiple stocks, and determined the fairest comparison point between No ESG, Sustainalytics, Reuters and S+R. The best performing model integrated ESG data from three different providers, reinforcing the idea that the subjectivity of ESG ratings can be alleviated through multiple rating agencies. This study further corroborates the discrepancies in ESG ratings between agencies, as the inclusion of different providers yielded different results, despite clearly defining a common ground for evaluation.

The NSiTransformer provided accurate predictions, and saw an improvement in performance when the dataset was augmented with ESG data, indicating their benefits for the predictive power of the model. Through the lens of interpretability, we determined the importance of the SASB categorization replacing the ticker as an identifier for companies. Interpretability also allowed us to study the strategy adopted by the NSiTransformer to make a prediction and highlighted the role of the financial and extra financial parts of the dataset. The model employs a general strategy based on the financial data and targets specifically the company using the categorical variables.

Future work should try to propose different embeddings of the ESG ratings, and include other providers. A possible angle could be to study the interaction of different predicted variables with ESG ratings. From a model point of view, building larger models that encompass more stocks and financial sectors could lead to better performance. New model-specific techniques can also be explored, notably with Kolmogorov-Arnold networks to replace multi-layer perceptrons. Kolmogorov-Arnold networks implement tuning activation functions instead of traditional neural network weights, and can offer a different perspective to timeseries forecasting.

Chapter 7

Fine-tuning Timeseries Predictors Using Reinforcement Learning

Fine-tuning is the action to specialize large models for local tasks. This chapter fine-tunes the predictors built in Chapter 1 using RL algorithms from Chapter 4 and recent research. This chapter includes repetitions from Chapter 4, notably in the background and methodology sections. The goal is to quickly summarize concepts previously introduced, in order for the chapter to also work as a standalone document.

7.1 Introduction

Time series predictors are generally trained using supervised learning on datasets. The standard setup divides the dataset into three segments: training, validation and testing. The model is initially fit on training data, then evaluated on the validation set to tune hyper-parameters and assess the predictive power. Finally, the test set is used by the final model to determine the accuracy on unseen data. These steps are well understood and constitute the backbone of supervised learning in timeseries prediction.

This methodology draws a strong parallel with large language models (LLMs), which are generally transformer-based models and use supervised learning for pre-training. The pre-training is a common step to all LLMs that starts with a large amount of raw data, compressed in the network. Once the pre-training is complete, alignment aims to tune the model to create an user-friendly experience. This step incentivizes answering and asking questions to contextualize requests, teaches the LLMs how to use external tools, or censors potentially harmful information that might lie within the embeddings. The novelty of this research lies in the extension of alignment to time-series prediction model.

Fine-tuning in large language models was initially based on human feedback. A standard setup consists of a human operator prompting a question and grading the answer. Later, practitioners aimed at removing subjectivity from the fine-tuning pipeline by instead proposing two answers to a prompt and have a human operator select the best one. These two methods fall under the umbrella of Reinforcement Learning with Human Feedback (RLHF), and although effective recent models have shown pure Reinforcement Learning (RL) approaches outperforming RLHF for a fraction of the cost.

The central idea of this chapter is to leverage the predictive power of supervised pre-training and to use RL algorithms to align the model with diverse constraints. These constraints can be domain specific, such as risk management or operational constraints, but can also be purely mathematical, such as incentivizing bolder out-of-sample predictions. The reward function is at the center of the tuning, and will determine which direction the model is pushed towards. This approach is more adapted to time-series prediction compared to RLHF, as it completely removes the human operator and the need to reduce subjectivity in the feedback. RL is also extremely cost effective, since it removes the need of a coordinated effort of human operators giving feedback on a large number of samples.

In the context of time-series prediction, RL for fine-tuning is novel. The standard implementations of reinforcement learning in time-series prediction consist of a completely untrained agent learning a policy over a simulated environment. In the case of finance, this environment might be a portfolio or an ensemble of assets. In this chapter, we used a pre-trained model from [41] that serves as a backbone for the RL implementation. The environment is set up to reflect the training data closely, with the main tuning tool available being the reward structure. The loss is back-propagated through the backbone, updating the weights according to the policy.

The research questions investigated in this chapter are: Can we fine-tune pre-trained models to enhance time-series predictions using reinforcement learning? What state-of-the-art reinforcement algorithms work best for fine-tuning?

This chapter is structured as follow: Section 7.2 presents a literature review, section 7.3 the data used to train/test the models, Section 7.4 the framework used to fine-tune and evaluate the models, Section 7.5 benchmarks the models on standard reinforcement learning tasks, Section 7.6 the results of the fine-tuning, Section 7.7 the tuning of the specific hyperparameters and finally Section 7.8 is the conclusion to the chapter.

7.2 Background

Fine-tuning has become an emerging trend since large pre-trained model became more widely available to the public [179]. Fine-tuning is a technique that in-

tends to specialize a pre-trained backbone model, often to increase performance on selected benchmarks [180] or to benefit from previously acquired knowledge through transfer learning [181]. The democratization of open source models with available weights in natural language processing [85], [182] and image processing [183] enabled researchers and enthusiasts to propose their own fine-tuned version of an advanced model without the high computational cost of pre-training. Fine-tuning was leveraged to propose fine art classification [184], fine-tuning large language models for better medical care [185], biomedical tasks in different languages [186], and malware detection in images [187].

As the size of models and the parameter number grow exponentially, fine-tuning the entire model for each downstream tasks was replaced with a sparser approach called parameter-efficient fine-tuning [188], [189]. Methods such as Adapter [190], [191], LoRA [192] and Prefix-tuning [193] propose to modify the architecture of the original model to benefit from higher order patterns learned during supervised learning while also specializing in a downstream task. Supervised fine-tuning uses labeled data after pre-training to align the model towards a downstream task. This method has grown in popularity as large language models hit the public sphere and adapted for more intuitive or safer usage [194], [195], [196].

As the cost of computation carried over to efficient data labeling [197], alternative techniques for fine-tuning were explored. Reinforcement learning, one of the major paradigms in machine learning, has become one of the prime candidate for efficient fine-tuning. Adversarial networks had previously shown promising results [198], and policy learning has been employed in text-to-image [199] and multi-modal models [200]. Perhaps the most impressive implementation of reinforcement learning based fine-tuning comes from the DeepSeek-v3 report [201], which implements group proximal policy optimization to fine-tune a pre-trained model and implement chain-of-thoughts reasoning.

Within time-series prediction, fine-tuning has been focused on domain adaptation. In a similar fashion to text and image generation, large pre-trained models are becoming available to researchers [202]. The models can then be fine-tuned for domain specific predictions and receive the same benefit as large language models [203], [204]. However, these methods involve supervised fine-tuning, which in the case of time-series prediction consists of adding data form the specific domain the model needs to be fine-tuned on. As large language models have proven in the past, this method of fine-tuning can quickly become unsustainable due to the increasing cost of data labeling. In this study, we follow the way paved by LLMs by proposing reinforcement learning to tune time-series predictors.

PPO is a policy gradient method developed by John Schulman et al. in 2017 [106]. The key innovation of this algorithm over older methods such as TRPO [107] or ACER [108] is the clip function that constrains policy updates of the agent. PPO has been used in a wide variety of applications: Atari games [109], track rac-

ing games [110], suspension monitoring for cars [111], and image captioning [112]. A number of articles have proposed innovations to the base algorithm, for instance an alternative minimization target [113], [114] introduced policy feedback; specifically improving early learning stages, which are recognized as a potential weak point of PPO [115]. Recently proposed improvements include a shift in learning to offline policy optimization [116] and including conservatism [117].

Multi-agent methods have gained significant attention in the field of reinforcement learning, particularly for their capability to simulate complex systems involving interactive agents. A notable early work in multi-agent systems is [118] which explored the dynamics of cooperative and competitive agents in a shared environment. Recent advancements have integrated PPO into multi-agent applications: [119] applied multi-agent PPO to competitive and cooperative tasks, [120] successfully employed multi-agent reinforcement learning in the complex environment of the Dota 2 game. The integration of PPO into multi-agent systems has also been explored in real-world scenarios such as traffic light control [121], and collaborative robotics [122]. Innovations specific to multi-agent PPO include [123] which introduced a meta-learning approach to enhance adaptability across different tasks and agent configurations and [124], which presented the concept of leniency in multi-agent learning, mitigating the non-stationary issue commonly faced in such environments.

Attention is a machine learning mechanism designed to imitate human awareness. Attention was brought to the forefront of the field with the transformer architecture, a self-attention-based architecture that enabled the recent breakthroughs in large language models [29]. It has since seen many implementations including in recurrent neural networks for search results customization [125], missing data imputation [126], and in computer vision [127]. In reinforcement learning, attention models have been developed within theoretical frameworks [128] and diverse applications such as source code summarizing [129], dynamic graph problems [130], and road networks management [131].

The novelty of the framework presented lies in the combination of staple reinforcement learning models with time-series predictors. This chapter also creates an opportunity for further applications of the framework in simulated environment encompassing diverse fields.

7.3 Data

To contextualize the fine-tuning we detail the financial datasets used to train the backbone and to build the fine-tuning environment. We also present the MuJoCo framework, which we use to benchmark pure reinforcement learning performance between algorithms.

7.3.1 Financial and ESG Data

The financial and ESG data used in this chapter span from intraday market prices to annual sustainability ratings. Our primary sources are:

- **Refinitiv** [9]: a global leader in financial data and analytics, covering over 80% of global market capitalization with more than 450 ESG metrics. We extract daily price and volume data via Refinitiv Eikon, together with the three ESG pillar scores (Environmental, Social, Governance) and the combined ESG score.
- **Sustainalytics** [10]: provides ESG Risk Ratings for listed firms, widely used by asset managers and banks to construct sustainable portfolios. We incorporate their flagship ESG Risk Ratings into our dataset.
- **SASB Standards soderstrom2007ifrs**, [136]: the Sustainability Accounting Standards Board identifies material sustainability issues by industry. Since August 2022, SASB standards underlie IFRS S1 and S2 disclosures. We one-hot encode each firm's material SASB issue set based on the 2018 publication.

Table 7.1 shows a snippet of Apple's daily price data from 2005-12-05 to 2005-12-13. The full time span of the dataset is 2005-12-05 through 2024-08-07.

Date	Open	Low	High	Close	Volume
2005-12-05	2.17	2.15	2.19	2.16	5.84e8
2005-12-06	2.23	2.21	2.25	2.23	8.57e8
2005-12-07	2.24	2.20	2.24	2.23	6.79e8
2005-12-08	2.21	2.19	2.23	2.23	7.90e8
2005-12-09	2.24	2.21	2.25	2.24	5.55e8
2005-12-12	2.26	2.25	2.27	2.26	5.25e8
2005-12-13	2.25	2.24	2.27	2.26	4.94e8

Table 7.1: Sample daily financial data for AAPL

To enrich the raw price and volume data, we compute:

- *Log returns*, controlling for market effects via the Fama–French 5 factors [60].
- Technical indicators from historical prices and volumes:
  - Relative Strength Index (RSI) [205],
  - Moving Average Convergence Divergence (MACD) [206],

– Bollinger Bands [207].

The target variable is the FF5-adjusted log return, following the methodology of [20]. Financial data are available at sub-daily frequency, whereas ESG scores refresh annually (Refinitiv) or "regularly" (Sustainalytics). We evaluated regression, interpolation, autoencoders and forward-fill strategies. To respect provider methodologies and avoid compounding model error, we adopt a forward-fill approach for ESG values between update dates.

7.3.2 MuJoCo Benchmarking Environments

Multi-Joint dynamics with Contact, commonly called MuJoCo [134], proposes several standard environments to train and benchmark models on. To evaluate pure reinforcement learning performance, we employ three standard MuJoCo tasks:

- **HalfCheetah-v4**,
- **Hopper-v4**,
- **Humanoid-v4**.

MuJoCo provides a high-fidelity physics simulator for continuous-control benchmarks, where:

- *State* $s_t \in \mathbb{R}^d$ consists of joint angles, velocities and (for Humanoid) contact forces.
- *Action* $a_t \in \mathbb{R}^m$ represents torque inputs to each joint.
- *Reward* combines forward progress, control costs, and (where applicable) healthy posture and contact penalties.

Environment	Reward
HalfCheetah-v4	$R = w_f F - w_{\text{ctrl}} C$
Hopper-v4	$R = w_f F + w_h H - w_{\text{ctrl}} C$
Humanoid-v4	$R = w_f F + w_h H - w_{\text{ctrl}} C - w_{\text{ctct}} C_{\text{tct}}$

Table 7.2: MuJoCo environment reward functions (forward reward F, healthy reward H, control cost C, contact cost C_{tct})

Here, $w_f, w_h, w_{\text{ctrl}}, w_{\text{ctct}}$ are environment-specific weights. We use the default observation and action spaces as defined in OpenAI Gym's MuJoCo suite.

7.4 Framework Details

As mentioned in [132], implementation is key in deep policy gradient algorithms. As such, the framework below is implemented using the clean-rl library [133]. We evaluate three state-of-the-art algorithms for fine-tuning: Proximal Policy Optimization (PPO), Centralized Multi-Agent PPO (CMAPPO), and Group Relative Policy Optimization (GRPO). In this section, we also detail the environment used during training and the integration of the pre-trained transformer in the algorithms.

7.4.1 Proximal Policy Optimization (PPO)

- **Policy Function:** For an agent x, its policy at time t is a probability density function denoted as $\pi_\theta(a_t|o_t)$, where θ are the parameters of the policy, o_t is the observation for agent x at time t, and a_t are the actions that can be taken. The policy is then sampled to obtain the action taken $\alpha_t \sim \pi_\theta(a_t|o_t)$.
- **Objective Function:** The PPO objective function is defined as:

$$L^{PPO}(\theta) = \mathbb{E}_t \left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t) \right]$$

  where $r_t(\theta) = \frac{\pi_\theta(a_t|o_t)}{\pi_{\theta_{\text{old}}}(a_t|o_t)}$ is the probability ratio, ϵ an hyperparameter and $\hat{A}_t$ is an estimator of the advantage at time t, typically computed using Generalized Advantage Estimation (GAE).

- Advantage Estimation: The advantage $\hat{A}_t$ is computed as:

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \ldots + (\gamma\lambda)^{T-t+1}\delta_{T-1} \tag{7.1}$$

  with $\delta_t = r_t + \gamma V(o_{t+1}) - V(o_t)$ and V a learned state-value function.

- **Training Process:** The agent is trained by iteratively updating its policy parameters. This involves:

  1. Collecting trajectories by interacting with the environment using the current policy.
  2. Estimating the advantages using GAE.
  3. Calculating the surrogate objective function.
  4. Optimizing the surrogate objective function using gradient ascent while ensuring the updates stay within a specified clipping range to maintain policy stability.

7.4.2 Centralized Multi-Agent PPO (CMAPPO)

- **Subagent Policy & Training:** Each subagent x_i observes its local state $o_{t,i}$, samples an action $\alpha_{t,i} \sim \pi_{\theta_i}(a_{t,i} \mid o_{t,i})$, and learns via its own reward $R_i(o_t, a_{t,i})$ using PPO:

  1. *Collect trajectories:* Interact with environment to gather $\{(o_{t,i}, \alpha_{t,i}, r_{t,i})\}_{t=1}^{T}\}$.
  2. *Advantage estimation:* Compute $\hat{A}_{t,i}$ via GAE: $\hat{A}_{t,i} = \delta_{t,i} + (\gamma\lambda)\delta_{t+1,i} + +(\gamma\lambda)^2\delta_{t+2,i} + \cdots$, with $\delta_{t,i} = r_{t,i} + \gamma V(o_{t+1,i}) - V(o_{t,i})$.
  3. *Surrogate objective:*

  $$L_i^{\text{PPO}}(\theta_i) = \mathbb{E}_t\big[\, \min(r_{t,i}\hat{A}_{t,i},\ \text{clip}(r_{t,i}, 1-\epsilon, 1+\epsilon)\hat{A}_{t,i})\big],$$

  where $r_{t,i} = \dfrac{\pi_{\theta_i}}{\pi_{\theta_i^{\text{old}}}}$.

  4. *Policy update:* Perform gradient ascent on L_i^{PPO}, clipping updates to maintain stability.

- **Attention-Enhanced Aggregation:** Encode the global state e_t and subagent actions $\{\alpha_{t,i}\}$ via linear layers, compute attention weights $[w_{\text{env}}, w_1, \ldots, w_n] = \text{softmax}([f_{\text{env}}(e_t),\ f_{\text{sub}}(\{\alpha_{t,i}\})])$, then aggregate:

$$d_t = w_{\text{env}}\, e_t \ + \ \sum_{i=1}^{n} w_i\, \alpha_{t,i}.$$

- **Superagent Decision:** The superagent samples its final action $\alpha_t^f \sim \pi_{\theta_f}(a_t^f \mid d_t)$, allowing coordinated, adaptive decisions across all agents.

7.4.3 Group Relative Policy Optimization (GRPO)

- **Policy Function:** As in PPO, we parameterize a stochastic policy $\pi_\theta(a \mid o)$ with parameters θ. At each step t, given observation o_t, we sample a group of G candidate actions

$$a_{t,i} \sim \pi_\theta(\cdot \mid o_t), \quad i = 1, \ldots, G.$$

- **Group Rewards and Relative Advantage:** Each candidate action $a_{t,i}$ is scored by a reward function $r(a_{t,i}, o_t)$, yielding

$$r_{t,i} = r(a_{t,i}, o_t).$$

We compute the group baseline (mean) and standard deviation:

$$\bar{r}_t = \frac{1}{G}\sum_{i=1}^{G} r_{t,i}, \quad \sigma_t = \sqrt{\frac{1}{G}\sum_{i=1}^{G}\big(r_{t,i} - \bar{r}_t\big)^2} + \epsilon.$$

The *relative advantage* of candidate i is then:

$$A_{t,i} \;=\; \frac{r_{t,i} - \bar{r}_t}{\sigma_t}.$$

- **Surrogate Objective:** Defining the probability ratio for each candidate,

$$\rho_{t,i}(\theta) = \frac{\pi_\theta(a_{t,i} \mid o_t)}{\pi_{\theta_{\text{old}}}(a_{t,i} \mid o_t)},$$

the GRPO loss uses the same clipped surrogate as PPO but averages over the group:

$$L^{\text{GRPO}}(\theta) = \mathbb{E}_t\left[\frac{1}{G}\sum_{i=1}^{G} \min\big(\rho_{t,i}(\theta)A_{t,i},\ \text{clip}\big(\rho_{t,i}(\theta), 1-\epsilon, 1+\epsilon\big)A_{t,i}\big)\right].$$

Optionally, one may add a KL-penalty term $\beta D_{\text{KL}}\big(\pi_\theta(\cdot \mid o_t)\|\pi_{\text{ref}}(\cdot \mid o_t)\big)$ to constrain policy drift.

- **Training Process:** GRPO proceeds in iterative updates:

1. *Sample Groups:* For each observation o_t in a batch, sample G actions $\{a_{t,i}\}$.
2. *Evaluate Rewards:* Compute $r_{t,i} = r(a_{t,i}, o_t)$ for $i = 1 \ldots G$.
3. *Compute Advantages:* Form relative advantages $A_{t,i} = (r_{t,i} - \bar{r}_t)/\sigma_t$.
4. *Surrogate Update:* Optimize θ by ascending the clipped surrogate $L^{\text{GRPO}}(\theta)$ (plus optional KL term), using minibatch gradient steps.
5. *Repeat:* Collect new groups under the updated policy and continue until convergence.

7.4.4 Design of the Reinforcement Learning Environment

The RL environment is designed to facilitate the fine-tuning of forecasting policies:

- **State:** At time t, the state $s_t \in \mathbb{R}^{T\times N}$ is a matrix containing historical observations.

- **Agent Action:** The agent produces a forecast $a_{t,i} \in \mathbb{R}^{P\times 1}$ based on its local observation $o_{t,i}$.
- **Transition Dynamics:** Following the agents' actions, the true future $y_t \in \mathbb{R}^{P\times 1}$ is revealed, and the state is updated (via a sliding window mechanism).
- **Reward:** The reward r_t is computed based on the forecast error and any additional domain-specific criteria:

$$r_t = -\ell(a_t, y_t) - \psi(a_t), \tag{7.2}$$

  where $\ell(\cdot)$ is an error metric (e.g., absolute or squared error) and $\psi(\cdot)$ encapsulates further constraints or penalties.

In practice, the reward function used was $r_t = 2 \times e^{(-MSE(a_t,y_t)} - 1$. This implementation constrains the reward between $[-1, 1]$, and is driven up as the MSE converges towards 0.

7.4.5 Latent Representation versus Actor Network

In practice, the probability distribution each of the algorithms sample from is a neural network. In a classic reinforcement learning approach, a new network is created to learn the latent representation between observations and actions (the action network). In the case of PPO and CMAPPO, networks are also created to learn the value function (the critic network). To fine-tune a pre-trained backbone model, we need to integrate the trained network in the framework. There are two main paradigms for fine-tuning the network:

- **The backbone outputs a latent representation of the observation space.** The action network takes the latent representation as input and outputs a probability distribution over actions, which when sampled outputs the forecast. The critic network estimates the state value for advantage estimation and the gradients flow back through the action network, critic network, and the backbone, which leads to fine-tuning.
- **The backbone is connected to a projection layer that converts the latent representation to a forecast directly.** This is what commonly happens when the backbone is used independently as a predictor. In this paradigm, the backbone takes the place of the actor network. The critic network estimates the state value and the gradients flow back through the backbone and the critic network.

Using a separate action network can improve the flexibility since the actor network has the opportunity to learn from the latent features. Decoupling the backbone and the action network also allows us to adjust the hyperparameters for the

action network individually. An actor network is also more likely to explore and better adapt to the reward structure of the environment, performing significantly better in the reinforcement learning environment. We can also delay the fine-tuning by temporarily freezing all the backbone layers. This can be beneficial to performance as it gives the opportunity for the action and value networks to learn about the environment before inducing changes in the backbone network. This process can help avoid catastrophic forgetting during the early stages of interacting with the environment.

By replacing the actor network with the backbone, we ensure that a new actor network will not corrupt the original predictor. This approach is simpler and more direct, as the actor network introduces new hyperparameters but directly using the backbones only involves a minor projection. With no actor network involved, there is also less risk of overfitting the reinforcement learning task, thus maintaining a good degree of generalization. However, without an intermediary network to adapt the learned features, the backbone might struggle to perform and learn in the reinforcement learning environment. This can lead to repeated poor performance which in turn can flow through the gradient and cause catastrophic forgetting. The environment also needs to be carefully designed to avoid a mismatch between the observations at each step of the training and the encoder size of the backbone.

Both methods are compared in Table 7.3 using standard PPO. The reference scores are the scores of the backbone without any fine-tuning. The latent paradigm performs significantly worse, with only a small improvement in the Financial sector and massive loss in Industrials and Technology. The Actor paradigm improves upon the reference on all datasets. As such, we implemented the actor paradigm when possible. The only latent representation used was in CMAPPO with the superagent, as the aggregation of the subagents action does not correspond to the encoder accepted size of the backbone.

Table 7.3: Latent vs Actor paradigms comparison. The backbone is fine-tuned using PPO on Financial, Industrials and Technology. Reference is the base model without fine-tuning. Lower is better, in bold the best metric.

Dataset	Latent		Actor		Reference	
Metric	MSE	MAE	MSE	MAE	MSE	MAE
Financial	0.202	0.206	**0.200**	0.271	0.203	**0.118**
Industrials	0.274	0.251	**0.119**	**0.116**	0.128	0.121
Technology	0.341	0.264	**0.126**	**0.119**	0.131	0.119

7.5 Benchmarking

Three MuJoCo environments were selected as experimental settings. The three environments are: Hopper-v4, Half-Cheetah-v4 and Humanoid-v4. In this experiment, we use standard 64 hidden dimensions networks for the action and value heads. Table 7.4 presents the results of the three algorithms tested on each MuJoCo environment. CMAPPO wins out on all three environments, followed closely by default PPO. The GRPO algorithm, which does not use a critic network, underperforms slightly in the pure reinforcement learning task, especially in the Hopper-v4 environment.

Table 7.4: Results of MuJoCo environment training. Higher is better, best value in bold.

Model Environment	PPO Reward	CMAPPO Reward	GRPO Reward
HalfCheetah-v4	-150.54	**-111.10**	-137.18
Hopper-v4	1185.06	**1960.75**	624.86
Humanoid-v4	2897.81	**3201.09**	2659.32

7.6 Results

Fine-tuning is by definition local and its performance is measurable on a case-by-case basis. To cover as many use cases as possible, we propose to examine the results through the use of two common techniques in fine-tuning: layers freezing and transfer learning.

7.6.1 Fine-tuning and Frozen Layers

In order to retain high level patterns learned during supervised training, we can freeze parts of the model to stop the loss propagation through the network. This technique is common in large language models alignment and is employed to build the results in Table 7.5. We fine-tune the model with no frozen layers, 25%, 50% and 75% frozen layers.

Table 7.5: Results of fine-tuning models on Financial, Industrials and Technology dataset compared to the original model. In rows, the model's layers are progressively frozen. In columns, each sector represents the testing set of the model. Lower is better, best value in bold.

Frozen %	Model	Financial		Industrials		Technology	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
0%	PPO	0.200	0.271	0.119	0.116	0.126	0.119
	CMAPPO	0.324	0.208	0.146	0.160	0.203	0.189
	GRPO	0.198	0.109	0.118	0.113	0.124	0.115
25%	PPO	0.199	0.114	0.120	0.116	0.125	0.118
	CMAPPO	0.300	0.204	0.202	0.211	0.525	0.341
	GRPO	0.198	0.108	**0.118**	**0.112**	0.124	0.113
50%	PPO	0.202	0.113	0.119	0.117	0.124	0.117
	CMAPPO	0.237	0.155	0.257	0.248	0.151	0.151
	GRPO	**0.195**	**0.108**	**0.118**	**0.112**	0.124	0.113
75%	PPO	0.200	0.114	0.119	0.117	0.124	0.117
	CMAPPO	0.270	0.183	0.289	0.272	0.137	0.135
	GRPO	**0.195**	**0.109**	0.118	0.113	**0.123**	**0.113**
Original	Backbone	0.202	0.111	0.120	0.115	0.124	0.116

GRPO performed the best overall, either improving or leaving the backbone model unchanged. Notably, freezing at least 50% of the encoder layers gave consistently the best performance when using GRPO. PPO proposed a minor improvement in some categories, for instance in Financial at 25%, but mostly left the model unchanged. CMAPPO performed the worst in the fine tuning, provoking large negative changes to the model even with 75% of the encoder frozen. The source of the performance of GRPO in fine-tuning is the same reason it was the worst performer in the pure reinforcement learning task: the absence of a value function. While this is mostly a disadvantage learning control tasks, in the case of fine-tuning the difference of complexity between the value network and the backbone severely hinders the performance of PPO and CMAPPO. In the case of CMAPPO, the latent representation offered by the subagents are also reconciled using an action network. This design is coherent with the original implementation of CMAPPO but also adds another layer of abstraction the model needs to learn. A possible improvement for PPO and CMAPPO would be to run the model without propagating the loss back to the backbone to train the value network. By delaying the learning, the value network could learn a proper representation of the advantage in the task and nudge the backbone in the right direction.

7.6.2 Transfer Learning

Transfer learning is a machine learning technique through which a model learns general concepts applicable across multiple datasets. We experiment on transfer learning by fine-tuning and testing the model on the three datasets.

Table 7.6: Reference values before fine-tuning.

Trained on	Financial		Industrials		Technology	
Tested on	MSE	MAE	MSE	MAE	MSE	MAE
Financial	**0.203**	0.118	0.207	0.113	0.203	**0.110**
Industrials	0.224	0.227	0.128	0.121	**0.122**	**0.114**
Technology	0.256	0.229	0.132	0.118	**0.131**	**0.117**

Table 7.6 presents the results of the model on the Finance, Industrials and Technology datasets before fine-tuning. Instead of training the backbone model on all three datasets and fine-tuning for one, we train the backbone on a single dataset and test the MSE/MAE on all three. The Financial appears as the most challenging dataset, performing quite worse than the baseline when tested on Industrials and Technology. The model trained on Industrials manages to nearly match the performance of the models trained on Financial and Technology. Finally, the Technology model is by far the best, outperforming Industrials even when tested on Industrials. This metric could be interpreted as the degree of high level patterns present in the dataset. These high level patterns can be applied to any similar dataset, and ultimately are more powerful predictive tools than the past history for a given example.

Table 7.7: Results of fine-tuning models on Financial, Industrials and Technology dataset compared to the original model. The model is fine-tuned and tested on the specified sector for each row. In columns, each sector represents the original training set of the model. Lower is better, best value in bold.

Trained on →		Financial		Industrials		Technology	
Fine-tuned on ↓	Method	MSE	MAE	MSE	MAE	MSE	MAE
Financial	PPO	0.279	0.196	0.213	0.219	0.247	0.219
	CMAPPO	0.224	0.143	**0.145**	0.152	**0.151**	0.151
	GRPO	0.230	0.156	0.155	0.167	0.170	0.165
	Baseline	**0.203**	**0.118**	0.207	**0.113**	0.203	**0.110**
Industrials	PPO	0.201	0.115	0.123	0.119	0.131	0.121
	CMAPPO	0.204	0.117	0.127	0.121	0.137	0.125
	GRPO	**0.197**	**0.110**	**0.119**	**0.113**	0.125	**0.114**
	Baseline	0.224	0.227	0.128	0.121	**0.122**	**0.114**
Technology	PPO	0.198	0.114	0.125	0.117	0.127	0.120
	CMAPPO	0.202	0.115	0.128	0.119	0.129	0.120
	GRPO	**0.189**	**0.108**	**0.118**	**0.112**	**0.123**	**0.113**
	Baseline	0.256	0.229	0.132	0.118	0.131	0.117

Table 7.7 presents the results of the model on the Finance, Industrials and Technology datasets after fine-tuning. A first observation is the improvement in performance in all nearly all models from the baseline in Table 7.6. Some of the most substantial gains are found in the model trained on Financial, which improved its performance in MSE for both Industrials and Technology but moreover completely dominates the MAE benchmark. On the MSE front, the model trained on Industrials had the best results and beat out the best reference values for each sector.

Notable exceptions are the model trained and fine-tuned on Financial, and the model trained on Technology and fine-tuned on Industrials. In both cases, neither PPO, CMAPPO or GRPO managed to improve the performance, and testing on unseen data yielded a worse result. For the first case, the likely explanation is an overfitting to the train data: effectively, the model was trained twice on the same dataset, once with supervised learning, and again using reinforcement learning. The second case is different: the original Technology model already performed outstandingly well in Industrials, beating out even the models trained on the complete dataset. The fine-tuning failed to further improve that performance, marking the importance of establishing baselines before introducing fine-tuning to the pipeline.

The patterns noticed in Table 7.5 largely stand, with GRPO clearly distinguishing as the better option in nearly all cases. CCMAPPO performed exceptionally well on Financial, outperforming both PPO and GRPO. The superagent managed

to reconcile the actions of the subagents despite the added complexity of the actor and critic network. PPO nearly always improves on the baseline and constitute a valid choice for fine-tuning. The recommended algorithm stands out as GRPO, which uses fewer computational resources and yields the best performance. Committing to the actor paradigm and removing the critic network greatly simplifies the fine-tuning architecture, allowing for direct backpropagation through the backbone without the need for intermediary networks.

These results also clearly indicate the value of transfer learning for timeseries predictor. One of the best use case for fine-tuning appears to be adapting models from their supervised training dataset to another. This is in line with the current state of fine-tuning in large language models, which often adapts model after pre-training to diverse specific tasks. This result also highlights two clear areas for improvement in timeseries predictors: firstly, large pre-trained models can be built, and later specialized to a given dataset. But the biggest challenge to generalize this method is to specify a model and a fine-tuning environment that allows for various observation spaces and exogenous features.

7.7 Key hyperparameters

PPO and its variants are known to be sensitive to hyperparameters. In order to compare each algorithm fairly, we show in this section specific and non-specific hyperparameters tuning. All models presented from this point onward use the backbone as the action network, and a value network with 2 layers and 256 hidden dimensions when relevant (PPO, CMAPPO).

7.7.1 Training time

Training time is common to PPO, CMAPPO and GRPO. A higher number of timesteps will lead to a better performance in the environment until the agent reaches a plateau, at the expense of a higher computational cost. We fine-tune the model on the Financial dataset using PPO at different timesteps and plot the MSE over time in Figure 7.1. We found 500 000 timesteps to be the best value as a balance between overfitting and underfitting.

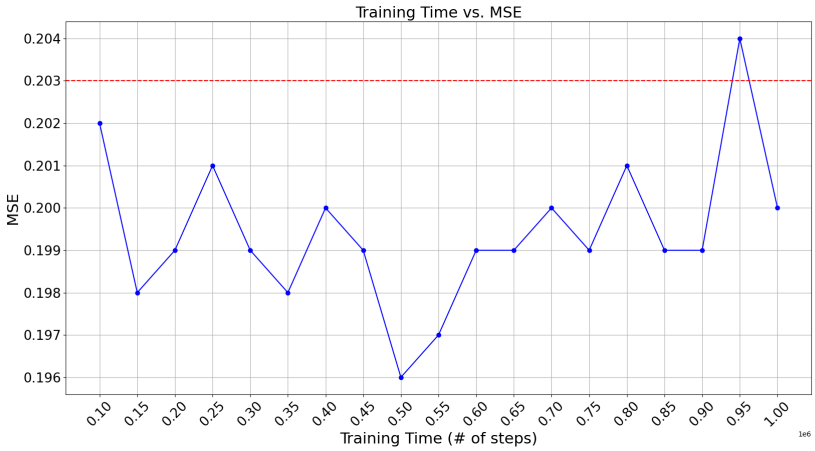


Figure 7.1: Training time vs MSE. Dotted line is the original model performance before fine-tuning. Training time is scaled down from 1e6 for readability.

7.7.2 Number of subagents (CMAPPO)

Number of subagents is specific to CMAPPO and controls how many subagents are trained before the superagent. We fine-tune a predictive model on the Financial dataset with an increasing number of subagents and test the MSE/MAE after fine-tuning. Table 7.8 presents the MSE and MAE with increasing numbers of subagents and compared to the backbone model. We found 10 subagents to be the best configuration, despite the backbone outperforming the fine-tuned model in all configurations.

Table 7.8: The influence of the number of subagents when fine-tuning the backbone compared to the non fine-tuned backbone.

Number of Subagents	Financial	
Metric	MSE	MAE
2	0.925	0.401
4	0.461	0.275
6	0.394	0.256
8	0.337	0.211
10	**0.271**	**0.183**
12	0.283	0.191
Backbone	0.202	0.111

7.7.3 Group size (GRPO)

Group size is specific to GRPO and determines the size of the group used to calculate the advantage. Similarly to Subesection 7.7.2, we fine-tune a predictive model on the Financial dataset with an increasing group size and test the MSE/MAE after fine-tuning. Table 7.9 presents the results of the model at group sizes from 2 to 12. We found that a group size of 8 is optimal for both computational load and model performance.

Table 7.9: The influence of the group size when fine-tuning the model on the Financial dataset.

Group Size	Financial	
Metric	MSE	MAE
2	0.200	0.204
4	0.198	0.199
6	0.197	0.199
8	**0.195**	0.196
10	0.199	0.201
12	0.201	0.203
Backbone	0.202	**0.111**

7.8 Conclusion

Fine-tuning timeseries predictors is emerging as an essential post-supervised training step to improve the performance of models. As the paradigm shifts from local models to larger, eclectic models harnessing the predictive power of many timeseries from diverse fields, fine-tuning becomes even more essential. At scale, is it far more cost effective to fine-tune a large model to a specific use case than retraining on large datasets. As the computational load for supervised learning gets higher and the models get larger, which has been the trend observed in LLMs and timeseries predictors, fine-tuning becomes even more attractive.

There are still several limitations, the most prominent being that pre-trained models use a fixed size input vector. This problem is not encountered in standard large language models, as the alphabet is tokenized to represent the entirety of the model output. But timeseries prediction is a continuous process, and further innovation is needed in foundational model to break out of fixed size vectors and scale up the models on large datasets, without relying on tricks such as projection layers. In the same spirit, architectural changes to foundational model allowing for variable output vector size would benefit the industry integration of timeseries predictors.

The environment definition and reward structure are key to the success of fine-tuning. Empirically, we noticed better results by bounding the reward to values between -1 and 1. The algorithm used is also a determining factor, and GRPO emerges as the clear winner in this chapter. This result is in line with the recent advances in LLMs, and further strengthens the conjecture that LLMs and timeseries predictors based on the same architecture share scaling features. If this conjecture reveals to be true, timeseries predictors are in a fantastic second mover position to implement even more innovations the thriving LLM community is building.

Chapter 8

Discussion & Conclusion

In this final chapter, we call back on the research questions from Section 1.2 and give a clear answer to each in relation to the work in the thesis. We then propose a discussion and a conclusion to the thesis, summarizing our findings and framing the results in the current landscape of ESG ratings, timeseries prediction and interpretability.

8.1 Answers to the Research Questions

RQ1 As shown in Chapter 2, ESG ratings are correlated to controlled log returns. Sections 2.5.2 and 2.5.3 detail how the sector and materiality issues influence the explanatory power of the ESG-returns correlation. ESG ratings also improve the predictive power of timeseries predictors in financial prediction, as shown in Chapter 6. Experiments in Section 6.6 demonstrated that the best performing model at all prediction length included the most ESG features.

RQ2 Chapter 3 introduces the NSiTransformer, which performs at either SOTA or near-SOTA level on benchmark datasets. Chapter 4 presents CMAPPO, a centralized multi-agent alternative to PPO for reinforcement learning tasks. Chapter 7 develops reinforcement learning based fine-tuning as a method that can be applied to numerous timeseries predictors to further improve performance. Section 7.6 compares RL algorithms performance: GRPO performed the best over CMAPPO and PPO in fine-tuning, while CMAPPO performed the best in pure reinforcement learning, as shown in 4.4.

RQ3 Section 3.4 proposes relevance maps to help us determine the key features and strategies employed by the models to predict accurately. Applying this technique in Subsection 6.7.2, we observed a contribution from ESG ratings

similar to financial features, while also benefiting to the performance. Subsection 6.7.3 uses the de-stationary factors in the NSiTransformer to interpret the relationship between features. In Section 4.7, we offer supplemental interpretability to the CMAPPO algorithm using attention weights and cosine distance between actions.

8.2 Discussion & Conclusion

Throughout the course of this thesis, ESG ratings have been shown to be a valuable tool to assess the value of a company. We showed that analyzing the data by sector and material issues is a relevant approach by establishing a correlation between the ratings and the controlled returns. By moving on to more powerful tools, such as the NSiTransformer, we determined that the addition of ESG ratings to the financial dataset improves the predictive power of the models. The integration of multiple providers was also central to this work, as ESG discrepancies are notably discussed in the literature. We determined that the improvement in performance was not only present when adding ESG ratings, but was also additive as more providers were added to the dataset. One key challenge encountered during this work was limited access to comprehensive ESG data, and the data that was available was sparse. ESG providers and rating agencies ask different questions related to their unique methodology, yet try to capture the same fundamental answer: how sustainable really is this company? Results from this work have proven that a truly good estimate is not only worthwhile for improved predictive power but also lie in the union of multiple providers. The integration of multiple providers is also a powerful hedge against greenwashing. But this comes at the high cost of several subscriptions to professional data vendors, which often get costlier for longer historical data. ESG ratings remain a fairly recent trend in finance, and do not have an extensive record of past values. Periodicity is also a major point of contention, since many rating agencies depend on financial disclosures that can happen yearly or quarterly depending on company location. However, a slow-moving indicator does not detract from its meaning, as evidenced by extremely important indicators exogenous to company, for instance interest rates. The pace at which the indicator evolves also reflects the pace at which companies are evolving: A global sustainable effort overhauling a company is not going to show direct signs on a day-to-day basis, as opposed to the close price or the trading volume. There is a clear incentive for new ESG ratings providers to appear, as younger retail investors are joining the market and have been sensitized to global warming and equality since childhood. With new providers come new methods, and ultimately the union of these methods will converge to capture a better picture of how truly sustainable companies are.

The contributions of this thesis to timeseries prediction are two-fold: first, the

introduction of the Non-stationary inverted Transformer as a state of the art model for timeseries prediction. Second, the fine-tuning of large pre-trained models using reinforcement learning. The approach that emerges as best performing was to pre-train a large model on many samples, and then fine-tune the model for a specific use. In fact, timeseries prediction positions itself with an incredible second-mover advantage: the research and development spent on transformers directly contributes to the inverted transformer architectures. For instance, Deepseek-R1 largely credits its success to Group Relative Policy Optimization, which is the reinforcement learning approach they used for fine-tuning. Using pure reinforcement learning for fine-tuning was not the first intuition of researchers: LLMs went through multiple human-aligned generations before it was established to ultimately perform worse than reinforcement learning. Human alignment required significant resources and organizational challenges, which have been circumvented to the benefit of timeseries prediction. These conclusions mirror the current state of the art in large language models (LLM), and future innovations should focus on implementing the tools created during the current surge of LLM research. In this thesis, we already observed transfer learning when pre-training the model on a sector and fine-tuning it on another sector, indicating the model learns general patterns on domain-adjacent data. At a larger scale, a pre-trained model can be trained on a variety of timeseries from any domain and learn general patterns. From a computational perspective, although the front cost of training a large model is expensive, the fine-tuning and inference are cheaper, ultimately covering many use cases that would have necessitated their own individual models. The current landscape of timeseries predictions can also benefit from the architectural innovations, such as reasoning tokens and chain-of-thoughts. The implementation would have to be different than a regular transformer that labels certain tokens as thinking tokens but could take the shape of intermediary predictions that converge towards the final prediction. Another idea with great potential is to use tools, which are now bundled in the most recent LLMs to allow the models to act directly on its environment, such as searching the internet or using a spreadsheet. Tools could help the model produce engineered features or look on the internet for supplemental features that can help with the prediction. Time series prediction finds itself right now in the unique position of benefiting from the hindsight of billions of dollars spent in research and development in adjacent models. In this position lies an incredible opportunity for researchers and private companies to propose models that can be duplicated and fine-tuned to tackle extremely specific needs.

Interpretability was shown in this thesis as a swiss-army knife to deal with a variety of issues that arise when building larger models. It is essential to determine the key features of a dataset, and assess the contribution of newly introduced features. In that capacity, we used it to determine the strategy used by the model to forecast accurately. This knowledge is of immense help considering the often

opaque relationships leveraged by large models to predict. Mirroring this idea, we can also recognize which features are not useful to the model, and as such reduce the noise and more importantly the computational overhead. Relevance maps especially can be used as an analogous tool to Principal Component Analysis that curates the dataset depending on what the model uses the most. There is also a place for interpretability in hyperparameter tuning, since quantifying and understanding the strategy of the model can help understand which lever needs to be pulled in order to reach a better forecast. For instance, a model with too high of a learning rate might be highly polarized towards certain features and struggle to learn nuances from the dataset. Interpretability also emerges as a key factor when integrating the model in critical industries. For instance, the legal liability in self-driving cars accident or the automation of loan assessments are still open questions. A total lack of transparency in systems that affect the public's physical or financial well-being will severely undermine trust in artificial intelligence. As such, there is a growing need for clear legislation about to which length a company training an artificial intelligence model needs to go to make sure that there are not only safety nets but also a capacity to explain failure when it inevitably happens. In this thesis, we endeavored ourselves to provide interpretability from pre-training to fine-tuning. This end-to-end interpretability was baked into the models from the beginning, and ultimately the time spent to implement them was paid back in better models. If performance and cost reduction are essential to the companies building the largest models in the world, then interpretability should be built into the model. There is definitely progress in that direction: in research, Kolmogorov-Arnold networks are being developed to potentially replace multilayer perceptrons. These networks are based on tuning several activation functions and splines, ultimately leading to a closed form expression of the network. In the industry, tools that have shown to improve the performance of the model coincidentally also tend to help with performance. This is the case for chain-of-thoughts and reasoning tokens, which forces the model into a reflective state transparent to the user.

Taken together, these findings reinforce a broader theme: quantitative tools can only influence capital flows if stakeholders actually trust them. When that trust breaks, momentum evaporates. There is a strong parallel between the lack of company transparency resulting in noisy metrics and the lack of interpretability in machine learning models. ESG ratings and interpretability both raise a fundamental challenge for trust in systems too complex to audit directly. Whether hidden layers of management or feedforward networks, there is an instinctive unease toward opacity. Ultimately, investors vote with their money: capital will concentrate around the companies and models that demonstrate transparency and accountability. In an era where data is plentiful but confidence scarce, trust becomes the decisive competitive advantage.

Bibliography

[1] UN, *Take action for the sustainable development goals – united nations sustainable development*, 2020. [Online]. Available: https://www.un.org/sustainabledevelopment/sustainable-development-goals/.

[2] M. Friedman, "The social responsibility of business is to increase its profits," *The New York Times Magazine*, 1970.

[3] Y. Meng and X. Wang, "Do institutional investors have homogeneous influence on corporate social responsibility? evidence from investor investment horizon," *Managerial Finance*, vol. 46, no. 3, pp. 301–322, 2020.

[4] A. L. Friedman and S. Miles, "Developing stakeholder theory," *Journal of management studies*, vol. 39, no. 1, pp. 1–21, 2002.

[5] R. Bénabou and J. Tirole, "Individual and corporate social responsibility," *Economica*, vol. 77, no. 305, pp. 1–19, 2010.

[6] G. Serafeim, "Public sentiment and the price of corporate sustainability," *Financial Analysts Journal*, vol. 76, no. 2, pp. 26–46, 2020.

[7] U. E. P. F. Initiative, *Un environment programme finance initiative*, https://www.unepfi.org/, 2024.

[8] Bloomberg, *Bloomberg*, https://www.bloomberg.com/europe/, 2024.

[9] Reuters, *Reuters*, https://www.reuters.com/, 2024.

[10] Sustainalytics, *Sustainalytics*, 2022. [Online]. Available: https://www.sustainalytics.com/.

[11] T. Verheyden, R. G. Eccles, and A. Feiner, "Esg for all? the impact of esg screening on return, risk, and diversification," *Journal of Applied Corporate Finance*, vol. 28, no. 2, pp. 47–55, 2016.

[12] A. Tsang, T. Frost, and H. Cao, "Environmental, social, and governance (esg) disclosure: A literature review," *The British Accounting Review*, vol. 55, no. 1, p. 101 149, 2023.

[13] B. Cheng, I. Ioannou, and G. Serafeim, "Corporate social responsibility and access to finance," *Strategic management journal*, vol. 35, no. 1, pp. 1–23, 2014.

[14] D. S. Dhaliwal, O. Z. Li, A. Tsang, and Y. G. Yang, "Voluntary nonfinancial disclosure and the cost of equity capital: The initiation of corporate social responsibility reporting," *The accounting review*, vol. 86, no. 1, pp. 59–100, 2011.

[15] S. El Ghoul, O. Guedhami, C. C. Kwok, and D. R. Mishra, "Does corporate social responsibility affect the cost of capital?" *Journal of banking & finance*, vol. 35, no. 9, pp. 2388–2406, 2011.

[16] A. Goss and G. S. Roberts, "The impact of corporate social responsibility on the cost of bank loans," *Journal of banking & finance*, vol. 35, no. 7, pp. 1794–1810, 2011.

[17] A. C. Ng and Z. Rezaee, "Business sustainability performance and cost of equity capital," *Journal of Corporate Finance*, vol. 34, pp. 128–149, 2015.

[18] X. Luo and C. B. Bhattacharya, "Corporate social responsibility, customer satisfaction, and market value," *Journal of marketing*, vol. 70, no. 4, pp. 1–18, 2006.

[19] E. Dimson, O. Karakaş, and X. Li, "Active ownership," *The Review of Financial Studies*, vol. 28, no. 12, pp. 3225–3268, 2015.

[20] H. Cazaux, R. Rudd, H. Stefánsson, S. Ólafsson, M. Raberto, and E. I. Ásgeirsson, "Correlation study between returns and esg ratings.," *Journal of Impact & ESG Investing*, vol. 5, no. 1, 2024.

[21] K. V. Lins, H. Servaes, and A. Tamayo, "Social capital, trust, and firm performance: The value of corporate social responsibility during the financial crisis," *the Journal of Finance*, vol. 72, no. 4, pp. 1785–1824, 2017.

[22] M. Khan, G. Serafeim, and A. Yoon, "Corporate sustainability: First evidence on materiality," *The accounting review*, vol. 91, no. 6, pp. 1697–1724, 2016.

[23] B. Jonsdottir, T. O. Sigurjonsson, L. Johannsdottir, and S. Wendt, "Barriers to using esg data for investment decisions," *Sustainability*, vol. 14, no. 9, p. 5157, 2022.

[24] M. Chen, R. von Behren, and G. Mussalli, "The unreasonable attractiveness of more esg data," *Available at SSRN 3881366*, 2021.

[25] S. Kotsantonis and G. Serafeim, "Four things no one will tell you about esg data," *Journal of Applied Corporate Finance*, vol. 31, no. 2, pp. 50–58, 2019.

[26] M. L. De Prado, *Advances in financial machine learning*. John Wiley & Sons, 2018.

[27] K. Shailaja, B. Seetharamulu, and M. Jabbar, "Machine learning in healthcare: A review," in *2018 Second international conference on electronics, communication and aerospace technology (ICECA)*, IEEE, 2018, pp. 910–914.

[28] S. Zhong, K. Zhang, M. Bagheri, *et al.*, "Machine learning: New ideas and tools in environmental science and engineering," *Environmental science & technology*, vol. 55, no. 19, pp. 12 741–12 754, 2021.

[29] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[30] B. Hutchinson, A. Smart, A. Hanna, *et al.*, "Towards accountability for machine learning datasets: Practices from software engineering and infrastructure," in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021, pp. 560–575.

[31] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, "Explainable ai: A review of machine learning interpretability methods," *Entropy*, vol. 23, no. 1, p. 18, 2020.

[32] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, p. 832, 2019.

[33] H. Nori, S. Jenkins, P. Koch, and R. Caruana, "Interpretml: A unified framework for machine learning interpretability," *arXiv preprint arXiv:1909.09223*, 2019.

[34] M. T. Ribeiro, S. Singh, and C. Guestrin, "Model-agnostic interpretability of machine learning," *arXiv preprint arXiv:1606.05386*, 2016.

[35] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu, "Definitions, methods, and applications in interpretable machine learning," *Proceedings of the National Academy of Sciences*, vol. 116, no. 44, pp. 22 071–22 080, 2019.

[36] C. Molnar, G. Casalicchio, and B. Bischl, "Interpretable machine learning–a brief history, state-of-the-art and challenges," in *Joint European conference on machine learning and knowledge discovery in databases*, Springer, 2020, pp. 417–431.

[37] M. Sundararajan and A. Najmi, "The many shapley values for model explanation," in *International conference on machine learning*, PMLR, 2020, pp. 9269–9278.

[38] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of research and development*, vol. 3, no. 3, pp. 210–229, 1959.

[39] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, pp. 1–40, 2016.

[40] A.-N. Liu, L.-L. Wang, H.-P. Li, J. Gong, and X.-H. Liu, "Correlation between posttraumatic growth and posttraumatic stress disorder symptoms based on pearson correlation coefficient: A meta-analysis," *The Journal of nervous and mental disease*, vol. 205, no. 5, pp. 380–389, 2017.

[41] H. Cazaux, R. Rudd, H. Stefánsson, S. Ólafsson, and E. I. Ásgeirsson, "Non-stationary inverted transformer with time2vec embedding," in *In Review at IEEE Transactions on Artificial Intelligence*, 2025.

[42] H. Cazaux, R. Rudd, H. Stefánsson, S. Ólafsson, and E. I. Ásgeirsson, "Inverted transformers interpretability beyond attention visualization," in *International Joint Conference on Neural Networks*, 2025.

[43] H. Cazaux, R. Rudd, H. Stefánsson, S. Ólafsson, and E. I. Ásgeirsson, "Controlled log returns prediction using nsitransformer on esg enhanced timeseries," in *Submitted at Journal of Sustainable Finance Investment*, Taylor Francis., 2025.

[44] H. Cazaux, R. Rudd, H. Stefánsson, S. Ólafsson, and E. I. Ásgeirsson, "Finetuning timeseries predictors using reinforcement learning," in *To be published in Recent Advances in Deep Learning*, Taylor Francis, 2025.

[45] D. Li and W.-L. Ng., "Optimal dynamic portfolio selection: Multiperiod meanvariance formulation," *Mathematical finance*, vol. 10, no. 3, pp. 387–406, 2000.

[46] V. Danciu, "The sustainable company: New challenges and strategies for more sustainability," *Theoretical and Applied Economics*, vol. 20, no. 9, pp. 7–26, 2013.

[47] E. Van Duuren, A. Plantinga, and B. Scholtens, "Esg integration and the investment management process: Fundamental investing reinvented," *Journal of Business Ethics*, vol. 138, no. 3, pp. 525–533, 2016.

[48] G. Giese, L.-E. Lee, D. Melas, Z. Nagy, and L. Nishikawa, "Foundations of esg investing: How esg affects equity valuation, risk, and performance," *The Journal of Portfolio Management*, vol. 45, no. 5, pp. 69–83, 2019.

[49] G. V. Research, "[14] gsia. global sustainable investment alliance investment review. 2018," *Available online: http://www.gsi-alliance.org/wpcontent/uploads/*, vol. 2018, p. 3, 2018. [Online]. Available: `https://www.grandviewresearch.com/industry-analysis/assetmanagement-market`.

[50] M. Friedman, "The social responsibility of business is to increase its profits," in *Corporate ethics and corporate governance*, Springer, 1970, pp. 173–178.

[51] M. T. Lee, R. L. Raschke, and A. S. Krishen, "Signaling green! firm esg signals in an interconnected environment that promote brand valuation," *Journal of Business Research*, vol. 138, pp. 1–11, 2022, ISSN: 0148-2963. DOI: https://doi.org/10.1016/j.jbusres.2021.08.061. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0148296321006287.

[52] I. Fafaliou, M. Giaka, D. Konstantios, and M. Polemis, "Firms' esg reputational risk and market longevity: A firm-level analysis for the united states," *Journal of Business Research*, vol. 149, pp. 161–177, 2022, ISSN: 0148-2963. DOI: https://doi.org/10.1016/j.jbusres.2022.05.010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0148296322004337.

[53] A. Fatemi, M. Glaum, and S. Kaiser, "Esg performance and firm value: The moderating role of disclosure," *Global Finance Journal*, vol. 38, pp. 45–64, 2018.

[54] L. T. Starks, P. Venkat, and Q. Zhu, "Corporate esg profiles and investor horizons," *Available at SSRN 3049943*, 2017.

[55] F. Alessandrini and E. Jondeau, "Optimal strategies for esg portfolios," *The Journal of Portfolio Management*, vol. 47, no. 6, pp. 114–138, 2021.

[56] A. B. Dor, J. Guan, and Y. Sun, "Is incorporating esg considerations costly?" *The Journal of Portfolio Management*, vol. 48, no. 7, pp. 75–87, 2022.

[57] W. F. Sharpe, "Capital asset prices: A theory of market equilibrium under conditions of risk," *The Journal of Finance*, vol. 19, no. 3, pp. 425–442, 1964.

[58] J. Lintner, "The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets," *The Review of Economics and Statistics*, pp. 13–37, 1965.

[59] E. F. Fama and K. R. French, "The cross-section of expected stock returns," *The Journal of Finance*, vol. 47, no. 2, pp. 427–465, 1992.

[60] E. F. Fama and K. R. French, "A five-factor asset pricing model," *Journal of Financial Economics*, vol. 116, no. 1, pp. 1–22, 2015.

[61] J. Derwall, N. Guenster, R. Bauer, and K. Koedijk, "The economic value of corporate eco-efficiency," *European Financial Management*, vol. 13, no. 5, pp. 684–717, 2007.

[62] R. Bauer, K. Koedijk, and R. Otten, "International evidence on ethical mutual fund performance and investment style," *Journal of Banking Finance*, vol. 33, no. 2, pp. 252–262, 2009.

[63] G. Friede, T. Busch, and A. Bassen, "Esg and financial performance: Aggregated evidence from more than 2000 empirical studies," *Journal of sustainable finance & investment*, vol. 5, no. 4, pp. 210–233, 2015.

[64] R. Gibson Brandon, P. Krueger, and P. S. Schmidt, "Esg rating disagreement and stock returns," *Financial Analysts Journal*, vol. 77, no. 4, pp. 104–127, 2021.

[65] R. G. Eccles, I. Ioannou, and G. Serafeim, "The impact of corporate sustainability on organizational processes and performance," *Management Science*, vol. 60, no. 11, pp. 2835–2857, 2014.

[66] D. K. Schooley and D. M. English, "Sasb: A pathway to sustainability reporting in the united states," *The CPA journal*, vol. 85, no. 4, p. 22, 2015.

[67] C. Busco, C. Consolandi, R. G. Eccles, and E. Sofra, "A preliminary analysis of sasb reporting: Disclosure topics, financial relevance, and the financial intensity of esg materiality," *Journal of Applied Corporate Finance*, vol. 32, no. 2, pp. 117–125, 2020.

[68] J. Grewal, C. Hauptmann, and G. Serafeim, "Material sustainability information and stock price informativeness," *Journal of Business Ethics*, vol. 171, pp. 513–544, 2021.

[69] K. R. French, *Kenneth r. french website*, 2022.

[70] K. French, *Data library ff5*, 2022.

[71] SASB, *Sasb*, `https://sasb.org/`, 2022.

[72] GICS, *Gics*, `https://www.spglobal.com/spdji/en/landing/topic/gics/`, 2022.

[73] M. Panna, "Note on simple and logarithmic return," *APSTRACT: applied studies in agribusiness and commerce*, vol. 11, no. 1033-2017-2935, pp. 127–136, 2017.

[74] P. Embrechts, R. Frey, and A. McNeil, *Quantitative risk management*. 2011.

[75] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *Journal of Political Economy*, vol. 81, no. 3, pp. 637–654, 1973, ISSN: 00223808, 1537534X. [Online]. Available: `http://www.jstor.org/stable/1831029` (visited on 11/14/2022).

[76] J. W. Lu and P. W. Beamish, "Sme internationalization and performance: Growth vs. profitability," *Journal of international entrepreneurship*, vol. 4, pp. 27–48, 2006.

[77] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" In *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, 2023, pp. 11 121–11 128.

[78] Y. Liu, T. Hu, H. Zhang, *et al.*, "Itransformer: Inverted transformers are effective for time series forecasting," *arXiv preprint arXiv:2310.06625*, 2023.

[79] Y. Wang, H. Wu, J. Dong, Y. Liu, M. Long, and J. Wang, *Deep time series models: A comprehensive survey and benchmark*, 2024. arXiv: 2407.13278 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2407.13278.

[80] R. Manuca and R. Savit, "Stationarity and nonstationarity in time series analysis," *Physica D: Nonlinear Phenomena*, vol. 99, no. 2-3, pp. 134–161, 1996.

[81] H. Chefer, S. Gur, and L. Wolf, "Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 397–406.

[82] OpenAI, *Openai*, https://openai.com/, 2024.

[83] Anthropic, *Claude*, https://www.anthropic.com/claude, 2024.

[84] Mistral, *Mistral 7b*, https://mistral.ai/, 2024.

[85] Meta, *Llama 3.1*, https://llama.meta.com/, 2024.

[86] Z. Zeng, C. Liu, Z. Tang, K. Li, and K. Li, "Acctfm: An effective intra-layer model parallelization strategy for training large-scale transformer-based models," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 12, pp. 4326–4338, 2022.

[87] V. A. Korthikanti, J. Casper, S. Lym, *et al.*, "Reducing activation recomputation in large transformer models," *Proceedings of Machine Learning and Systems*, vol. 5, pp. 341–353, 2023.

[88] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in neural information processing systems*, vol. 34, pp. 22 419–22 430, 2021.

[89] H. Zhou, S. Zhang, J. Peng, *et al.*, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, 2021, pp. 11 106–11 115.

[90] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *The eleventh international conference on learning representations*, 2023.

[91] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730*, 2022.

[92] Y. Liu, H. Wu, J. Wang, and M. Long, "Non-stationary transformers: Exploring the stationarity in time series forecasting," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9881–9893, 2022.

[93] A. Jha, O. Dorkar, A. Biswas, and A. Emadi, "Itransformer network based approach for accurate remaining useful life prediction in lithium-ion batteries," in *2024 IEEE Transportation Electrification Conference and Expo (ITEC)*, IEEE, 2024, pp. 1–8.

[94] L. Wang, Z. Li, Y. Chen, J. Wang, and J. Fu, "Maxent seismosense model: Ionospheric earthquake anomaly detection based on the maximum entropy principle," *Atmosphere*, vol. 15, no. 4, p. 419, 2024.

[95] W. Jia, S. Guan, and Y. Xue, "Tl-itransformer: Revolutionizing sea surface temperature prediction through itransformer and transfer learning," *Earth Science Informatics*, pp. 1–11, 2024.

[96] S. Jain and B. C. Wallace, "Attention is not explanation," *arXiv preprint arXiv:1902.10186*, 2019.

[97] S. Serrano and N. A. Smith, "Is attention interpretable?" *arXiv preprint arXiv:1906.03731*, 2019.

[98] M. Rigotti, C. Miksovic, I. Giurgiu, T. Gschwind, and P. Scotton, "Attention-based interpretability with concept transformers," in *International conference on learning representations*, 2021.

[99] S. Kim, J. Nam, and B. C. Ko, "Vit-net: Interpretable vision transformers with neural tree decoder," in *International conference on machine learning*, PMLR, 2022, pp. 11 162–11 172.

[100] Y. Qiang, C. Li, P. Khanduri, and D. Zhu, "Interpretability-aware vision transformer," *arXiv preprint arXiv:2309.08035*, 2023.

[101] S. M. Kazemi, R. Goel, S. Eghbali, *et al.*, "Time2vec: Learning a vector representation of time," *arXiv preprint arXiv:1907.05321*, 2019.

[102] Y. Wang, H. Wu, J. Dong, Y. Liu, M. Long, and J. Wang, "Deep time series models: A comprehensive survey and benchmark," 2024.

[103] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," *arXiv preprint arXiv:2210.02186*, 2022.

[104] Z. Liu, Y. Wang, S. Vaidya, *et al.*, "Kan: Kolmogorov-arnold networks," *arXiv preprint arXiv:2404.19756*, 2024.

[105] B. Ning, S. Jaimungal, X. Zhang, and M. Bergeron, "Arbitrage-free implied volatility surface generation with variational autoencoders," *SIAM Journal on Financial Mathematics*, vol. 14, no. 4, pp. 1004–1027, 2023.

[106] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[107] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, PMLR, 2015, pp. 1889–1897.

[108] Z. Wang, V. Bapst, N. Heess, *et al.*, "Sample efficient actor-critic with experience replay," *arXiv preprint arXiv:1611.01224*, 2016.

[109] L. Kaiser, M. Babaeizadeh, P. Milos, *et al.*, "Model-based reinforcement learning for atari," *arXiv preprint arXiv:1903.00374*, 2019.

[110] M. S. Holubar and M. A. Wiering, "Continuous-action reinforcement learning for playing racing games: Comparing spg to ppo," *arXiv preprint arXiv:2001.05270*, 2020.

[111] S.-Y. Han and T. Liang, "Reinforcement-learning-based vibration control for a vehicle semi-active suspension system via the ppo approach," *Applied Sciences*, vol. 12, no. 6, p. 3078, 2022.

[112] L. Zhang, Y. Zhang, X. Zhao, and Z. Zou, "Image captioning via proximal policy optimization," *Image and Vision Computing*, vol. 108, p. 104 126, 2021.

[113] T. Kobayashi, "Proximal policy optimization with relative pearson divergence," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 8416–8421.

[114] Y. Gu, Y. Cheng, C. P. Chen, and X. Wang, "Proximal policy optimization with policy feedback," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 7, pp. 4600–4610, 2021.

[115] C. C.-Y. Hsu, C. Mendler-Dünner, and M. Hardt, "Revisiting design choices in proximal policy optimization," *arXiv preprint arXiv:2009.10897*, 2020.

[116] Q. Cai, Z. Yang, C. Jin, and Z. Wang, "Provably efficient exploration in policy optimization," in *International Conference on Machine Learning*, PMLR, 2020, pp. 1283–1294.

[117] T. Yu, A. Kumar, R. Rafailov, A. Rajeswaran, S. Levine, and C. Finn, "Combo: Conservative offline model-based policy optimization," *Advances in neural information processing systems*, vol. 34, pp. 28 954–28 967, 2021.

[118] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," *Proceedings of the Tenth International Conference on Machine Learning*, 1993.

[119] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in Neural Information Processing Systems*, 2017.

[120] C. Berner, G. Brockman, B. Chan, *et al.*, "Dota 2 with large scale deep reinforcement learning," *arXiv preprint arXiv:1912.06680*, 2019.

[121] X. Liang, X. Du, G. Wang, and Z. Han, "Deep reinforcement learning for traffic light control in vehicular networks," *arXiv preprint arXiv:1904.08117*, 2019.

[122] L. Matignon, G. Laurent, and N. Le Fort-Piat, "Coordinated multi-agent learning: The state of the art," *Artificial Intelligence Review*, vol. 37, no. 3, pp. 219–250, 2012.

[123] T. Yu, G. Qu, A. Singh, S. Levine, and C. Finn, "Meta-learning with latent embedding optimization in multi-agent systems," in *International Conference on Learning Representations*, 2020.

[124] G. Palmer, K. Tuyls, D. Bloembergen, and R. Savani, "Lenient multi-agent deep reinforcement learning," *arXiv preprint arXiv:1805.04566*, 2018.

[125] X. Guo, H. Zhang, H. Yang, L. Xu, and Z. Ye, "A single attention-based combination of cnn and rnn for relation classification," *IEEE Access*, vol. 7, pp. 12 467–12 475, 2019.

[126] R. Wu, A. Zhang, I. Ilyas, and T. Rekatsinas, "Attention-based learning for missing data imputation in holoclean," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 307–325, 2020.

[127] M. J. Er, Y. Zhang, N. Wang, and M. Pratama, "Attention pooling-based convolutional neural network for sentence modelling," *Information Sciences*, vol. 373, pp. 388–403, 2016, ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2016.08.084. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025516306673.

[128] L. Bramlage and A. Cortese, "Generalized attention-weighted reinforcement learning," *Neural Networks*, vol. 145, pp. 10–21, 2022.

[129] W. Wang, Y. Zhang, Y. Sui, *et al.*, "Reinforcement-learning-guided source code summarization using hierarchical attention," *IEEE Transactions on software Engineering*, vol. 48, no. 1, pp. 102–119, 2020.

[130] U. Gunarathna, R. Borovica-Gajic, S. Karunasekara, and E. Tanin, "Solving dynamic graph problems with multi-attention deep reinforcement learning," *arXiv preprint arXiv:2201.04895*, 2022.

[131] C. Liu and G. Liu, "Jointppo: Diving deeper into the effectiveness of ppo in multi-agent reinforcement learning," *arXiv preprint arXiv:2404.11831*, 2024.

[132] L. Engstrom, A. Ilyas, S. Santurkar, *et al.*, "Implementation matters in deep policy gradients: A case study on ppo and trpo," *arXiv preprint arXiv:2005.12729*, 2020.

[133] S. Huang, R. F. J. Dossa, C. Ye, *et al.*, "Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms," *Journal of Machine Learning Research*, vol. 23, no. 274, pp. 1–18, 2022. [Online]. Available: http://jmlr.org/papers/v23/21-1342.html.

[134] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2012, pp. 5026–5033. DOI: 10.1109/IROS.2012.6386109.

[135] G. Qian, S. Sural, Y. Gu, and S. Pramanik, "Similarity between euclidean and cosine angle distance for nearest neighbor queries," in *Proceedings of the 2004 ACM symposium on Applied computing*, 2004, pp. 1232–1237.

[136] IFRS, *Ifrs s1*, https://www.ifrs.org/issued-standards/ifrs-sustainability-standards-navigator/ifrs-s1-general-requirements/, 2023.

[137] Y.-W. Cheung and K. S. Lai, "Lag order and critical values of the augmented dickey–fuller test," *Journal of Business & Economic Statistics*, vol. 13, no. 3, pp. 277–280, 1995.

[138] Sustainalytics, *Sustainalytics esg risk ratings methodology*, https://www.sustainalytics.com/docs/knowledgehublibraries/default-document-library/sustainalytics_-esg-risk-ratings_-version-3-1_-methodology-abstract_-june-2024.pdf?sfvrsn=567c06bc_1, 2024.

[139] T.-T. Li, K. Wang, T. Sueyoshi, and D. D. Wang, "Esg: Research progress and future prospects," *Sustainability*, vol. 13, no. 21, p. 11 663, 2021.

[140] E. Feigenbaum and H. Shrobe, "The japanese national fifth generation project: Introduction, survey, and evaluation," *Future Generation Computer Systems*, vol. 9, no. 2, pp. 105–117, 1993, FGCS Conference, ISSN: 0167-739X. DOI: https://doi.org/10.1016/0167-739X(93)90003-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0167739X93900038.

[141] B. Krollner, B. Vanstone, and G. Finnie, "Financial time series forecasting with machine learning techniques: A survey," in *European Symposium on Artificial Neural Networks: Computational Intelligence and Machine Learning*, 2010, pp. 25–30.

[142] K.-j. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1-2, pp. 307–319, 2003.

[143] F. E. Tay and L. Cao, "Application of support vector machines in financial time series forecasting," *omega*, vol. 29, no. 4, pp. 309–317, 2001.

[144] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," *Applied soft computing*, vol. 90, p. 106 181, 2020.

[145] J. Wang, S. Hong, Y. Dong, Z. Li, and J. Hu, "Predicting stock market trends using lstm networks: Overcoming rnn limitations for improved financial forecasting," *Journal of Computer Science and Software Applications*, vol. 4, no. 3, pp. 1–7, 2024.

[146] S. Hansun and J. C. Young, "Predicting lq45 financial sector indices using rnn-lstm," *Journal of Big Data*, vol. 8, no. 1, p. 104, 2021.

[147] K. Pawar, R. S. Jalem, and V. Tiwari, "Stock market price prediction using lstm rnn," in *Emerging Trends in Expert Applications and Security: Proceedings of ICETEAS 2018*, Springer, 2019, pp. 493–503.

[148] E. Hoseinzade and S. Haratizadeh, "Cnnpred: Cnn-based stock market prediction using a diverse set of variables," *Expert Systems with Applications*, vol. 129, pp. 273–285, 2019.

[149] M. U. Gudelek, S. A. Boluk, and A. M. Ozbayoglu, "A deep learning based stock trading model with 2-d cnn trend detection," in *2017 IEEE symposium series on computational intelligence (SSCI)*, IEEE, 2017, pp. 1–8.

[150] Y.-C. Chen and W.-C. Huang, "Constructing a stock-price forecast cnn model with gold and crude oil indicators," *Applied Soft Computing*, vol. 112, p. 107 760, 2021.

[151] S. Siami-Namini and A. S. Namin, "Forecasting economics and financial time series: Arima vs. lstm," *arXiv preprint arXiv:1803.06386*, 2018.

[152] J. Cao, Z. Li, and J. Li, "Financial time series forecasting model based on ceemdan and lstm," *Physica A: Statistical mechanics and its applications*, vol. 519, pp. 127–139, 2019.

[153] A. H. Bukhari, M. A. Z. Raja, M. Sulaiman, S. Islam, M. Shoaib, and P. Kumam, "Fractional neuro-sequential arfima-lstm for financial market forecasting," *Ieee Access*, vol. 8, pp. 71 326–71 338, 2020.

[154] X.-Y. Liu, H. Yang, Q. Chen, *et al.*, "Finrl: A deep reinforcement learning library for automated stock trading in quantitative finance," *arXiv preprint arXiv:2011.09607*, 2020.

[155] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 3, pp. 653–664, 2016.

[156] K. Lei, B. Zhang, Y. Li, M. Yang, and Y. Shen, "Time-driven feature-aware jointly deep reinforcement learning for financial signal representation and algorithmic trading," *Expert Systems with Applications*, vol. 140, p. 112 872, 2020.

[157] K. Mishev, A. Gjorgjevikj, I. Vodenska, L. T. Chitkushev, and D. Trajanov, "Evaluation of sentiment analysis in finance: From lexicons to transformers," *IEEE access*, vol. 8, pp. 131 662–131 682, 2020.

[158] D. Othan, Z. H. Kilimci, and M. Uysal, "Financial sentiment analysis for predicting direction of stocks using bidirectional encoder representations from transformers (bert) and deep learning models," in *Proc. int. conf. innov. intell. technol*, vol. 2019, 2019, pp. 30–35.

[159] R. Pan, J. A. García-Díaz, F. Garcia-Sanchez, and R. Valencia-García, "Evaluation of transformer models for financial targeted sentiment analysis in spanish," *PeerJ Computer Science*, vol. 9, e1377, 2023.

[160] E. Ramos-Pérez, P. J. Alonso-González, and J. J. Núñez-Velázquez, "Multi-transformer: A new neural network-based architecture for forecasting s&p volatility," *Mathematics*, vol. 9, no. 15, p. 1794, 2021.

[161] C. Yañez, W. Kristjanpoller, and M. C. Minutolo, "Stock market index prediction using transformer neural network models and frequency decomposition," *Neural Computing and Applications*, pp. 1–21, 2024.

[162] C. Xu, J. Li, B. Feng, and B. Lu, "A financial time-series prediction model based on multiplex attention and linear transformer structure," *Applied Sciences*, vol. 13, no. 8, p. 5175, 2023.

[163] J. Xu, "Ai in esg for financial institutions: An industrial survey," *arXiv preprint arXiv:2403.05541*, 2024.

[164] V. D'Amato, R. D'Ecclesia, and S. Levantesi, "Esg score prediction through random forest algorithm," *Computational Management Science*, vol. 19, no. 2, pp. 347–373, 2022.

[165] R. Hisano, D. Sornette, and T. Mizuno, "Prediction of esg compliance using a heterogeneous information network," *Journal of Big Data*, vol. 7, no. 1, p. 22, 2020.

[166] Y. Zou, "Predicting future esg performance using past corporate financial information: Application of deep neural networks," in *Proceedings of the 5th International Conference on Computer Information and Big Data Applications*, 2024, pp. 284–289.

[167] H. N. Bhandari, N. R. Pokhrel, R. Rimal, K. R. Dahal, and B. Rimal, "Implementation of deep learning models in predicting esg index volatility," *Financial Innovation*, vol. 10, no. 1, p. 75, 2024.

[168] V. D'Amato, R. D'Ecclesia, and S. Levantesi, "Fundamental ratios as predictors of esg scores: A machine learning approach," *Decisions in Economics and Finance*, vol. 44, no. 2, pp. 1087–1110, 2021.

[169] M. A. F. Chowdhury, M. Abdullah, M. A. K. Azad, Z. Sulong, and M. N. Islam, "Environmental, social and governance (esg) rating prediction using machine learning approaches," *Annals of Operations Research*, pp. 1–25, 2023.

[170] T. Guo, N. Jamet, V. Betrix, L.-A. Piquet, and E. Hauptmann, "Esg2risk: A deep learning framework from esg news to stock volatility prediction," *arXiv preprint arXiv:2005.02527*, 2020.

[171] H. Lee, J. H. Kim, and H. S. Jung, "Deep-learning-based stock market prediction incorporating esg sentiment and technical indicators," *Scientific Reports*, vol. 14, no. 1, p. 10 262, 2024.

[172] F. Ghallabi, B. Souissi, A. M. Du, and S. Ali, "Esg stock markets and clean energy prices prediction: Insights from advanced machine learning," *International Review of Financial Analysis*, p. 103 889, 2024.

[173] J. Park, H. J. Na, and H. Kim, "Development of a success prediction model for crowdfunding based on machine learning reflecting esg information," *IEEE Access*, 2024.

[174] C. Li, A. R. Keeley, S. Takeda, D. Seki, and S. Managi, "Investor's esg tendency probed by pre-trained transformers," *Corporate Social Responsibility and Environmental Management*, 2024.

[175] B. Sandwidi and S. P. Mukkolakal, "Transformers-based approach for a sustainability term-based sentiment analysis (stbsa)," in *Proceedings of the Second Workshop on NLP for Positive Impact (NLP4PI)*, 2022, pp. 157–170.

[176] S. Basso, A. Ceselli, and A. Tettamanzi, "Random sampling and machine learning to understand good decompositions," *Annals of Operations Research*, vol. 284, no. 2, pp. 501–526, 2020.

[177] D. O. Oyewola, E. G. Dada, and J. N. Ndunagu, "A novel hybrid walk-forward ensemble optimization for time series cryptocurrency prediction," *Heliyon*, vol. 8, no. 11, 2022.

[178] J. M. Cebrian, L. Natvig, and M. Jahre, "Scalability analysis of avx-512 extensions," *The Journal of supercomputing*, vol. 76, no. 3, pp. 2082–2097, 2020.

[179] K. W. Church, Z. Chen, and Y. Ma, "Emerging trends: A gentle introduction to fine-tuning," *Natural Language Engineering*, vol. 27, no. 6, pp. 763–778, 2021.

[180] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, *et al.*, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1299–1312, 2016.

[181] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146*, 2018.

[182] J. Devlin, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[183] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[184] E. Cetinic, T. Lipic, and S. Grgic, "Fine-tuning convolutional neural networks for fine art classification," *Expert Systems with Applications*, vol. 114, pp. 107–118, 2018.

[185] H. Xiong, S. Wang, Y. Zhu, *et al.*, "Doctorglm: Fine-tuning your chinese doctor is not a herculean task," *arXiv preprint arXiv:2304.01097*, 2023.

[186] L. Luo, J. Ning, Y. Zhao, *et al.*, "Taiyi: A bilingual fine-tuned large language model for diverse biomedical tasks," *Journal of the American Medical Informatics Association*, ocae037, 2024.

[187] D. Vasan, M. Alazab, S. Wassan, H. Naeem, B. Safaei, and Q. Zheng, "Imcfn: Image-based malware classification using fine-tuned convolutional neural network architecture," *Computer Networks*, vol. 171, p. 107 138, 2020.

[188] L. Xu, H. Xie, S.-Z. J. Qin, X. Tao, and F. L. Wang, "Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment," *arXiv preprint arXiv:2312.12148*, 2023.

[189] Z. Fu, H. Yang, A. M.-C. So, W. Lam, L. Bing, and N. Collier, "On the effectiveness of parameter-efficient fine-tuning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, 2023, pp. 12 799–12 807.

[190] R. Zhang, J. Han, C. Liu, *et al.*, "Llama-adapter: Efficient fine-tuning of language models with zero-init attention," *arXiv preprint arXiv:2303.16199*, 2023.

[191] R. He, L. Liu, H. Ye, *et al.*, "On the effectiveness of adapter-based tuning for pretrained language model adaptation," *arXiv preprint arXiv:2106.03164*, 2021.

[192] E. J. Hu, Y. Shen, P. Wallis, *et al.*, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[193] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," *arXiv preprint arXiv:2101.00190*, 2021.

[194] B. Gunel, J. Du, A. Conneau, and V. Stoyanov, "Supervised contrastive learning for pre-trained language model fine-tuning," *arXiv preprint arXiv:2011.01403*, 2020.

[195] Y. Zhou and V. Srikumar, "A closer look at how fine-tuning changes bert," *arXiv preprint arXiv:2106.14282*, 2021.

[196] T. Zhang, F. Wu, A. Katiyar, K. Q. Weinberger, and Y. Artzi, "Revisiting few-sample bert fine-tuning," *arXiv preprint arXiv:2006.05987*, 2020.

[197] T. Fredriksson, D. I. Mattos, J. Bosch, and H. H. Olsson, "Data labeling: An empirical investigation into industrial challenges and mitigation strategies," in *International Conference on Product-Focused Software Process Improvement*, Springer, 2020, pp. 202–216.

[198] T. Chen, S. Liu, S. Chang, Y. Cheng, L. Amini, and Z. Wang, "Adversarial robustness: From self-supervised pre-training to fine-tuning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 699–708.

[199] Y. Fan, O. Watkins, Y. Du, *et al.*, "Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 79 858–79 885, 2023.

[200] S. Zhai, H. Bai, Z. Lin, *et al.*, "Fine-tuning large vision-language models as decision-making agents via reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 37, pp. 110 935–110 971, 2025.

[201] A. Liu, B. Feng, B. Xue, *et al.*, "Deepseek-v3 technical report," *arXiv preprint arXiv:2412.19437*, 2024.

[202] Y. Liu, H. Zhang, C. Li, X. Huang, J. Wang, and M. Long, "Timer: Generative pre-trained transformers are large time series models," in *Forty-first International Conference on Machine Learning*, 2024.

[203] C. Chang, W.-C. Peng, and T.-F. Chen, "Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms," *arXiv preprint arXiv:2308.08469*, 2023.

[204] Y. Liang, H. Wen, Y. Nie, *et al.*, "Foundation models for time series analysis: A tutorial and survey," in *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, 2024, pp. 6555–6565.

[205] P. C. Belafsky, G. N. Postma, and J. A. Koufman, "Validity and reliability of the reflux symptom index (rsi)," *Journal of voice*, vol. 16, no. 2, pp. 274–277, 2002.

[206] T. T.-L. Chong and W.-K. Ng, "Technical analysis and the london stock exchange: Testing the macd and rsi rules using the ft30," *Applied Economics Letters*, vol. 15, no. 14, pp. 1111–1114, 2008.

[207] J. Bollinger, "Using bollinger bands," *Stocks & Commodities*, vol. 10, no. 2, pp. 47–51, 1992.

[208] H. Cazaux, R. Rudd, H. Stefánsson, S. Ólafsson, and E. I. Ásgeirsson, "Centralized multi-agent proximal policy optimization with attention," in *2024 International Conference on Machine Learning and Applications (ICMLA)*, IEEE, 2024, pp. 834–840.

Appendix A

Article 1 - Correlation Study between Returns and ESG Ratings [20]

Correlation Study Between Returns and ESG Ratings

Hugo Cazaux[a,c], Ralph Rudd[a,d], Hlynur Stefánsson[a,e], Sverrir Ólafsson[a,f], Marco Raberto[g,b], Eyjólfur Ingi Ásgeirsson[a,h]

[a]*Reykjavik University, Department of Engineering, Menntavegur 1, Reykjavik, 102, Iceland*
[b]*University of Genoa 6 Piazza della Nunziata Italy*
[c]*Corresponding author: hugot20@ru.is +354 7668580 Hatun 4 105 Reykjavik*
[d]*ralphr@ru.is*
[e]*hlynur@ru.is*
[f]*sverriro@ru.is*
[g]*marco.raberto@unige.it*
[h]*eyjolfur@ru.is*

Preprint submitted to The Journal of Impact & ESG Investing *December 18, 2023*

Correlation Study Between Returns and ESG Ratings

Abstract

ESG ratings have become a central topic in finance amid global initiatives for sustainability. This study examines the relationship between ESG ratings and log returns through Pearson's correlation coefficient, offering a granular analysis across various sectors and pinpointing specific materiality issues as defined by the Sustainability Accounting Standards Board (SASB).

The article uncovers empirical evidence that demonstrates a spectrum of negative to positive correlations, which are dependent on the sector in question and the relevant materiality issues. The study delves into the potential repercussions of such correlations, considering the perspectives of both corporations and investors. It underscores how positive correlations might incentivize companies to enhance their ESG ratings, whereas negative correlations could signal to investors considerations to take into account in their investment strategies.

The research also illuminates the pivotal role that SASB-defined materiality issues play in refining the understanding of ESG's impact on financial performance, suggesting that a nuanced approach to ESG investment could be beneficial. The study contributes to a deeper comprehension of how ESG factors are a valuable signal with financial outcomes, which could guide future corporate strategies and investment decisions towards sustainable growth.

Highlights:

- Controlling market factors significantly increases the correlation between ESG ratings and returns.
- Pearson's correlation coefficient is in majority weakly positive once controlled for market factors.
- Understanding the materiality of ESG factors by sector helps investors identify which companies might be better equipped to manage risks.
- Incorporating ESG ratings into risk management strategies can provide investors with a more nuanced approach to assessing firm-specific risks.

Keywords: ESG ratings, Correlation coefficient, Materiality, Log returns
PACS: G11, G14, G32, M14, Q56

1. Introduction

Investments and finance are critical components of society, for governments, companies, and individuals. The global asset management market, historically focused on maximizing return on investment (Li and Ng., 2000), strategically balances risk and reward in a complex financial environment to optimize investors' wealth. Investments drive the activities of companies and are essential for their survival and create clear incentives for companies to tailor their activities to suit the priorities of investors. There is a growing focus on sustainability and ethical behavior of companies (Danciu, 2013). The investment industry is a key driver in moving companies to be more sustainable and help reaching the UN Sustainable Development Goals (UN, 2020).

Multiple initiatives have already been undertaken to foster more sustainable investment practices. Specifically, the inclusion of Environmental, Social, and Governance (ESG) metrics (Van Duuren et al., 2016) in asset management has seen steady growth, indicating a paradigm shift towards sustainable, socially responsible, and ethically governed investment strategies. The ESG metrics are used to estimate how well companies are doing regarding sustainability, social rights, consumer protection, animal welfare, business ethics, and governance. ESG ratings have been shown to influence the systemic and idiosyncratic risk of companies (Giese et al., 2019). The Global Sustainable Investment Alliance defines ESG integration as "the systematic and explicit inclusion by investment managers of environmental, social and governance factors into financial analysis" (Research, 2018).

The research questions investigated in this article are the following: is there a correlation between the ESG ratings and returns of companies? Can the company sectors help identify groups with different relationships? Can the materiality issues provide more explanatory power?

This paper is structured as follows: Section 2 includes a contextualization and a review of the existing literature on ESG ratings. Section 3 examines the methodology used to control for market factors and compute the correlation. Section 4 contains details on the results, average correlation per sector, and statistically significant correlation for individual companies. Section 5 contains results of the correlation between variations of returns and ESG metrics. Section 6 is a discussion of their implications for investors and companies alike. Section 7 is the conclusion of the study. The Appendix provides a summary of the dataset variables, mathematical caveats about correlation and supplementary data.

2. Background

The search for a relationship between sustainability and corporate performance can be traced back to the 1970s (Friedman, 1970). Scholars have studied the impact on branding (Lee et al., 2022), market longevity (Fafaliou et al., 2022), and equity valuation

(Giese et al., 2019). Results of these studies indicate that failing to communicate strong ESG performance, specifically expressing low carbon emissions and employee satisfaction, reduces the odds for external financing and increases both the systematic and stock-specific risks. Studies have discussed the authenticity of the disclosure by companies (Fatemi et al., 2018), finding that disclosure can weaken the negative or positive valuation effects on company. Scholars also examined the integration of ESG ratings in portfolio strategies (Starks et al., 2017) (Alessandrini and Jondeau, 2021), showing that a strong ESG rating will attract long-term-oriented investors with a lower sensibility to immediate negative earnings (Dor et al., 2022).

While the exploration of ESG ratings and their financial implications has gained momentum in recent years, the underpinnings of this research lie in foundational asset pricing theories. Asset pricing theories, evolving over the decades, provide the scaffolding for understanding the determinants of asset prices and returns. The Capital Asset Pricing Model (CAPM) is a seminal theory in this domain, introduced by Sharpe (1964) and Lintner (1965). CAPM posits that the expected return on an asset is a function of its systematic risk, often measured by its beta relative to the market. While the model offers a simplistic view, it laid the foundation for subsequent models that incorporated multiple factors. Recognizing the limitations of CAPM, Fama and French (1992) introduced the Fama-French three-factor model, adding size and value factors to the market risk factor in CAPM. This model was further expanded into the Fama-French five-factor model by (Fama and French, 2015), incorporating profitability and investment factors, offering a more comprehensive understanding of asset returns.

As ESG ratings became more standardized and prevalent, the focus shifted towards these quantifiable metrics. Studies such as those by (Derwall et al., 2007) and (Bauer et al., 2009) explored the correlation between ESG ratings and stock performance. Their findings, while providing valuable insights, were often mixed. Some research indicated a positive relationship between high ESG scores and superior stock returns (Friede et al., 2015), (Gibson Brandon et al., 2021). Fewer studies delve into the causality of this relationship. (Eccles et al., 2014), for instance, suggested that firms with a long-term focus on ESG issues tend to outperform their counterparts in the long run, hinting at a potential causal link between ESG practices and financial performance.

As the ESG landscape continues to evolve, tools and frameworks that offer a more standardized approach to materiality assessment are emerging. Among these, the Sustainable Accounting Standards Board (SASB) materiality map stands out. SASB's approach to materiality emphasizes the financial materiality of ESG issues, identifying which issues are likely to affect the financial performance of companies within specific sectors (Schooley and English, 2015). Studies have praised the data quality of the reporting (Busco et al., 2020), specifically through the narrative of linking financial data to non-financial data.

The data has been used to evaluate performance of firms with different level of materiality ratings (Khan et al., 2016), finding that firms with strong ratings on material sustainability issues have better future performance than firms with inferior ratings. Grewal et al. (2021) concluded that scholars interested in understanding how sustainability information impacts economic value and stock prices need to incorporate a materiality lens into their analysis.

3. Methodology

3.1. Data Collection

For this study, the primary source of data was Thomson Reuters, offering a comprehensive dataset to investigate the relationship between ESG ratings and stock returns. The data used in this work was pulled in February 2022 and constitutes of the companies listed on the S&P500 at this point in time. Below are the fields obtained through the Thomson Reuters data stream.

- Date: Captures the date of each data entry, essential for time-series analysis and understanding temporal trends.
- Instrument: Denotes the specific stock or financial instrument under consideration.
- Open Price, Close Price, High Price, Low Price: These columns detail daily stock prices, crucial for computing daily and subsequently, annual log returns.
- ESG Score: An aggregate rating based on a firm's adherence and performance across environmental, social, and governance dimensions.
- Environmental Pillar Score: Focuses solely on a company's environmental practices and impacts.
- Social Pillar Score: Reflects how a company fares in social responsibilities, such as labor practices, product responsibility, and community relations.
- Governance Pillar Score: Offers insights into a firm's governance structures, ethical practices, and overall corporate accountability.

While the stock data was updated daily, the ESG scores were updated annually at each new fiscal year. This periodic update typically mirrors annual disclosures of key metrics. Scores range from 0 to 100.

To control the results with established financial frameworks, additional data will be incorporated. Established market factors from the Fama-French five-factors (FF5) models were obtained from the website of one of the creators of the model (French, 2022b)

The values of these factors are constructed using the 6 value-weight portfolios formed on size and book-to-market, the 6 value-weight portfolios formed on size and operating profitability, and the 6 value-weight portfolios formed on size and investment. The portfolios used are from the proof of concept available on the Fama-French 5 data library (French, 2022a). The coefficients are:

- Rm-Rf : Excess return on the market.
- Small Minus Big (SMB): Average return of the nine small stock portfolios minus the average return on the nine big stock portfolios.
- High Minus Low (HML): Average return on the two value portfolios minus the average return on the two growth portfolios.
- Robust Minus Weak (RMW): Average return on the two conservative investment portfolios minus the average return on the two aggressive investment portfolios.
- Conservative Minus Aggressive (CMA): Average return on the two value portfolios minus the average return on the two growth portfolios.

To contextualize and provide further explanatory power to this study, SASB materiality data were gathered. Their website (SASB, 2022) provides a Materiality Finder tool used to look up individual companies and which of the 26 issues are recognized as significant, mapping out the relevant issues for each company in the S&P500. Tables A.6 and A.7 summarize the datasets created for this study.

There are significant gaps in the availability of ESG data. Out of the 501 unique companies on the S&P index, every company exhibited at least one year of missing ESG data. 10 companies had no ESG data available and were removed from the dataset, bringing the total to 491. Figure 1 provides a year-by-year breakdown of the number of companies with complete ESG data. Notably, the years 2000-2005 have the most pronounced data omissions. This can be explained by a lesser focus on ESG ratings from companies at this time. We place our cut off in 2006, where 198 companies have a complete history, yet we maintain enough data points (15 per company). The final dataset contains 198 firms. This dataset is then used to calculate the correlation between ESG Metrics and returns for each ticker.

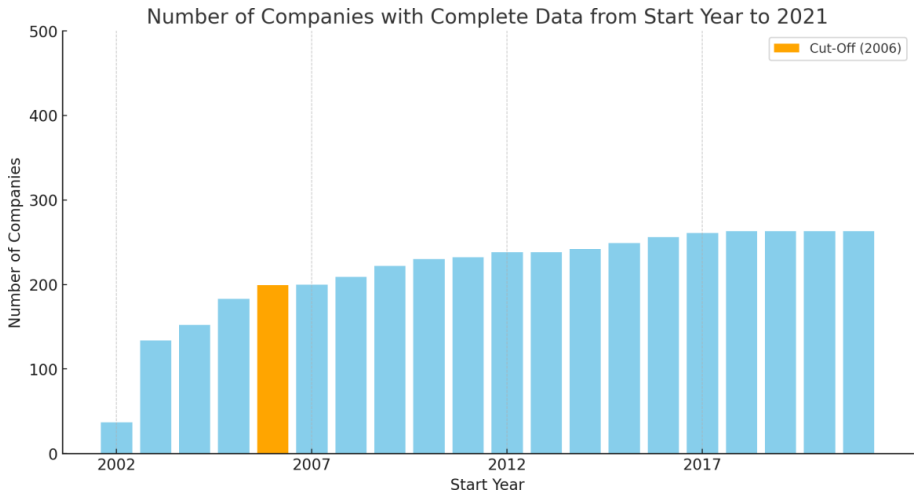


Figure 1: Number of Companies with Complete ESG Data Over The Years.

3.2. Sectors

The data is separated by sectors. The sectors are determined by the GICS (GICS, 2022). There are 11 sectors: Industrials, Financial Services, Healthcare, Consumer Cyclical, Consumer Defensive, Real Estate, Utilities, Technology, Basic Materials, Energy, Communication Services. Table 1 presents the number of companies in each sector before and after accounting for missing ESG data. The most represented GICS sector is the Industrial sector with 32 companies, while the least represented is Communication Services with 5 companies. Sectors Industrials and Basic Materials retained respectively 43.1% and 70% of their population after the cut-off. The Technology sector was left with 11.3% despite having the second highest number of companies. Communication Services was also left with 21.7% but encompasses only 23 companies before cut-off.

Table 1: Number of Companies and Unique Tickers in Each Sector.

	# of Companies	# of Companies after Cut-off	% of Companies Left after Cut-off
Industrials	73	32	43,1%
Financial Services	67	32	47.8%
Consumer Cyclical	57	25	43.9%
Healthcare	64	25	39%
Utilities	28	17	60.7%
Consumer Defensive	36	16	44.4%
Basic Materials	20	14	70%
Real Estate	31	14	43,1%
Energy	20	10	45.2%
Technology	71	8	11.3%
Communication Services	23	5	21.7%
Total	491	198	40.32%

3.3. Materiality

Materiality refers to an individual factor within a sector that influences a firm's financial performance. The SASB Materiality Standards help to increase the granularity of the analysis. Specifically, the materiality standards were used to highlight which score correlates the most with performance in the returns. Materiality factors can be found on the SASB website using a tool called "Materiality Finder" (SASB, 2022). This website was scraped for each company and the data stored in a binary vector. Each company can appear in multiple fields. Table A.7 summarizes the different materiality issues recognized by SASB. Figure 2 presents the number of companies per SASB issue before and after the cut-off. The most represented field is Product Design and Lifecycle Management with 117 companies. The least represented field is Competitive Behavior with 15 companies, except for Customer Privacy and Physical Impacts of Climate Change that are not represented in the dataset of companies with complete data post cut-off.

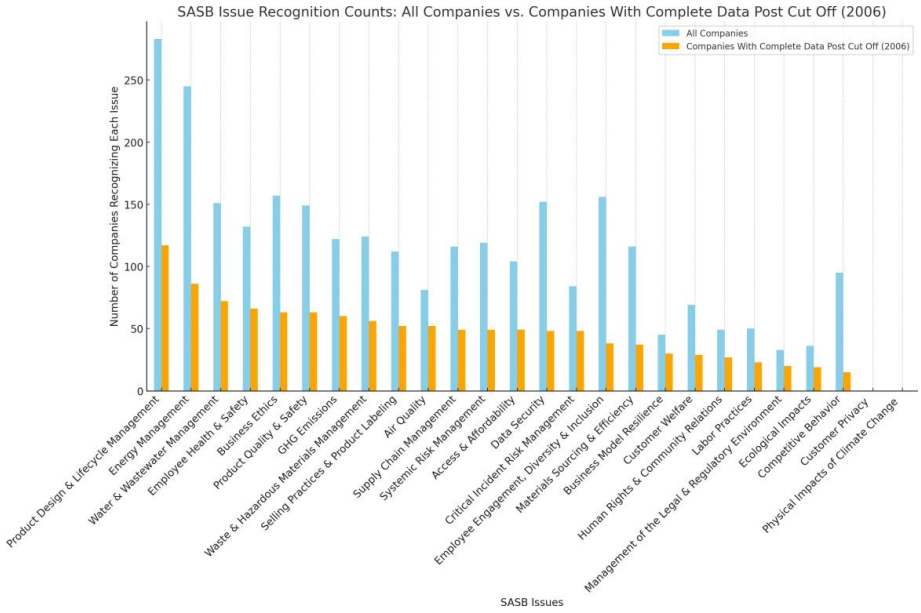


Figure 2: Number of Total Companies Per SASB Issue.

3.4. Data Integration and Analytical Model

Since the ESG data is yearly, to assess the annual performance of each stock, annualized returns were calculated. In order to obtain additive properties, returns are logged (Panna, 2017). Daily log returns for a company are given by: $r_t = ln(\frac{P_t}{P_{t-1}})$ where r_t represents the log return at time t, P_t is the closing price at time t and P_{t-1} is the closing price at time $t-1$.

Given daily log returns, the annualized log returns for a company are computed as:

$$r_{annualized} = \sum_{i=1}^{n} r_i$$

Equation 1

where $r_{annualized}$ is the annual log returns, r_i is the $i-th$ daily log return, and n the number of trading days in a year. For the rest of this article, annualized log returns will be referred to as returns.

3.5. Controlling for Market Factors

To isolate stock-specific characteristics, common market factors were controlled using the Fama-French Five-Factor Model. The model is given by:

$$R_{it} - R_f = \alpha_i + \beta_m(R_{mt} - R_f) + \beta_s \times SMB_t + \beta_v \times HML_t + \beta_{rmw} \times RMW_t + \beta_{cma} \times CMA_t + \varepsilon_{it}$$

Equation 2

where:

- R_{it}: Return on stock i at time t.
- R_f: Risk-free rate.
- R_{mt}: Market return at time t.
- SMB_t: Size factor (Small Minus Big), capturing the historical excess returns of small-caps over big-caps at time t.
- HML_t: Value factor (High Minus Low), capturing the historical excess returns of value stocks over growth stocks at time t.
- RMW_t: Profitability factor, capturing the difference in returns between companies with robust (high) and weak (low) at time t.
- CMA_t: Investment factor, capturing the difference in returns between companies with conservative and aggressive investments at time t.
- $\beta_m, \beta_s, \beta_v, \beta_{rmw}, \beta_{cma}$: Weights of the factors.
- α_i: Intercept, capturing stock *i*'s abnormal return unexplained by the factors.
- ε_{it}: Error term for stock i at time t.

Using this model, the returns are controlled for diverse common market factors. SMB_t, HML_t, RMW_t and CMA_t are given by the Fama-French 5 library. The coefficients $\alpha_i, \beta_m, \beta_s, \beta_v, \beta_{rmw}, \beta_{cma}$ and ε_{it} are fit per ticker using linear regression. Table 2 presents the coefficients regressed for four tickers. ε_{it} is not included in the table as it is unique per observation.

Table 2: Coefficients for tickers AAPL, AMZN, MSFT, and GOOGL

Ticker	α_i	β_m	β_s	β_v	β_{rmw}	β_{cma}
AAPL	-0.0052	0.0112	-0.0011	-0.0037	-0.0004	-0.0062
AMZN	-0.0053	0.0110	-0.0013	-0.0038	-0.0065	-0.0106
MSFT	-0.0057	0.0110	-0.0031	-0.0036	0.000059	-0.0030
GOOGL	-0.0042	0.0100	-0.0015	-0.0015	0.0002	-0.0076

3.6. Yearly Variation

To compute the correlation between the variation of both ESG metrics and log returns, the yearly difference for each is calculated and added to the dataset. For a given year k, the difference is defined as:

$$r_{variations,k} = r_{annualized,k} - r_{annualized,k-1}$$

Equation 3

$$ESG_{variation,k} = ESG_k - ESG_{k-1}$$

Equation 4

The variations are then normalized using Standard Score:

$$\frac{X - \mu}{\sigma}$$

Equation 5

where: X is the data point, μ is the mean of either the returns or ESG metric, and σ the standard deviation of either the returns or ESG metric.

3.7. Correlation Analysis

With both the annualized and controlled returns in hand, we computed Pearson's correlation between returns and the various ESG metrics on a per-stock basis.

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

Equation 6

where:

- r: Pearson's correlation coefficient, which measures the linear relationship between two datasets.
- $x_i \; and \; y_i$: Data values from the controlled or uncontrolled returns and an ESG metric being compared.
- $\bar{x}$ and $\bar{y}$: Mean values of controlled or uncontrolled returns and ESG metric being compared respectively.

A caveat that must be addressed when discussing correlation is the maximum attainable correlation for two given distributions. Pearson's correlation takes a value between [-1, 1], but this is only true if the two random vectors X_1 and X_2 are of the same type (Embrechts et al., 2011). The distribution model chosen for the annual log returns will be the normal distribution. This is an assumption commonly made within the Black-Scholes model (Black and Scholes, 1973). A Kolmogorov–Smirnov goodness of fit test was performed on the ESG ratings comparing the underlying distribution of a sample to a given distribution. The highest scoring distribution ended up being Johnson's SU. Given these two distributions, details of the calculations are available at Appendix B. These calculations indicate $[\rho_{max}, \rho_{min}] = [-0.9998, 0.9998]$. This interval is very close to [-1,1], analog to comparing 2 datasets with underlying normal distribution.

4. Results - Correlation between ESG Metrics and Returns

This section outlines the results of analyzing the correlation between stock returns and ESG scores. All the correlations are calculated for companies with a long enough rating history and are averaged in their specific group. The discussion begins with a broad overview of these findings, then delves into specifics related to industry sectors and sustainability issues.

4.1. Preliminary Statistics and Distributions

Table 3 presents the descriptive statistics from the correlations between ESG ratings and uncontrolled or controlled annualized log returns. After controlling with the Fama-French 5 model, the means and standard deviations for score-by-score comparison has increased for every metric.

Table 3: Statistics for Correlation between (Un)controlled Returns and ESG Metrics (2006 cut-off)

	mean	std	min	25%	50%	75%	max
Uncontrolled							
ESG Score	0.03	0.22	-0.46	-0.11	0.04	0.18	0.70
Environmental Pillar Score	0.03	0.22	-0.50	-0.11	0.02	0.17	0.66
Governance Pillar Score	0.02	0.22	-0.50	-0.12	0.02	0.15	0.69
Social Pillar Score	0.02	0.23	-0.51	-0.16	0.02	0.16	0.72
Controlled							
ESG Score	0.36	0.23	-0.44	0.25	0.39	0.52	0.81
Environmental Pillar Score	0.37	0.26	-0.38	0.24	0.41	0.55	0.84
Governance Pillar Score	0.21	0.31	-0.57	0.02	0.25	0.45	0.79
Social Pillar Score	0.30	0.26	-0.58	0.15	0.30	0.47	0.79

As shown in Table 3, the uncontrolled annualized log returns present little to no correlation with the ESG metrics. The controlled annualized log returns present a much higher average correlation and higher standard deviation. The global ESG Score and Environmental Pillar Score appear to be the most correlated with the returns. As such, for the rest of the results, the correlations presented will be calculated using the returns controlled by Fama-French 5.

In Table 3, the most significant values are observed in the maximum correlations for ESG Score and Governance Pillar Score. The ESG Score records the highest maximum correlation in both uncontrolled (0.70) and controlled (0.81) returns, indicating situations where the ESG Score and market performance move together to a notable degree. Similarly, the Governance Pillar Score exhibits notable peak correlations (0.69 uncontrolled and 0.79 controlled), reflecting instances of concurrent movements between governance factors and return correlations. The Environmental Pillar Score also shows a particularly high maximum correlation in controlled returns (0.84).

Figure 3 represents the distribution of correlation between ESG metrics and annualized

controlled log returns. The score distributions appear to be right skewed across all metrics.

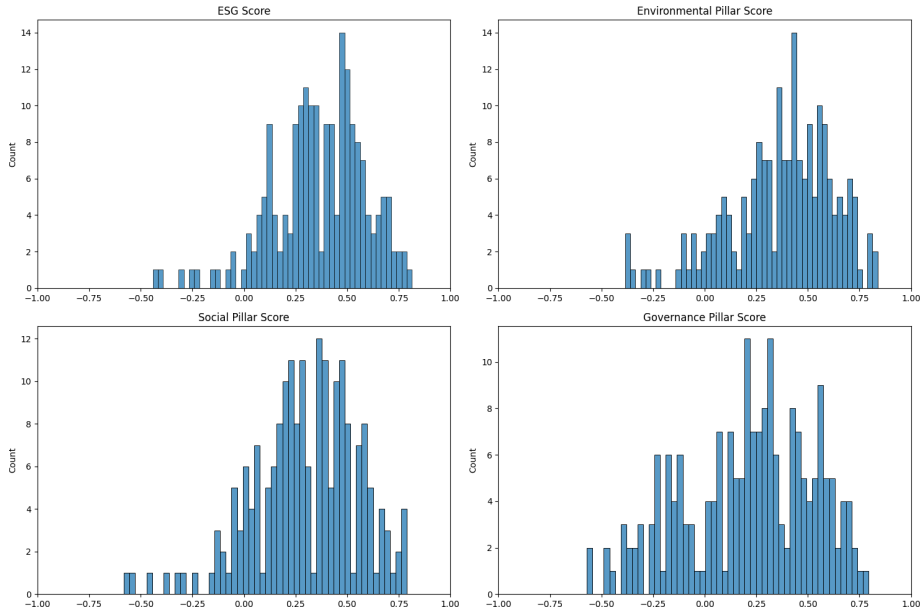


Figure 3: Distribution of Correlation between ESG Metrics and Controlled Annualized Log Return.

Table 4 presents the number of companies that have statistically significant correlation for a threshold at $p < 0.05$. The Environmental Pillar Score and ESG Score, being the most correlated, naturally have a higher number of statistically significant correlations. Since the effect is weaker in the Social Pillar Score and Governance Pillar Score, there are fewer companies with a statistically significant correlation.

Table 4: Statistically Significant Correlation between Controlled Annualized Returns and ESG Metrics

Metric	Correlation Significant Count	% of Companies	≤ 0 Count	≥ 0 Count
ESG Score	77	38.69%	0	77
Environmental Pillar Score	81	40.70%	0	81
Social Pillar Score	56	28.14%	3	53
Governance Pillar Score	53	26.63%	3	50

4.2. Correlation between ESG Metrics and Returns by Sectors

A deeper dive into the distribution of correlations within each sector illuminates the significant spread and variability. Heatmaps were used to highlight sectors and metrics with the most significant correlations. All subsequent heatmaps are on the same color ranging from [-0.5, 0.5]. Table C.8 from Appendix C displays the full data.

Figure 4 displays the heatmap of correlations between controlled annualized log returns and ESG metrics by sector. The returns have overall a weak to medium positive correlation with the ESG metrics. Financial Services exhibit a correlation of 0.46 with the ESG Score, indicating a notable association. The Healthcare sector shows a strong correlation as well, with a 0.41 correlation to the ESG Score, and is similarly aligned with the Environmental and Social Pillar Scores at 0.39 and 0.31 respectively. On the lower end, the Technology sector shows a distinctively weaker correlation, particularly with the Governance Pillar Score at 0.12. Consumer Cyclical stands out with a 0.40 correlation to the ESG Score and a 0.34 correlation to the Governance Pillar Score. Communication Services also demonstrate a substantial correlation with the ESG Score at 0.42.

The Environmental sector appears to be the most correlated and driving the correlation of the general ESG Score. The Social Score is also quite correlated among most sectors, leaving the Governance score as clearly the least correlated. The Industrial sector illustrates this, with an average 0.03 correlation with the Governance score.

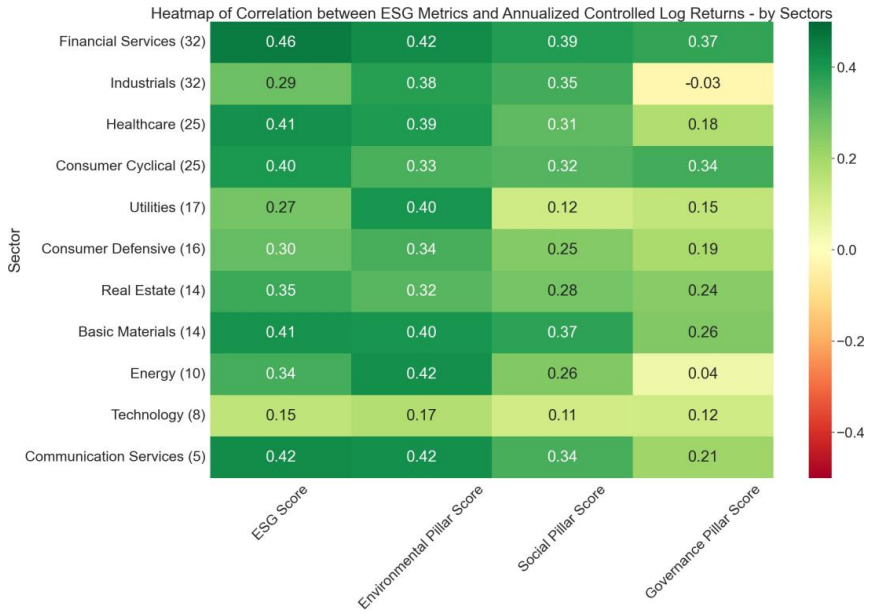


Figure 4: Heatmap of Correlations between Controlled Annualized Log Returns and ESG Metric by Sector.

4.3. Correlation between ESG Metrics and Returns by Materiality

In order to provide further granularity in this study, the 26 materiality issues isolated by SASB were used to group companies together. The correlation between returns and diverse ESG metrics is then computed individually for each company. These correlations are then averaged across all the companies presenting the corresponding SASB issue.

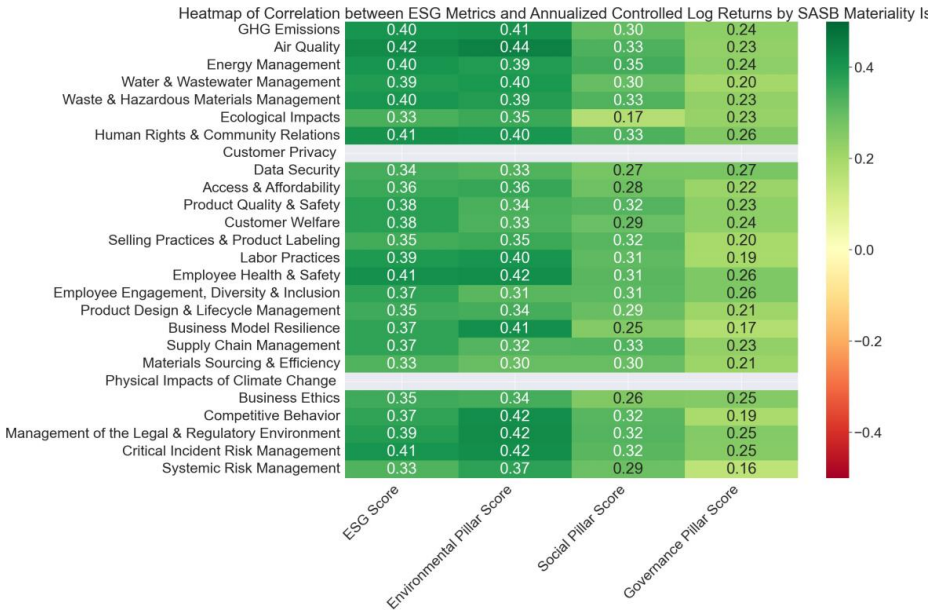


Figure 5: Heatmap of Correlations Between Controlled Annualized Log Return and ESG Metrics.

Figure 5 displays the heatmap of correlations between controlled annualized log returns and ESG metrics by SASB materiality issues. Once controlled, the correlation appears to be much stronger on a score-by-score comparison. Overall, the ESG metrics maintain a generally positive correlation with the returns. Specific SASB issues such as GHG Emissions (0.40), Air Quality (0.42), and Human Rights & Community Relations (0.41) show a notable association with the ESG Score. These issues, along with Critical Incident Risk Management (0.41), are among the most correlated within the Environmental Pillar Score. This correlation underscores the significance of these environmental and social issues in relation to financial performance.

In the context of the Social Pillar Score, Human Rights & Community Relations (0.40) and Employee Health & Safety (0.42) emerge as highly correlated issues, reflecting the importance of these aspects in corporate social responsibility. Additionally, the Governance Pillar Score reveals a strong correlation with issues such as Management of the Legal & Regulatory Environment (0.39) and Critical Incident Risk Management (0.42), which are integral to governance and risk oversight within organizations.

5. Results - Correlation between Variations of Returns and ESG Metrics

In this section, the correlation between the variations year per year of returns and ESG metrics is calculated over the integrality of companies. The variations are then normalized to avoid scale effect. Table 5 presents the global results, which shows an overall neutral correlation. The study is then refined with sectors and materiality issues. Finally, a time-lagged version of the correlations is proposed to explore potential delays between ESG initiatives and returns. When calculating the variations, every company out of the 491 that has a history ≥ 2 is used, bringing the number of companies considered to 263.

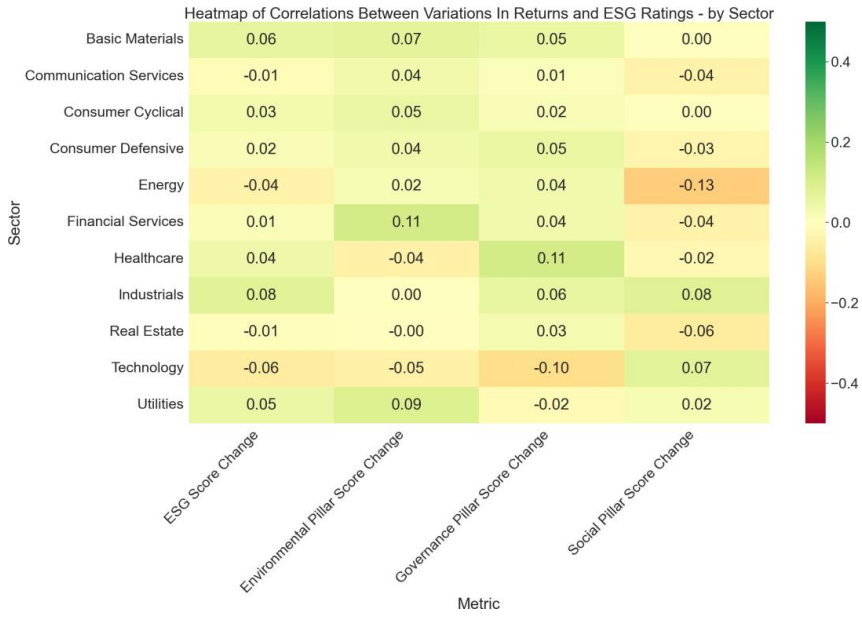
Table 5: Correlation between Variations of Returns and ESG Metrics

Metric	Correlation	P-Value
ESG Score Change	0.027	9.13e-10
Social Pillar Score Change	-0.005	2.34e-01
Governance Pillar Score Change	0.037	8.56e-17
Environmental Pillar Score Change	0.029	2.77-11

5.1. Correlation between Variations of Returns and ESG Metrics by Sectors

Figure 6 breaks down the correlation between the variation in returns and ESG metrics for companies in a given sector. The correlation between the variations appears to be neutral. The coefficients remain weak, with the Energy sector having a slightly higher negative correlation with the variation in the Social Pillar score. The Technology sector also has a slight negative correlation with the changes in Governance score. The Financial Services sector has the highest correlation between variations of returns and Environmental Pillar Score Change with 0.11. The Healthcare sector has the highest correlation with the Governance Pillar Change. The strongest correlation on the heatmap is the Energy sector with the Social Pillar Score, standing at -0.13.

Figure 6: Heatmap of Correlations Between Variations of Returns and ESG Metrics by Sectors.



5.2. Correlation between Variations of Returns and ESG Metrics by Materiality

Figure 7 breaks down the correlation between the variation in returns and ESG metrics for companies for a given materiality issue. The correlation remains weak at this level too, indicating that there is no linear relationship identifiable between the variations in returns and variations in ESG metrics at the granularity studied in this article.

Figure 7: Heatmap of Correlations Between Variations of Returns and ESG Metrics by SASB Materiality Issues.

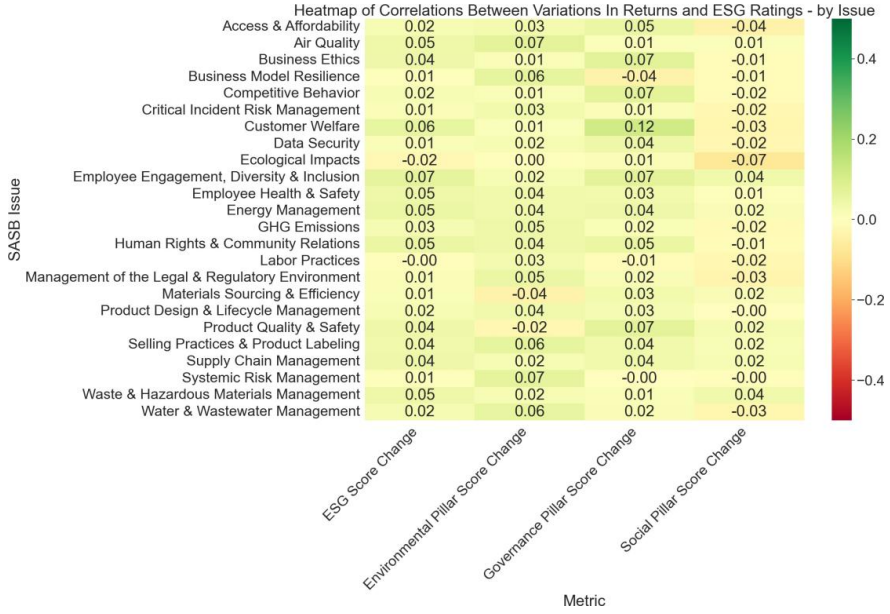


Figure 8 breaks down the correlation between the variation in returns and ESG score for companies for a given materiality issue with different time lags. Time lag for a given year t is defined as the correlation between $ESG_{(t+Lag)}$ with the returns R_t, with

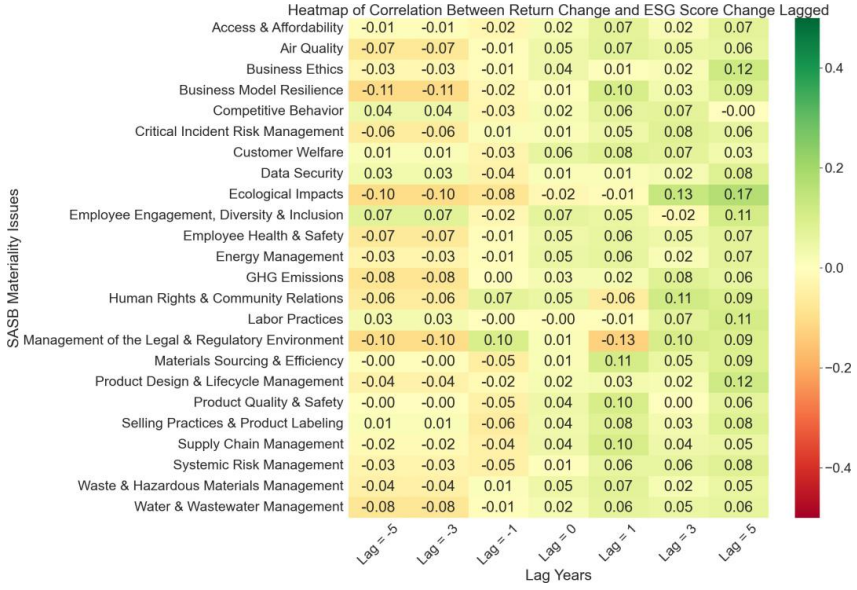
Lag = 5, 3, 1, 0, 1, 3, 5. Negative values of Lag effectively represent lagging the return variations as opposed to lagging the rating variations. When the ESG ratings variations are lagged by one, three or five years, the correlation with the changes in returns is neutral for most issues. Business Model Resilience, Supply Chain Management and Materials Sourcing & Efficiency appear to be the most correlated after one year. Ecological impact and Business Ethics are weakly positively correlated after 5 years. Management of the Legal & Regulatory Environment is the least correlated issue after one year.

When the returns are delayed, the correlation is weak to neutral with a one-year delay. When delayed three years, there is a weak negative correlation for certain issues, including Supply Chain Management, Customer Welfare and Selling Practices & Product Labelling. At a five-year delay, there is a weak negative correlation, with the most correlated being

Management of the Legal & Regulatory Environment and Business Model Resilience. The other issues remain neutral.

The introduction of time-lagged analysis in these correlations reveals interesting cross-correlation dynamics between ESG scores and company returns. Cross-correlation in time series analysis helps in understanding how two variables, like ESG scores and returns, are related and interact over different time lags. In this context, it suggests that ESG factors may not change at the same time as return but could have similar variations over extended periods. This cross-correlation analysis is particularly insightful for identifying which ESG factors change in a similar fashion to returns. For instance, issues like Labor Practices and Employee Engagement show stronger correlations at different time lags, suggesting that the effect of these ESG aspects on financial performance unfolds over a longer horizon.

Figure 8: Heatmap of Correlations Between Variations of Returns and ESG Score by SASB Materiality Issues with Time-Lag *Lag*. At *Lag* = 5 variations in returns are effectively lagged by 5 years and at *Lag* = 5 variations in ESG Score are lagged by 5 years. *Lag* = 0 corresponds to no lag.



6. Discussion and analysis of the results

There is a correlation between the Thomson Reuters ESG ratings and the returns. This correlation can further be refined by filtering down by sectors and materiality to better highlight which group of companies is more exposed. Certain sectors such as Technology have weak positive correlation on average, while Financial Services companies have a stronger correlation. Materiality issues further magnify those discrepancies, by emphasizing the difference in relevance between the pillar scores, ranging from the moderately correlated Environmental score to the weakly correlated Governance score.

A first observation that can be drawn from the data is that the controlled have a much higher correlation with the ESG ratings than the uncontrolled returns. One possible explanation could be that the market factors trimmed using Fama-French 5 acted as signal noise between the two variables.

Another notable finding is the absence of correlation between the variation of ESG ratings and variation of annualized log returns, regardless of controlling. This indicates that ESG ratings and annualized log returns tend to not increase or decrease concurrently year over year. A plausible explanation could be a parallel with a company acting for growth or for profit. It was evidenced in previous study (Lu and Beamish, 2006) that a company may experience higher growth with stagnating profitability and vice-versa depending on the business plan. Increasing sustainability or log returns in one's business requires concentrated efforts and could have an opportunity cost in the other area.

It was observed in this study that there appears to be no significant correlation at lag periods of one, three, or five years. This lack of correlation over time suggests that the impact of ESG factors on financial performance might not be immediate, but rather indirect or lagged. It raises critical questions about the temporal nature of ESG integration in financial analysis. One hypothesis could be that the benefits of high ESG ratings, such as enhanced reputation, better stakeholder engagement, and risk mitigation, may appear over a longer period. This delay could also be indicative of the market's slow adjustment, reflecting a lag in the incorporation of ESG considerations into investment decisions.

In the context of asset pricing, the relationship between ESG ratings and returns can be understood through the lens of systematic versus unsystematic risk. Systematic risk, which affects the entire market or a large segment of the market, can be paralleled by broad environmental concerns that impact multiple industries, while unsystematic risk is specific to individual companies or sectors. For instance, the stronger correlation of the Environmental metric in the Industrials sector could suggest that governance practices are a significant source of unsystematic risk, affecting firm-specific returns and investment decisions.

Materiality becomes particularly relevant when considering the correlation of ESG

metrics with sector performance. Material ESG factors vary by industry and can have a direct impact on a firm's risk profile and cost of capital. For example, environmental risks are highly material for the companies recognizing the Air Quality and Employee Health & Safety issues, suggesting that a high Environmental Pillar Score might display an attempt at mitigating those risks, potentially lowering the cost of equity for firms with strong environmental practices.

The regulatory environment is another factor to consider, as it can significantly affect company risk. Firms with high Governance Pillar Scores may be better prepared to face upcoming regulations. In sectors like Financial Services, for instance, the slight negative correlation with Governance Pillar Scores could reflect a market perception that less regulated firms might experience short-term gains. However, this could expose investors to higher long-term risks, if regulatory scrutiny were to increase.

In the long-term investment horizon, High scores in ESG ratings can signal a company's commitment to sustainability and resilience, which can be crucial for long-term value creation. This is particularly relevant for metrics such as Business Model Resilience, that presents a 0.41 correlation with returns on average, suggesting that companies prioritizing long-term sustainability initiatives may enjoy more stable returns over time.

ESG ratings can aid in portfolio construction and diversification. By using ESG scores to identify companies that are potentially less exposed to ESG-related risks or are better managed, investors could reduce the risk profile of their portfolios and enhance their resilience to market shocks driven by ESG factors.

7. Conclusion

This study evaluated the correlation between annualized log returns and Thomson Reuters ESG metrics among the companies in the S&P500. Annualized log returns were controlled using Fama- French 5 to remove market factors. The correlation between the variations of both data year by year was also computed. Our findings indicate an overall positive correlation between controlled log returns and ESG metrics. Sector-specific analysis revealed that the company sector does influence the relationship between ESG metrics and returns. Finally, incorporating materiality issues enhances the explanatory power of the ESG-returns correlation by focusing on the most relevant ESG factors for each sector and therefore refining the correlation analysis. This highlights the importance of sector-specific and company specific ESG considerations in financial analysis, aligning with contemporary asset pricing models.

Appendix A. Dataset Summary Tables

Table A.6 presents the different dataset fields with their type and frequency. Noticeably, the ESG metrics are the temporal bottleneck, as the scores provided by Reuters are updated yearly.

Table A.6: Summary of Dataset Variables

Variable	Nature	Frequency	Description
Date	Time-series	Daily	Date of data entry.
Instrument	Categorical	-	Specific stock or financial instrument.
Sector	Categorical	-	GICS sector of the company.
Open Price	Continuous	Daily	Opening price of the stock.
Close Price	Continuous	Daily	Closing price of the stock.
High Price	Continuous	Daily	Highest price of the stock during the day.
Low Price	Continuous	Daily	Lowest price of the stock during the day.
ESG Score	Continuous	Yearly	Aggregate ESG rating.
Environmental Pillar Score	Continuous	Yearly	Rating based on environmental practices.
Social Pillar Score	Continuous	Yearly	Rating based on social responsibilities.
Governance Pillar Score	Continuous	Yearly	Rating based on governance structures.
Rm-Rf	Continuous	Daily	Excess return on the market (FF5).
SMB	Continuous	Daily	Small Minus Big (FF5).
HML	Continuous	Daily	High Minus Low (FF5).
RMW	Continuous	Daily	Robust Minus Weak (FF5).
CMA	Continuous	Daily	Conservative Minus Aggressive (FF5).

Table A.7 presents the different flags identified by SASB as the material issues. These flags are represented as a binary vector in the dataset, indicating whether a company is sensitive to a given issue.

Table A.7: Summary of SASB Flags

Variable	Category
GHG Emissions	Environment
Air Quality	Environment .
Energy Management	Environment
Water & Wastewater Management	Environment
Waste & Hazardous Materials Management	Environment
Ecological Impacts	Environment
Human Rights & Community Relations	Social Capital
Customer Privacy	Social Capital
Data Security	Social Capital
Access & Affordability	Social Capital
Product Quality & Safety	Social Capital
Customer Welfare	Social Capital
Selling Practices & Product Labeling	Social Capital
Labor Practices	Human Capital
Employee Health & Safety	Human Capital .
Employee Engagement, Diversity & Inclusion	Human Capital
Product Design & Lifecycle Management	Business Model and Innovation
Business Model Resilience	Business Model and Innovation
Supply Chain Management	Business Model and Innovation
Materials Sourcing & Efficiency	Business Model and Innovation
Physical Impacts of Climate Change	Business Model and Innovation
Business Ethics	Leadership and Governance
Competitive Behavior	Leadership and Governance
Management of the Legal & Regulatory Environment	Leadership and Governance
Critical Incident Risk Management	Leadership and Governance
Systemic Risk Management	Leadership and Governance

Appendix B. Maximum correlation interval

Pearson's correlation takes a value between $[-1, 1]$, but this is only true if the two random vectors X_1 and X_2 are of the same type (Embrechts et al., 2011). To provide further interpretability to the coefficent, the maximum attainable interval is calculated below. Starting with the upper bound, the first step is to calculate the covariance. Let $z \sim \mathcal{N}(0,1)$, then $X = \lambda \sinh\left(\frac{z-\gamma}{\delta}\right) + \xi$ and $Y = \sigma z + \mu$.

$$\begin{aligned}\operatorname{cov}(X,Y) &= \mathrm{E}[(X-\mathrm{E}[X])(Y-\mathrm{E}[Y])] \\ &= \operatorname{cov}\left(\lambda\sinh\left(\frac{z-\gamma}{\delta}\right)+\xi, \sigma z+\mu\right) \\ &= \mathrm{E}\left(\lambda\sinh\left(\frac{z-\gamma}{\delta}\right)+\xi-\mathrm{E}\left(\lambda\sinh\left(\frac{z-\gamma}{\delta}\right)+\xi\right)\right)(\sigma z+\mu-\mathrm{E}(\sigma z+\mu)) \\ &= \lambda\sigma\mathrm{E}\left(\operatorname{zsinh}\left(\frac{z-\gamma}{\delta}\right)\right)\end{aligned}$$

Let $z=\delta w$ where $\delta>0$ and $\alpha=\frac{\gamma}{\delta}$ so $w\sim\mathcal{N}\left(\frac{0,1}{\delta^2}\right)$ and $\mathrm{E}\left[\frac{\operatorname{zsinh}(z-\gamma)}{\delta}\right]$

$$\begin{aligned}&= \int_{-\infty}^{\infty} \operatorname{wexp}\left(\pm w-\frac{\delta^2 w^2}{2}\right)dw \\ &= \exp\left(\frac{1}{2\delta^2}\right)\int_{-\infty}^{\infty} \operatorname{wexp}\left(-\frac{\delta^2\left(w\mp\frac{1}{\delta^2}\right)^2}{2}\right)dw = \pm\exp\left(\frac{1}{2\delta^2}\right)\frac{\sqrt{2\pi}}{\delta^3}\end{aligned}$$

using standard Gaussian integral identities:

$$\mathrm{E}\left[z\sinh\frac{z-\gamma}{\delta}\right] = \exp\left(\frac{1}{2\delta^2}\right)\frac{\exp(-\alpha)+\exp(\alpha)}{2\delta} = \frac{\exp\left(\frac{1}{2\delta^2}\right)}{\delta}\cosh\frac{\gamma}{\delta}.$$

Finally,

$$\operatorname{cov}(X,Y) = \frac{\lambda\sigma}{\delta}\exp\left(\frac{1}{2\delta^2}\right)\cosh\frac{\gamma}{\delta}$$

Equation 7

The process is now repeated for the lower bound with $z\sim\mathcal{N}(0,1), X=\lambda\sinh\left(\frac{z-\gamma}{\delta}\right)+\xi$ and $Y=-\sigma z+\mu$.

$$\operatorname{cov}(X,Y) = -\frac{\lambda\sigma}{\delta}\exp\left(\frac{1}{2\delta^2}\right)\cosh\frac{\gamma}{\delta}$$

Equation 8

We now have the variance for both distributions:

$$\mathrm{Var}(X) = \frac{\lambda^2}{2}\left(\exp(\delta^{-2}) - 1\right)\left(\exp(\delta^{-2})\cosh\left(\frac{2\gamma}{\delta}\right) + 1\right)$$
$$\mathrm{Var}(Y) = \sigma^2$$

Equation 9

Leading to $[\rho_{\min}, \rho_{\max}\ ]$, with

$$\rho_{\min} = -\frac{\frac{\lambda\sigma}{\delta}\exp\left(\frac{1}{2\delta^2}\right)\cosh\frac{\gamma}{\delta}}{\sqrt{\left(\frac{\lambda^2}{2}\left(\exp(\delta^{-2}) - 1\right)\left(\exp(\delta^{-2})\cosh\left(\frac{2\gamma}{\delta}\right) + 1\right)\right)(\sigma^2)}}$$

Equation 10

And

$$\rho_{\max} = \frac{\frac{\lambda\sigma}{\delta}\exp\left(\frac{1}{2\delta^2}\right)\cosh\frac{\gamma}{\delta}}{\sqrt{\left(\frac{\lambda^2}{2}\left(\exp(\delta^{-2}) - 1\right)\left(\exp(\delta^{-2})\cosh\left(\frac{2\gamma}{\delta}\right) + 1\right)(\sigma^2)\right.}}$$

Equation 11

For a normal fit to the controlled returns of (μ, σ)= 0.03, 0.06) and a Johnson SU's fit to the ESG metrics of $(\lambda, \gamma, \delta, \xi)$ = (1.39, −1.22, 7.91, −0.49) the calculations indicate $[\rho_{min}, \rho_{max}]$ = [−0.9998, 0.9998].

Appendix C. Sector-Wise Correlation

Table C.8: Correlation between Controlled Log Returns and ESG Metrics. Legend: ESG - ESG Score, EPS - Environmental Pillar Score, SPS - Social Pillar Score, GPS - Governance Pillar Score

	ESG	EPS	SPS	GPS
Basic Materials	0.39	0.32	0.30	0.35
Communication Services	0.17	0.26	0.21	-0.08
Consumer Cyclical	0.29	0.29	0.21	0.11
Consumer Defensive	0.31	0.27	0.26	0.27
Energy	0.08	0.17	-0.03	0.06
Financial Services	0.18	0.29	0.17	0.09
Healthcare	0.20	0.14	0.15	0.15
Industrials	0.24	0.23	0.20	0.20
Real Estate	0.16	0.20	0.09	0.01
Technology	0.10	0.15	0.10	0.07
Utilities	0.35	0.35	0.26	0.16

References

Alessandrini, F. and Jondeau, E. (2021). Optimal strategies for esg portfolios. *The Journal of Portfolio Management*, 47(6):114–138.

Bauer, R., Koedijk, K., and Otten, R. (2009). International evidence on ethical mutual fund performance and investment style. *Journal of Banking Finance*, 33(2):252–262.

Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654.

Busco, C., Consolandi, C., Eccles, R. G., and Sofra, E. (2020). A preliminary analysis of sasb reporting: Disclosure topics, financial relevance, and the financial intensity of esg materiality. *Journal of Applied Corporate Finance*, 32(2):117–125.

Danciu, V. (2013). The sustainable company: new challenges and strategies for more sustainability. *Theoretical and Applied Economics*, 20(9):7–26.

Derwall, J., Guenster, N., Bauer, R., and Koedijk, K. (2007). The economic value of

corporate eco-efficiency. *European Financial Management*, 13(5):684–717.

Dor, A. B., Guan, J., and Sun, Y. (2022). Is incorporating esg considerations costly? *The Journal of Portfolio Management*, 48(7):75–87.

Eccles, R. G., Ioannou, I., and Serafeim, G. (2014). The impact of corporate sustainability on organizational processes and performance. *Management Science*, 60(11):2835–2857.

Embrechts, P., Frey, R., and McNeil, A. (2011). Quantitative risk management.

Fafaliou, I., Giaka, M., Konstantios, D., and Polemis, M. (2022). Firms' esg reputational risk and market longevity: A firm-level analysis for the united states. *Journal of Business Research*, 149:161–177.

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

Fama, E. F. and French, K. R. (2015). A five-factor asset pricing model. *Journal of Financial Economics*, 116(1):1–22.

Fatemi, A., Glaum, M., and Kaiser, S. (2018). Esg performance and firm value: The moderating role of disclosure. *Global Finance Journal*, 38:45–64.

French, K. (2022a). Data library ff5. https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html#International.

French, K. R. (2022b). Kenneth r. french website. https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/index.html.

Friede, G., Busch, T., and Bassen, A. (2015). Esg and financial performance: aggregated evidence from more than 2000 empirical studies. *Journal of sustainable finance & investment*, 5(4):210–233.

Friedman, M. (1970). The social responsibility of business is to increase its profits. In *Corporate ethics and corporate governance*, pages 173–178. Springer.

Gibson Brandon, R., Krueger, P., and Schmidt, P. S. (2021). Esg rating disagreement and stock returns. *Financial Analysts Journal*, 77(4):104–127.

GICS (2022). Gics.
https://www.spglobal.com/spdji/en/landing/topic/gics/.

Giese, G., Lee, L.-E., Melas, D., Nagy, Z., and Nishikawa, L. (2019). Foundations of esg investing: How esg affects equity valuation, risk, and performance. *The Journal of Portfolio Management*, 45(5):69–83.

Grewal, J., Hauptmann, C., and Serafeim, G. (2021). Material sustainability information and stock price informativeness. *Journal of Business Ethics*, 171:513–544.

Khan, M., Serafeim, G., and Yoon, A. (2016). Corporate sustainability: First evidence on materiality. *The accounting review*, 91(6):1697–1724.

Lee, M. T., Raschke, R. L., and Krishen, A. S. (2022). Signaling green! firm esg signals in an interconnected environment that promote brand valuation. *Journal of Business Research*, 138:1–11.

Li, D. and Ng., W.-L. (2000). Optimal dynamic portfolio selection: Multiperiod mean-variance formulation. *Mathematical finance*, 10(3):387–406.

Lintner, J. (1965). The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. *The Review of Economics and Statistics*, pages 13–37.

Lu, J. W. and Beamish, P. W. (2006). Sme internationalization and performance: Growth vs. profitability. *Journal of international entrepreneurship*, 4:27–48.

Panna, M. (2017). Note on simple and logarithmic return. *APSTRACT: applied studies in agribusiness and commerce*, 11(1033-2017-2935):127–136.

Research, G. V. (2018). [14] gsia. global sustainable investment alliance investment review. 2018. *Available online: http://www.gsi-alliance.org/wpcontent/uploads/*, 2018:3.

SASB (2022). Sasb.
https://sasb.org/.

Schooley, D. K. and English, D. M. (2015). Sasb: A pathway to sustainability reporting in the united states. *The CPA journal*, 85(4):22.

Sharpe, W. F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. *The Journal of Finance*, 19(3):425–442.

Starks, L. T., Venkat, P., and Zhu, Q. (2017). Corporate esg profiles and investor horizons. *Available at SSRN 3049943*.

UN (2020). Take action for the sustainable development goals – united nations sustainable development.

Van Duuren, E., Plantinga, A., and Scholtens, B. (2016). Esg integration and the investment management process: Fundamental investing reinvented. *Journal of Business Ethics*, 138(3):525–533.

Appendix B

Article 2 - Centralized Multi-Agent Proximal Policy Optimization with Attention [208]

Centralized Multi-Agent Proximal Policy Optimization with Attention

Hugo Cazaux*†, Ralph Rudd*, Hlynur Stefánsson*, Sverrir Ólafsson*, Eyjólfur Ingi Ásgeirsson*
*Department of Engineering, Reykjavik University, Menntavegur 1, 102 Reykjavik, Iceland
†Corresponding author: hugot20@ru.is, Menntavegur 1, 102 Reykjavik, Iceland

***Abstract*—This paper introduces a novel centralized multi-agent framework employing proximal policy optimization (PPO), a state-of-the-art reinforcement learning algorithm. Multiple subagents focus on different aspects of the environment, and the actions suggested by the subagents are used to augment the environment space for a superagent that encompasses the subagents' different approaches. A key feature of this model is the built-in attention module that balances the weight attributed to the environment variables and the suggested actions. This architecture is designed to promote emergent behaviour and exploration, completing the standard training of a PPO agent. We study the efficiency and decision accuracy from the model across several MuJoCo scenarios, and perform an ablation study to demonstrate the influence of the attention mechanism. Our results indicate a trade-off in performance linked to the dimensions of the action space and address the ideal use case for this framework.**

***Index Terms*—Machine Learning, Proximal Policy Optimization, Reinforcement Learning, Attention**

I. Introduction

In the current landscape of machine learning, the complexity and volume of data require innovative solutions to harness the most out of a dataset. This paper aims to lay down the foundations of a multi-agent theoretical framework, specifically a centralized multi-agent proximal policy optimization approach. Proximal policy optimization (PPO) [1] is presently considered state of the art in reinforcement learning, a subset of machine learning. The framework presented here harnesses the decision-making of several subagents tuned to each prioritize certain aspects of the dataset. These decisions are then ultimately processed by a superagent, tasked with synthesizing the opinion of each subagent and reaching the final action. The superagent is equipped with an attention module that dynamically balances between environment variables and subagent input. From this decision-making process comes the term *centralized*, there is no communication between the subagents and the resulting subactions are part of the superagent observation space.

The core research questions developed in this article are the following: how can one build a resilient and versatile artificial intelligence framework using a centralized multi-agent approach? What are the pros and cons of this approach in terms of performance, sample-efficiency and interpretability?

The paper is structured as follows: Section 2 includes a contextualization and a review of the existing literature. Section 3 introduces the mathematical definition of the framework. Section 4 presents the results of the framework against standard strategies and alternative models. Section 5 proposes an ablation study. Section 6 discusses training time. Section 7 develops interpretability tools for the model. Section 8 documents the hardware used. Section 9 is the conclusion to this study.

II. Background

PPO is a policy gradient method developed by John Schulman et al. in 2017 [1]. The key innovation of this algorithm over older methods such as TRPO [2] or ACER [3] is the clip function that constrains policy updates of the agent. PPO has been used in a wide variety of applications: Atari games [4], track racing games [5], suspension monitoring for cars [6], and image captioning [7]. A number of articles have proposed innovations to the base algorithm, for instance an alternative minimization target [8], [9] introduced policy feedback; specifically improving early learning stages, which are recognized as a potential weak point of PPO [10]. Recently proposed improvements include a shift in learning to offline policy optimization [11] and including conservatism [12].

Multi-agent methods have gained significant attention in the field of reinforcement learning, particularly for their capability to simulate complex systems involving interactive agents. A notable early work in multi-agent systems is [13] which explored the dynamics of cooperative and competitive agents in a shared environment. Recent advancements have integrated PPO into multi-agent applications: [14] applied multi-agent PPO to competitive and cooperative tasks, [15] successfully employed multi-agent reinforcement learning in the complex environment of the Dota 2 game. The integration of PPO into multi-agent systems has also been explored in real-world scenarios such as traffic light control [16], and collaborative robotics [17]. Innovations specific to multi-agent PPO include [18], which introduced a meta-learning approach to enhance adaptability across different tasks and agent configurations and [19], which presented the concept of leniency in multi-agent learning, mitigating the non-stationary issue commonly faced in such environments.

Attention is a machine learning mechanism designed to imitate human awareness. Attention was brought to the forefront of the field with the transformer architecture, a self-attention-based architecture that enabled the recent breakthroughs in

large language models [20]. It has since seen many implementations including in recurrent neural networks for search results customization [21], missing data imputation [22], and in computer vision [23]. In reinforcement learning, attention models have been developed within theoretical frameworks [24] and diverse applications, such as source code summarizing [25], dynamic graph problems [26], and road networks management [27].

The novelty of this model lies in the combination of staple reinforcement learning concepts. Multi-agents models have been explored in adversarial and cooperative settings, but to our knowledge not in an independent centralized manner. The addition of the attention module to the superagent provides avenues of interpretability and fine-tuning for the model that were not previously studied. The method-agnostic nature of this design also increases its potential for future studies, as this study only explores its application with PPO. This study also creates an opportunity for further applications of the design in simulated environment encompassing diverse fields.

III. Framework Details

As mentioned in [28], implementation is key in deep policy gradient algorithms. As such, the framework below is implemented using the clean-rl library [29].

A. Proximal Policy Optimization (PPO)

- Policy Function: For an agent x, its policy at time t is a probability density function denoted as $\pi_\theta(a_t|o_t)$, where θ are the parameters of the policy, o_t is the observation for agent x at time t, and a_t are the actions that can be taken. The policy is then sampled to obtain the action taken $\alpha_t \sim \pi_\theta(a_t|o_t)$.
- Objective Function: The PPO objective function is defined as:

  $$L^{PPO}(\theta) = \mathbb{E}_t \left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t) \right]$$

  where $r_t(\theta) = \frac{\pi_\theta(a_t|o_t)}{\pi_{\theta_{\text{old}}}(a_t|o_t)}$ is the probability ratio, ϵ an hyperparameter and $\hat{A}_t$ is an estimator of the advantage at time t, typically computed using Generalized Advantage Estimation (GAE).
- Advantage Estimation: The advantage $\hat{A}_t$ is computed as:

  $$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \ldots + (\gamma\lambda)^{T-t+1}\delta_{T-1} \quad (1)$$

  with $\delta_t = r_t + \gamma V(o_{t+1}) - V(o_t)$ and V a learned state-value function.
- **Training Process:** The agent is trained by iteratively updating its policy parameters. This involves:
  1) Collecting trajectories by interacting with the environment using the current policy.
  2) Estimating the advantages using GAE.
  3) Calculating the surrogate objective function.
  4) Optimizing the surrogate objective function using gradient ascent while ensuring the updates stay within a specified clipping range to maintain policy stability.

B. Centralized Multi-Agent Model

- Centralization: Each subagent is trained independently on its own environment. The action taken by a subagent on its given environment does not influence the other environments, and as such there is no communication between the subagents. The local observation for agent x_i at time t is represented by $o_{t,i}$.
- Policy Representation: The policy of an agent i is $\pi_{\theta_i}(a_{t,i}|o_{t,i})$.
- Sampling of the Policy: $\alpha_t, i \sim \pi_{\theta_i}(a_{t,i}|o_{t,i})$ where α_t, i is the action taken by agent i at time t.
- Reward Function: Each agent x_i has its own reward function $R_i(o_t, a_{t,i})$.
- Training Process: Agents are trained iteratively, updating their policy parameters using the PPO objective function.

C. Superagent Decision-Making Model

- Superagent's role: The superagent x_f makes the overarching decision, influenced by the decisions of the subagents $\{x_1, x_2, \ldots, x_n\}$ and the current state of the environment o_t.
- Aggregation Function: the aggregation function $\mathcal{F}$ is a linear or non-linear function that combines the outputs of the subagents and the current state of the environment:

  $$s_t^f = \mathcal{F}(\alpha_{t,1}, \alpha_{t,2}, \ldots, \alpha_{t,n}, o_t; \phi) \quad (2)$$

  where s_t^f is the state at time t, and ϕ are the parameters of the aggregation function.
- Final Decision-Making Policy: The superagent's policy $\pi_{\theta_f}(a_t^f, s_t^f)$ is then sampled to produce the final action α_t^f.

D. Attention Mechanism in Decision-Making

To enhance the decision-making process, an attention mechanism is integrated into the superagent's framework. This mechanism is designed to dynamically prioritize the influence of subagent actions and the environmental state on the final decision-making process.

- **Attention Module Construction**: The attention module consists of two main components:
  - Linear transformations that compute the attention scores for environmental states and subagent actions respectively, denoted as f_{env} and f_{sub}.
  - A softmax layer that normalizes these scores to form attention weights.
- **Input Representation**: Let e_t represent the encoded environmental state and $\alpha_{t,1}, \alpha_{t,2}, \ldots, \alpha_{t,n}$ represent the actions taken by the subagents at time t. z_{env} and z_{sub} are the linearly transformed environmental state and actions. These are processed through their respective linear layers:

  $$z_{\text{env}} = f_{\text{env}}(e_t; \theta_{\text{env}}), \quad (3)$$

  $$z_{\text{sub}} = f_{\text{sub}}([\alpha_{t,1}, \alpha_{t,2}, \ldots, \alpha_{t,n}]; \theta_{\text{sub}}), \quad (4)$$

where θ_{env} and θ_{sub} are the parameters of the linear transformations for the environment and subagent actions, respectively.

- **Attention Weights Calculation**: The attention weights w_{env} and w_{sub} are computed as follows:

$$[w_{\text{env}}, w_{\text{sub}}] = \text{softmax}\left([z_{\text{env}}, z_{\text{sub}}]\right). \tag{5}$$

These weights determine the relative influence of the environmental states and the subagent actions on the decision-making process of the superagent.

- **Feature Aggregation**: The weighted sum of features, influenced by the calculated attention weights, forms the input to the decision-making layers of the superagent:

$$d_t^f = w_{\text{env}} \cdot e_t + w_{\text{sub}} \cdot [\alpha_{t,1}, \alpha_{t,2}, \ldots, \alpha_{t,n}] \tag{6}$$

where d_t is the aggregated decision input for the superagent at time t.

- **Policy Decision**: The superagent uses d_t along with the state s_t to determine the appropriate action α_t^f through its policy network:

$$\alpha_t^f \sim \pi_{\theta_f}(a_t^f | d_t^f) \tag{7}$$

where θ_f are the parameters of the superagent's policy network.

- Finally, we can express the action taken by the superagent relative to the subagents' policy as:

$$\begin{aligned}\alpha_t^f \sim \pi_{\theta_f}(a_t^f | w_{\text{env}} \cdot e_t + w_{\text{sub}} \cdot [\alpha_{t,1} \sim \pi_{\theta_1}(a_{t,1}|o_{t,1}) \\ , \ldots, \alpha_{t,2} \sim \pi_{\theta_2}(a_{t,2}|o_{t,2}), \alpha_{t,n} \sim \pi_{\theta_n}(a_{t,n}|o_{t,n})])\end{aligned} \tag{8}$$

This attention mechanism allows the superagent to adaptively focus more on either the subagent actions or the environmental state based on the current scenario, enhancing the flexibility and effectiveness of the decision-making process.

IV. Results

Multi-Joint dynamics with Contact, commonly called MuJoCo [30], proposes several standard environments to train and benchmark models on. Three MuJoCo environments were selected as experimental settings. The three environments are: Hopper-v4, Half-Cheetah-v4 and Humanoid-v4. In these environments, the reward (R) is calculated using several factors: the forward reward (F_r) and control cost ($Ctrl_c$) are common to all tasks. The forward reward is the movement alongside the x-axis, while the control cost is a penalty for each action taken. The Hopper and Humanoid implement a healthy reward (H_r) that determines whether the action is damaging. The Humanoid also implements a contact cost ($Ctct_c$) that penalizes the agent if the contact force with the ground is too high. The subagents are then tested on the base environment over 10 epochs, and ranked by the average cumulative reward. Based on the ranking, eight superagents are then trained with an increasing number of subagents contributing to the observation space. All agents are trained over four million timesteps.

TABLE I: Reward function formulas

Environment	Reward formula
HalfCheetah-v4	$R = w_f \cdot F - w_{ctrl} \cdot Ctrl$
Hopper-v4	$R = w_f \cdot F + w_h \cdot H - w_{ctrl} \cdot Ctrl$
Humanoid-v4	$R = w_f \cdot F + w_h \cdot H - w_{ctrl} \cdot Ctrl - w_{ctct} \cdot Ctct$

Table I presents the reward formula for each environment. The term w_i is the weight for a reward term i.

A. Subagents Performance - Same Reward

In this experiment, eight agents were trained with identical reward functions. Each environment uses the default configuration, but a different random seed.

TABLE II: Subagents performance across HalfCheetah-v4, Hopper-v4, and Humanoid-v4

Subagent	Average Reward (HalfCheetah-v4)	Average Reward (Hopper-v4)	Average Reward (Humanoid-v4)
1	-719.19	1207.45	**3187.72**
2	-287.17	**1023.79**	2857.51
3	-388.11	1203.78	3050.35
4	-342.99	1172.62	2971.13
5	-667.03	1152.54	2988.99
6	**-145.95**	1212.92	**2801.29**
7	-180.96	1142.96	2984.01
8	**-898.91**	**1237.77**	3018.10

Table II shows the performance of subagents across the three test environments. Highest and lowest performing agents for each environment in bold. The average cumulative reward varies by environment, which indicates that the environment state in certain random seeds is more suited for policy gradient learning.

B. Superagents Performance - Same Reward

TABLE III: Superagents performance across environments

Number of Subagent(s)	Average Reward (HalfCheetah-v4)	Average Reward (Hopper-v4)	Average Reward (Humanoid-v4)
1	-239.99	1204.68	**2873.40**
2	-214.03	1267.98	2993.25
3	-323.44	**1336.50**	3057.08
4	**-180.76**	1207.91	2999.07
5	-380.27	1259.87	2931.13
6	-414.66	**969.09**	2974.15
7	-440.89	1211.66	2888.64
8	**-498.05**	1266.79	**3178.57**

Table III displays the average cumulative reward across the three environments. When using subagents with the same reward function, the performance slowly increases for HalfCheetah-v4, until four subagents. When adding more subagents, the model drops in performance. A similar pattern can be observed with Hopper-v4, which peaks at three subagents. The Humanoid-v4 sees no major gain or loss and remains stable across the board, despite a slight boost in performance at eight subagents. A possible explanation could be that using the same reward function, the subagents are unlikely to explore new behaviours that could then be passed to the superagent.

C. Subagents Performance - Mixed Reward

In this experiment, the reward function coefficients of the subagents were altered to promote emergent behaviours and exploration. In each subagent configurations table, the configurations in bold are the default settings of the environment.

TABLE IV: Subagent configurations for HalfCheetah-v4

(Forward reward weight, Control cost weight)	Average reward
(0.5, 0.5)	105.06
(1.0, 0.5)	-132.32
(1.0, 0.1)	-297.40
(0.5, 0.1)	-650.33
(1.0, 0.01)	-659.02
(1.0, 0)	-680.37
(1.0, 0.1)	**-707.54**
(2.0, 0.001)	-734.57
(3.0, 0.001)	-765.30

As shown in Table IV, the HalfCheetah-v4 altered configurations had a wide range of performance on the original environment. The average cumulative reward degraded significantly when the forward reward weight and control cost weight were changed. One explanation could be that techniques to move forward with a very high control cost might have been learnt by the last two subagents, which hindered their performance on the base environment.

TABLE V: Subagent configurations for Hopper-v4

(Forward reward weight, Control cost weight, Healthy reward weight, Healthy state range, End when unhealthy)	Average reward
(3.0, 0.0001, 0.0, [-100, 100], N)	1468.63
(0.5, 0.01, 2.0, [-100, 100], Y)	1402.84
(0.5, 0.0005, 1.0, [-100, 100], Y)	1267.66
(1.0, 0.001, 0.5, [-100, 100], Y)	1233.06
(1.0, 0.05, 1.0, [-100, 100], Y)	1227.56
(1.5, 0.01, 0.5, [-150, 150], Y)	1192.80
(2.0, 0.001, 0.5, [-100, 100], Y)	1161.34
(3.0, 0.0, 0.5, [-300, 300], Y)	1157.77
(1.0, 0.001, 1.0, [-100, 100], Y)	**894.40**

Table V presents the configurations for the Hopper-v4 environment. The best performer was surprisingly one of the most altered configuration. This configuration prioritized heavily forward reward by discounting the control costs and health penalties, encouraging risky behaviours. But this strategy fell apart when the health penalty was reintroduced, despite completely removing the control cost and increasing the accepted healthy range. The average rewards are however much closer from one configuration to the other, indicating that the environment could be less sensitive to extreme configurations.

TABLE VI: Subagent configurations for Humanoid-v4

(Forward reward weight, Control cost weight, Contact force cost weight, Healthy reward weight, Terminate when unhealthy)	Average reward
(0.5, 1.0, 1.5, 1.0, Y)	4768.19
(1.0, 0.5, 2.0, 1.0, Y)	3524.77
(1.0, 0.5, 1.0, 1.0, Y)	3457.94
(1.5, 0.5, 0.5, 0.5, Y)	3449.48
(0.5, 0.5, 0.5, 2.0, Y)	3265.30
(3.0, 0.4, 0.5, 0.5, N)	3251.51
(1.25, 0.1, 5e-7, 5.0, Y)	**3175.08**
(0.8, 0.1, 0.5, 1.0, Y)	3135.93
(5.0, 0.01, 0.5, 0.0, N)	2881.68

In the Humanoid-v4 environment, the best performer was the complete opposite of the Hopper-v4 top performer, as shown in Table VI. The best performing configuration was tweaked to have an increased control cost weight, and a discounted forward reward weight, promoting a safer and minimalist approach. The configuration with increased forward reward and discounted control cost performed poorly, with two of the bottom configurations disregarding the healthy reward completely.

D. Superagent Performance - Mixed Reward

TABLE VII: Superagents performance across environments

Number of Subagent(s)	Average Reward (HalfCheetah-v4)	Average Reward (Hopper-v4)	Average Reward (Humanoid-v4)
1	**-638.57**	1211.08	2947.44
2	-580.10	1218.86	3006.55
3	-370.44	**1070.40**	2924.49
4	-509.26	1239.58	**2847.21**
5	**-166.40**	1240.30	**3013.42**
6	-411.29	**1960.75**	2895.03
7	-269.72	1124.24	2924.15
8	-201.02	1317.25	3003.72

Table VII presents the superagent performance across the three environments according to the number of subagents used. In the Hopper-v4 environment, where the action space is smaller, the addition of subagents seems to directly contribute to performance enhancement. The performance consistently improves with the number of subagents up to five, achieving the highest average reward. This trend suggests that the lower dimensionality of the action space allows for effective integration and utilization of the diverse strategies provided by multiple subagents without excessively complicating the decision-making process. Beyond five subagents, the benefits stabilize, indicating a potential optimal number of subagents for balancing decision complexity and performance gain in this environment.

V. Ablation

We perform an ablation study by removing the attention module from the model. The observation state is an aggregation of the environment state and the subagent actions.

TABLE VIII: Superagents performance across environments - Same reward

Number of Subagent(s)	Average Reward (HalfCheetah-v4)	Average Reward (Hopper-v4)	Average Reward (Humanoid-v4)
1	-816.96	750.50	**3201.09**
2	-774.47	895.89	**3134.73**
3	**-845.92**	**1007.24**	3190.50
4	-743.34	916.88	3198.28
5	-831.56	959.84	3192.95
6	-685.47	785.57	3194.82
7	-822.90	907.81	3191.17
8	**-638.94**	**838.87**	3193.77

As shown in Table VIII, introducing subagent actions to the environment space without the attention module provides a reduction of the variance of average cumulative reward, especially in the Humanoid-v4 environment. The HalfCheetah-v4 environment presents a lower average cumulative reward with eight subagents, which could be due to the higher number of dimension in the observation space not being counterbalanced by the attention mechanism.

TABLE IX: Superagents performance across environments - Mixed reward

Number of Subagent(s)	Average Reward (HalfCheetah-v4)	Average Reward (Hopper-v4)	Average Reward (Humanoid-v4)
1	-403.16	1086.12	3046.32
2	-411.80	1181.87	**2909.05**
3	-398.84	**1228.52**	2999.23
4	-240.12	1186.45	2992.99
5	-389.94	1254.67	3061.87
6	-354.67	**992.95**	**3097.98**
7	**-736.08**	1158.10	3093.66
8	**-111.10**	1175.43	3072.68

The absence of the attention module is further illustrated in Table IX when using mixed rewards. The HalfCheetah-v4 environment shows a pattern of increasing the average reward as the number of subagents grows, up to five where the performance goes down. However, the best performer is surprisingly the one using eight subagents, indicating that more subagents could potentially enhance the performance regardless. In the Hopper-v4 environment, the lack of attention is the most evident, as the best performer with attention becomes the worst performer without it. This indicates a failure to capture the potential emerging behaviours brought up by the subagents. Finally the Humanoid-v4 environment remains consistent across the board, with no discernable pattern that can be linked to the number of subagents or the presence of the attention mechanism. The ablation study shows that the attention module has at best positive impact on the average reward for the Hopper-v4 and at worst no impact for the Humanoid-v4. The integration of the attention module also allows for further interpretability, as demonstrated in section VII-A.

VI. Training Time

The subagents need to be trained before suggesting relevant subactions. This means that the superagent can only be trained after the subagents have completed their own learning. In section IV, all agents were trained using four million timesteps. This means that a superagent with 2 subagents would have been trained effectively a total of 12 million timesteps. The following section will compare the performance of the centralized multi-agent model with attention with different numbers of subagents versus the performance of baseline PPO at different timesteps. The training time of each subagent and superagent remains 4M timesteps. Since the superagent can only be trained sequentially after the subagents, the total timesteps required to train can be approximated in two different ways:

- The subagents can be trained in parallel in separate environments and learn their respective policies independently. The total training time is then $t_{total} = t_{subagent} + t_{superagent}$, which in the Results section adds up to eight million timesteps.
- The subagents are trained in parallel, but the total training time of the model is a function of the number of subagents: $t_{total} = n_{subagent} * t_{subagent} + t_{superagent}$.

With t denoting the number of timesteps in training and $n_{subagent}$ the number of subagents.

Tables X, XI and XII compare the average reward of the superagent versus vanilla PPO depending of the number of subagents/timesteps and their respective environment.

TABLE X: Superagent performance versus PPO at different number of subagents/timesteps in the HalfCheetah-v4 environment

Number of Subagent(s)/Timesteps (millions)	Average Reward (Superagent)	Average Reward (Baseline PPO)
1 / 8	**-638.57**	-266.09
2 / 12	-580.10	-527.12
3 / 16	-370.44	-453.93
4 / 20	-509.26	-628.43
5 / 24	**-166.40**	**-150.54**
6 / 28	-411.29	-328.68
7 / 32	-269.72	**-628.64**
8 / 36	-201.02	-496.64

In Table X, the 24 million timesteps baseline PPO performs the best. The model experiences diminishing returns at a higher number of timesteps. At ≤ 6 subagents, the superagent gets outperformed by baseline PPO. But as the number of subagents increases, the superagent beats out baseline PPO.

TABLE XI: Superagent performance versus PPO at different number of subagents/timesteps in the Hopper-v4 environment

Number of Subagent(s)/Timesteps (millions)	Average Reward (Superagent)	Average Reward (Baseline PPO)
1 / 8	1211.08	1036.82
2 / 12	1218.86	802.06
3 / 16	**1070.40**	820.14
4 / 20	1239.58	909.84
5 / 24	1240.30	756.75
6 / 28	**1960.75**	**1185.06**
7 / 32	1124.24	**294.01**
8 / 36	1317.25	735.11

In Table XI, the superagent beats out baseline PPO. According to the second approach, the best performer in vanilla PPO has an equivalent training time to the best performer of the superagents, but a much lower reward. This result further indicates the adequacy of the centralized multi-agent model with attention for this environment.

TABLE XII: Superagent performance versus PPO at different number of subagents/timesteps in the Humanoid-v4 environment

Number of Subagent(s)/Timesteps (millions)	Average Reward (Superagent)	Average Reward (Baseline PPO)
1 / 8	2947.44	2818.45
2 / 12	3006.55	2224.58
3 / 16	2924.49	2058.88
4 / 20	**2847.21**	2178.31
5 / 24	**3013.42**	**1854.18**
6 / 28	2895.03	2108.89
7 / 32	2924.15	2283.85
8 / 36	3003.72	**2897.81**

In Table XII, the baseline PPO experiences heavy diminishing return as the number of timesteps increases, before increasing again. This could be due to an overfit in training in larger timesteps. The superagent remains stable with the number of subagents increasing, while not significantly improving upon the result of baseline 8M PPO. A possible avenue for improvement could be to further explore the optimal configurations for the subagents that cover a targeted range of useful behaviours.

TABLE XIII: Superagents performance across environments - Mixed reward at 4M timesteps

Number of Subagent(s)	Average Reward (HalfCheetah-v4)	Average Reward (Hopper-v4)	Average Reward (Humanoid-v4)
1	-434.83	1211.15	**3068.60**
2	-400.35	1220.97	3152.63
3	-489.26	1178.29	3116.24
4	-232.20	**1250.41**	3143.65
5	**-188.02**	1225.38	3143.34
6	-214.85	1200.25	3149.51
7	-315.33	1213.10	3182.25
8	-247.26	1198.95	3149.07
Baseline PPO	**-579.76**	**895.19**	**3190.66**

Table XIII presents the subagents performance across environments. The subagents and superagents were trained two million timesteps each, and the baseline PPO 4M timesteps. In HalfCheetah-v4, the top performer remains the superagent with five subagents. There is little variance in average reward for the Hopper-v4 environment, as all values except baseline are within ± 60. In both HalfCheetah-v4 and Hopper-v4, the superagents significantly outperform baseline. In Humanoid-v4, baseline is the best performer and outperforms all superagents. With seven subagents, the superagent comes close to outperforming benchmark PPO.

VII. Interpretability

A. Attention weights

Extracting the weights attributed to each component of the superagent state can help us interpret the model's decision making. We record and plot the attention weight for the three superagents with the best cumulative average reward.

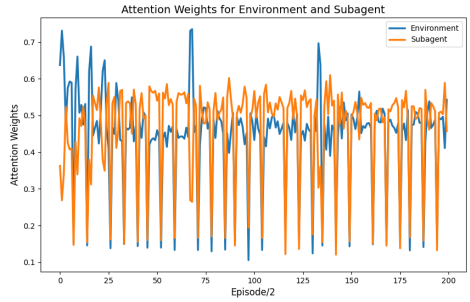


Fig. 1: Attention weights per episode/2 - HalfCheetah-v4

In Figure 1, The HalfCheetah-v4 strikes a balance of attention between the subactions and the environment state. A possible explanation for this distribution of attention could be that further training is needed to reach more stable attention weights.

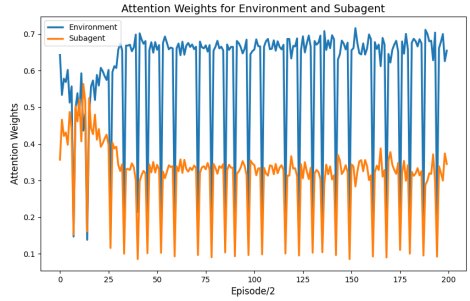


Fig. 2: Attention weights per episode/2 - Hopper-v4

The Hopper-v4 rapidly stabilises at a 70-30 split between the environment state and the subactions, as shown in Figure 2. Since this model outperformed the baseline considerably, a possible tool to tune the number of subagents and their configurations could be the distribution of attention weights. A quick convergence to a stable split of attention between subactions and environment state could indicate efficient prioritization from the superagent.

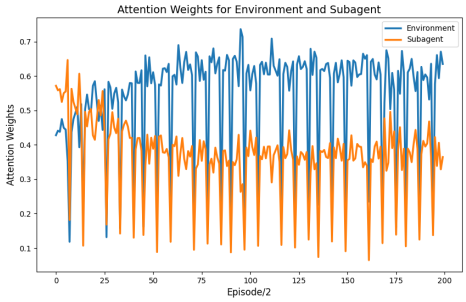


Fig. 3: Attention weights per episode/2 - Humanoid-v4

In Figure 3, the Humanoid-v4 is shown to follow a similar yet slower trend than the Hopper-v4. This could be due to needing a longer training time, or the difference in dimensions in the action space. The latter seems more likely, as the action space of the Hopper-v4 is only 2-dimensional and the action space of the Humanoid-v4 has 64 dimensions.

B. Cosine distance between actions of the superagent and subagents

In order to evaluate how close the action of the superagent are from the action of its subagents, we calculate the cosine distance [31] between the action vectors given by the subagents, and the one calculated by the superagent. We recorded this data over 10 epochs on a randomly seeded environment for the three superagents with the best cumulative average reward.

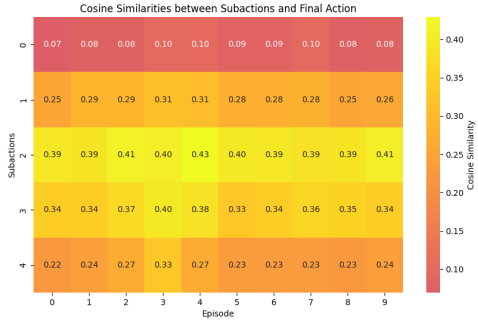


Fig. 4: Heatmap of cosine similarities between the subactions and the superagent actions - HalfCheetah-v4

In Figure 4, the superagent actions show very little cosine similarity with the actions taken by the subagent 0, despite this subagent earning the highest reward out of all the altered configurations. Instead, the superagent actions have a high cosine similarity with the actions of subagent 3, which used the base parameters. This could mean that the attention module failed to recognize behaviours that could potentially earn a higher reward. This could also mean that the behaviour proposed by the subagents were not sustainable long term strategies and instead were shortcuts to a local optimum.

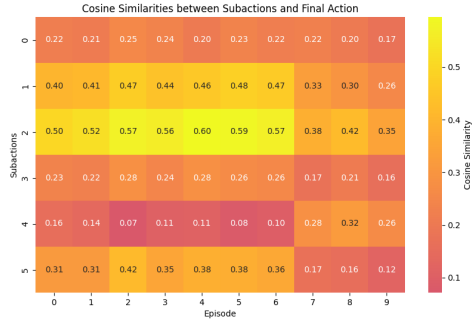


Fig. 5: Heatmap of cosine similarities between the subactions and the superagent actions - Hopper-v4

The Hopper-v4 environment presents the clearest trend, as shown in Figure 5: the three first subagents, which are the best performers, have a higher cosine similarity between the superagent actions and the subactions. This means that in this environment the attention managed to capture the relevancy of the actions advised by the subagent. The superagent's actions also demonstrate fluctuations across episodes in the cosine similarity with subactions, indicating that different strategies are prioritized depending on the environment state and the attention given to the subactions.

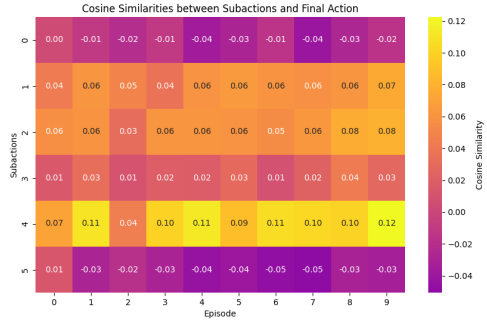


Fig. 6: Heatmap of cosine similarities between the subactions and the superagent actions - Humanoid-v4

In Figure 6, the Humanoid-v4 environment presents no similarity between the actions of the superagent and the subactions. The subagent 0, which performed outstandingly on the base environment, has a nearly 0 cosine similarity in its action with the superagent. The conclusions are similar to the HalfCheetah-v4 environment, and the model fails to

capture the value of the subactions despite the weighting of the aggregation function through the attention module.

VIII. Hardware

Prototyping and testing were done on a Nvidia 3090 FE. Experiments ran on a Nvidia A10 (24Gb PCIe, 30 CPU cores).

IX. Conclusion

This study presents a novel multi-agent architecture for Proximal Policy Optimization which harnesses attention to prioritize behaviours from subagents depending on the environment state. The model outperforms the baseline in the Hopper-v4 and HalfCheetah-v4 environments in mixed reward, and performs at baseline level in the Humanoid-v4 environment. Using mixed reward functions, the framework is versatile and applicable to real world problems.

The model performs best in a continous action space with few dimensions, as the benefits of augmenting the environment state with the suggested actions of the subagents fall off as the number of dimensions in the action space increases. This approach also provides additional interpretability tools by studying the attention weights and the cosine similarity between actions and subactions.

Future research could focus on implementing this model in real-life reinforcement learning problems. The attention module could also be extended into multi-headed attention, in order to have separate channels for each subagent. Another possible improvement could be to apply this method to another algorithm than PPO, as the principle behind the centralized multi-agent approach is model agnostic.

References

[1] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[2] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*. PMLR, 2015, pp. 1889–1897.

[3] Z. Wang, V. Bapst, N. Heess, V. Mnih, R. Munos, K. Kavukcuoglu, and N. De Freitas, "Sample efficient actor-critic with experience replay," *arXiv preprint arXiv:1611.01224*, 2016.

[4] L. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine *et al.*, "Model-based reinforcement learning for atari," *arXiv preprint arXiv:1903.00374*, 2019.

[5] M. S. Holubar and M. A. Wiering, "Continuous-action reinforcement learning for playing racing games: Comparing spg to ppo," *arXiv preprint arXiv:2001.05270*, 2020.

[6] S.-Y. Han and T. Liang, "Reinforcement-learning-based vibration control for a vehicle semi-active suspension system via the ppo approach," *Applied Sciences*, vol. 12, no. 6, p. 3078, 2022.

[7] L. Zhang, Y. Zhang, X. Zhao, and Z. Zou, "Image captioning via proximal policy optimization," *Image and Vision Computing*, vol. 108, p. 104126, 2021.

[8] T. Kobayashi, "Proximal policy optimization with relative pearson divergence," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8416–8421.

[9] Y. Gu, Y. Cheng, C. P. Chen, and X. Wang, "Proximal policy optimization with policy feedback," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 7, pp. 4600–4610, 2021.

[10] C. C.-Y. Hsu, C. Mendler-Dünner, and M. Hardt, "Revisiting design choices in proximal policy optimization," *arXiv preprint arXiv:2009.10897*, 2020.

[11] Q. Cai, Z. Yang, C. Jin, and Z. Wang, "Provably efficient exploration in policy optimization," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1283–1294.

[12] T. Yu, A. Kumar, R. Rafailov, A. Rajeswaran, S. Levine, and C. Finn, "Combo: Conservative offline model-based policy optimization," *Advances in neural information processing systems*, vol. 34, pp. 28 954–28 967, 2021.

[13] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," *Proceedings of the Tenth International Conference on Machine Learning*, 1993.

[14] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in Neural Information Processing Systems*, 2017.

[15] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse *et al.*, "Dota 2 with large scale deep reinforcement learning," *arXiv preprint arXiv:1912.06680*, 2019.

[16] X. Liang, X. Du, G. Wang, and Z. Han, "Deep reinforcement learning for traffic light control in vehicular networks," *arXiv preprint arXiv:1904.08117*, 2019.

[17] L. Matignon, G. Laurent, and N. Le Fort-Piat, "Coordinated multi-agent learning: The state of the art," *Artificial Intelligence Review*, vol. 37, no. 3, pp. 219–250, 2012.

[18] T. Yu, G. Qu, A. Singh, S. Levine, and C. Finn, "Meta-learning with latent embedding optimization in multi-agent systems," in *International Conference on Learning Representations*, 2020.

[19] G. Palmer, K. Tuyls, D. Bloembergen, and R. Savani, "Lenient multi-agent deep reinforcement learning," *arXiv preprint arXiv:1805.04566*, 2018.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[21] X. Guo, H. Zhang, H. Yang, L. Xu, and Z. Ye, "A single attention-based combination of cnn and rnn for relation classification," *IEEE Access*, vol. 7, pp. 12 467–12 475, 2019.

[22] R. Wu, A. Zhang, I. Ilyas, and T. Rekatsinas, "Attention-based learning for missing data imputation in holoclean," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 307–325, 2020.

[23] M. J. Er, Y. Zhang, N. Wang, and M. Pratama, "Attention pooling-based convolutional neural network for sentence modelling," *Information Sciences*, vol. 373, pp. 388–403, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025516306673

[24] L. Bramlage and A. Cortese, "Generalized attention-weighted reinforcement learning," *Neural Networks*, vol. 145, pp. 10–21, 2022.

[25] W. Wang, Y. Zhang, Y. Sui, Y. Wan, Z. Zhao, J. Wu, S. Y. Philip, and G. Xu, "Reinforcement-learning-guided source code summarization using hierarchical attention," *IEEE Transactions on software Engineering*, vol. 48, no. 1, pp. 102–119, 2020.

[26] U. Gunarathna, R. Borovica-Gajic, S. Karunasekara, and E. Tanin, "Solving dynamic graph problems with multi-attention deep reinforcement learning," *arXiv preprint arXiv:2201.04895*, 2022.

[27] C. Liu and G. Liu, "Jointppo: Diving deeper into the effectiveness of ppo in multi-agent reinforcement learning," *arXiv preprint arXiv:2404.11831*, 2024.

[28] L. Engstrom, A. Ilyas, S. Santurkar, D. Tsipras, F. Janoos, L. Rudolph, and A. Madry, "Implementation matters in deep policy gradients: A case study on ppo and trpo," *arXiv preprint arXiv:2005.12729*, 2020.

[29] S. Huang, R. F. J. Dossa, C. Ye, J. Braga, D. Chakraborty, K. Mehta, and J. G. Araújo, "Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms," *Journal of Machine Learning Research*, vol. 23, no. 274, pp. 1–18, 2022. [Online]. Available: http://jmlr.org/papers/v23/21-1342.html

[30] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[31] G. Qian, S. Sural, Y. Gu, and S. Pramanik, "Similarity between euclidean and cosine angle distance for nearest neighbor queries," in *Proceedings of the 2004 ACM symposium on Applied computing*, 2004, pp. 1232–1237.

Appendix C

Article 3 - Inverted Transformers Interpretability Beyond Attention Visualization [42]

Inverted Transformers Interpretability Beyond Attention Visualization

Hugo Cazaux*†, Ralph Rudd*, Hlynur Stefánsson*, Sverrir Ólafsson*, Eyjólfur Ingi Ásgeirsson*
*Department of Engineering, Reykjavik University, Menntavegur 1, 102 Reykjavik, Iceland
†Corresponding author: hugot20@ru.is, Menntavegur 1, 102 Reykjavik, Iceland

***Abstract*—Transformer-based forecasters have recently become state-of-the-art in time series predictions over linear forecasters. However, this promotion comes at the loss of the innate interpretability of linear forecasters. Real life examples of time series prediction often involve critical data, and interpretability is crucial for continuous improvement and accountability when predicting. This paper introduces a novel approach that extends existing transformer-specific interpretability methodologies to bridge this gap. This paper exploits relevance calculation that are employed in adjacent transformer-based architecture in image and natural language processing. This method is then applied to the inverted transformer architecture, one of the latest and best performing transformer-based forecaster. The method produces a relevance map linking the relationship drawn between features by the model. The relevance map gives feedback on the key variables that influence the model's decision the most, further enhancing the human understanding of the dataset. A key result of this study is a significant improvement in transformer based model explainability, without relying directly on visualizing the attention weights. Another key outcome is the potential adaptation of other transformer-specific tools to the inverted transformer.**

***Index Terms*—Time series analysis, Interpretability, Transformer, Regression**

I. Introduction

Transformers have revolutionized the natural language processing with their self-attention mechanism and layered feed-forward networks. The self-attention mechanism enables the model to weigh the importance of each input token in relation to others, allowing it to capture long-range dependencies, while the feed-forward networks refine these relationships across layers. However, their application to time series forecasting, especially with larger lookback windows, has lagged behind.

This lag was underscored by the surprising effectiveness of linear forecasters, which outperformed earlier adaptations of the transformer architecture for time series prediction [1]. With a low computational cost and a strong base of interpretability, the linear forecasters beat out the modified transformer architectures, especially for long-term predictions. As research progressed on Transformer-based models, linear forecasters are no longer the leading architecture, a step forward in performance at the cost of interpretability.

The inverted transformer (iTransformer) architecture is among the state-of-the-art models in time series analysis [2]. By inverting the typical duties of the attention mechanism and feed-forward networks of the standard transformer architecture, the architecture is better suited to forecast series with larger lookback windows. The iTransformer currently ranks first in the long-term forecasting task of the Time Series Analysis benchmarks [3]. As the iTransformer does not introduce any adaptation to the basic components, this architecture also benefits from the tools developed for the original Transformer architecture.

This paper leverages the similarity between the original Transformer architecture and the iTransformer to adapt and extend Transformer-specific interpretability methods. Specifically, we explore the application of Chefer's generic method for transformer interpretability [4]. This paper reformulates the original method for a continuous output, and adapts it to the inverted transformer architecture. The result is a continuous relevance map highlighting critical variates that are influential in the predictive power of the model, and greatly contribute to the tuning and accountability of the model.

The core research questions in this article are the following: can tools designed for Transformers be extended to the iTransformer architecture? Can the interpretability of iTransformer be improved using Transformer-specific techniques? What insights can be learned about the dataset and the model through relevance maps?

The paper is structured as follow: Section 2 includes a contextualization and a review of the existing literature. Section 3 introduces the mathematical definition of the method. Section 4 details the tokenization mechanism and draws the link between tokens and features. Section 5 presents the results of the method on a standard benchmark task. Section 6 presents an alternative representation of relevance for larger datasets. Section 7 is the conclusion to this study.

II. Background

The transformer architecture [5] has become a cornerstone of deep learning, particularly in natural language processing tasks. The self-attention mechanism allows the model to weight the importance of different tokens in a sequence relative to one another. This architecture is the foundation behind most of the mainstream models, such as ChatGPT [6], Claude [7], Mistral [8], and Llama [9]. The surge in research has provided fast improvement in parallelization [10] and diverse optimizations [11].

Various modification paradigms have been proposed to improve the accuracy of Transformer-based forecasters. Auto-

former [12] and Informer [13] propose to replace the attention component respectively with an autocorrelation and sparse attention mechanisms. Crossformer [14] focuses on modeling the cross-time and cross-dimension dependency using a two-stage attention and modified hierarchical encoder-decoder architecture. Finally, PatchTST [15] and NSTransformer [16] focused on the processing of time series using patching and stationarization respectively.

The inverted transformer introduces no modification to the original Transformer components [5], which opens up the possibility of adapting any transformer specific technique. By inverting the duties of the attention mechanism and the feed-forward network, this architecture aims to reduce performance degradation and computation explosion in larger lookback windows. This recent model has been successfully used to predict the useful life of Lithium-Ion batteries [17], earthquake detection [18] and predict sea surface temperature [19].

Interpretability methods for Transformers are sparse in the literature. Exploiting the raw attention weights to draw attention maps has earned a skeptical reputation for interpretability [20] [21]. The ConceptTransformer [22] proposed to modify the architecture for better explainability. Vision Transformer is the most prolific source of interpretability attempts, with neural tree decoder [23] and interpretability-aware training objectives [24].

The original Chefer et al. method assigns local relevance, circumventing the suboptimal nature of considering the mean attention heads due to the nonequivalent relevance of attention heads in each layer [25]. The method presented in this study does not rely on Layer-wise Relevance Propagation [26], which was limited to self-attention based models. Instead, we base our research on the generic attention-model explainability framework [4] for encoder-decoder transformers. While this method is applied to build image segmentation masks in the original paper, this study opts to adapt the idea to time-series prediction.

III. Preliminaries

We start by introducing the iTransformer architecture and Chefer's method for explainability in their original form as preliminaries.

A. iTransformer

Given historical observations $X = \{x_1, x_2, \ldots, x_T\} \in \mathbb{R}^{T\times N}$ with T time steps and N variates, the goal is to predict the future S time steps $Y = \{x_{T+1}, x_{T+2}, \ldots, x_{T+S}\} \in \mathbb{R}^{S\times N}$. In the iTransformer, each time series of a variate is embedded into variate tokens, which are then utilized by the attention mechanism to capture multivariate correlations. This mechanism is further detailed in Section V. The feed-forward network is applied to each variate token to learn nonlinear representations, and the final output is generated by projecting these representations back to the time series domain.

The process can be formulated as follows:

$$h_0^n = \text{Embedding}(X_{:,n}) \tag{1}$$

$$H^{l+1} = \text{TrmBlock}(H^l), \quad l = 0, \ldots, L-1 \tag{2}$$

$$\hat{Y}_{:,n} = \text{Projection}(h_L^n) \tag{3}$$

where $H = \{h_1, h_2, \ldots, h_N\} \in \mathbb{R}^{N\times D}$ contains N embedded tokens of dimension D. The functions Embedding : $\mathbb{R}^T \to \mathbb{R}^D$ and Projection : $\mathbb{R}^D \to \mathbb{R}^S$ are implemented by multi-layer perceptrons (MLP). The self-attention and feed-forward network operations in each Transformer block (TrmBlock) enable the model to learn complex dependencies across variates and time steps.

B. Chefer's Method for Attention Models Explainability

For two modalities (e.g., text with t tokens and image with i tokens), the initial relevance maps are set as:

$$R_{tt} = I_t, \quad R_{ii} = I_i, \tag{4}$$

$$R_{ti} = 0, \quad R_{it} = 0, \tag{5}$$

where I denotes the identity matrix. and then modified using gradients to yield a class-specific, head-averaged map:

$$\bar{A} = E_h\Big((\nabla_A \odot A)^+\Big), \tag{6}$$

where $\odot$ is the element-wise product, $E_h(\cdot)$ averages over heads, and $(\cdot)^+$ zeros negative contributions. For self-attention layers, the relevance update is:

$$R_{ss} \leftarrow R_{ss} + \bar{A}\, R_{ss}. \tag{7}$$

For bi-modal (cross-modal) interactions, after normalizing the aggregated self-attention relevance, the update becomes:

$$R_{sq} \leftarrow R_{sq} + \bar{R}_{ss}^{\top}\, \bar{A}\, \bar{R}_{qq}, \tag{8}$$

with $\bar{R}_{xx}$ denoting the normalized self-attention relevance for modality x.

IV. Methodology

In this section, we adapt the explainability method proposed by Chefer et al. for the iTransformer architecture, focusing on enhancing interpretability in time series regression tasks. Our approach involves initializing and updating relevancy maps to trace back the contribution of each variate token to the final output.

A. Relevancy Initialization

We initialize the relevancy maps as follows:

$$R_{vd} = I_{v\times d} \tag{9}$$

$$R_{vt} = 0_{v\times t} \tag{10}$$

$$R_{vv} = Concat(R_{vd}, R_{vt}) \tag{11}$$

Here, R_{dv} represents the self-attention relevancy map for variate tokens, where d is the number of variate tokens. R_{vt} represents the interaction between all tokens and time-related tokens, where t is the number of time tokens and $v = d+t$. The

identity matrix $I_{v\times d}$ ensures that each variate token initially has a relevance score focused on itself, while the zero matrix $0_{v\times d}$ indicates no initial interaction between variate and time tokens. R_{vv} is the concatenation of R_{vd} and R_{vt} alongside the dimension 0 (v)

B. Self-Attention Relevancy Update

In the iTransformer, self-attention is applied to variate tokens. The relevancy maps are updated using the attention weights:

$$R_{vv} \leftarrow R_{vv} + \overline{A}_{vv} \odot R_{vv} \tag{12}$$

where $\overline{A}_{vv}$ represents the averaged attention weights for variate tokens, computed as:

$$\overline{A}_{vv} = \frac{1}{H} \sum_{h=1}^{H} \text{ReLU}\left(\nabla A_{vv}^{h} \odot A_{vv}^{h}\right) \tag{13}$$

Here, H denotes the number of attention heads, ∇A_{vv}^{h} represents the gradients of the attention weights with respect to the output, and $\odot$ denotes element-wise multiplication. ReLU is used to ensure that only positive contributions are considered, aligning with the focus on interpretability..

C. Feed-Forward Network Relevancy Update

The feed-forward network relevancy update is applied independently to each variate token:

$$R_{\text{ff}} = Expand(\text{ReLU}(\nabla y \odot y)) \tag{14}$$

where y represents the outputs of the feed-forward network, and ∇y denotes the gradients of these outputs with respect to the loss.

$$R_{vv} \leftarrow R_{vv} + R_{\text{ff}} \tag{15}$$

Equation 15 refines the relevancy scores by applying learned transformations, emphasizing the most influential tokens.

D. Visualization and Analysis

The relevancy maps for each variate token can be visualized similarly to attention maps. For interpretability, we focus on how each variate token contributes to the final regression output. All relevance scores are normalized between $[0, 1]$ for readability. Each cell at position (i, j) shows the relevance of token i (on the y-axis) to token j (on the x-axis).

E. Algorithm Implementation

The following pseudo-code outlines the implementation of the adapted Chefer method for the iTransformer:

The relevancy mapping allows for tracing back and visualizing the contribution of each variate token to the final regression output, enhancing the interpretability of the iTransformer in time series regression tasks.

Algorithm 1 Relevancy Mapping for iTransformer

Input: Number of variate tokens v, number of time tokens t, weights W_1, W_2
Output: Relevancy maps R_{output}
Initialize $R_{vd} \leftarrow I_{v\times v}$ ▷ Self-attention for variate tokens
Initialize $R_{vt} \leftarrow 0_{v\times t}$ ▷ Interaction of variate tokens with time-related tokens, if applicable
Initialize $R_{vv} \leftarrow Concat(R_{vd}, R_{vt})$ ▷ Concatenation alongside dimension 0 (v)
for each layer in iTransformer **do**
    $A_{vv} \leftarrow$ layer.variate_attention_map()
    $\overline{A_{vv}} \leftarrow \frac{1}{H}\sum_{h=1}^{H} \text{ReLU}\left(\nabla A_{vv}^{h} \odot A_{vv}^{h}\right)$ ▷ Averaged across heads
    $R_{vv} \leftarrow R_{vv} + \overline{A_{vv}} \cdot R_{vv}$
    $R_{vv} \leftarrow R_{vv} + Expand(\text{ReLU}(\nabla y \odot y))$ ▷ Feed-Forward update
Return R_{output}

V. Token Embedding

In this section, we provide a detailed analysis of the embedding process used by the iTransformer architecture. The embedding process is crucial for transforming the input time series data into a set of tokens that the model can effectively process to capture important multivariate correlations.

A. Mathematical Representation of the Embedding Process

The iTransformer architecture inverts the traditional roles of the attention mechanism and the feed-forward network found in conventional transformers. Instead of using tokens to represent time steps, the iTransformer generates tokens that correspond to the variates (or features) of the time series.

The embedding process involves the following steps:

- **Concatenation of Features and Temporal Information:** The variates and temporal features are first concatenated along the feature dimension:

$$\mathbf{Z} = [\mathbf{X}, \mathbf{M}] \in \mathbb{R}^{B\times T\times (V+F)} \tag{16}$$

  where $\mathbf{X} \in \mathbb{R}^{B\times T\times V}$ is the input time series data, $\mathbf{M} \in \mathbb{R}^{B\times T\times F}$ represent the temporal features, B is the batch size, T is the sequence length, V is the number of variates (features) and F is the number of temporal features (e.g., time of day, day of the week, seasonality). $\mathbf{Z} \in \mathbb{R}^{B\times T\times (V+F)}$ is the concatenated representation where each time step t in the sequence now has $V + F$ dimensions, accounting for both the variates and temporal features.
- **Linear Transformation to Token Space:** A linear transformation is applied to map each combined feature vector at time step t to a d-dimensional token space:

$$\mathbf{E}_t = \mathbf{W}_e \mathbf{Z}_t + \mathbf{b}_e, \quad \mathbf{E} \in \mathbb{R}^{B\times T\times d} \tag{17}$$

Fig. 1: Relevance maps for training and testing of CT with the baseline lookback (96) and prediction length (96) for the ETT2 dataset. This experiment serves as the benchmark for the different other experiments by using the shortest lookback and prediction.

where $\mathbf{W}_e \in \mathbb{R}^{(V+F)\times d}$ is the weight matrix, $\mathbf{b}_e \in \mathbb{R}^d$ is the bias vector, and $\mathbf{E}_t \in \mathbb{R}^{B\times d}$ is the embedded token for time step t.

- **Generation of Variate Tokens:** The attention mechanism processes the sequence of tokens $\mathbf{E}$ to capture the relationships between different variates across the entire time series. Each variate token $\mathbf{e}_v$ can be represented as:

$$\mathbf{e}_t = \sum_{v=1}^{V+F} \alpha_{t,v}\mathbf{E}_v, \quad t = 1, \ldots, (T) \tag{18}$$

where $\alpha_{v,t}$ are the attention weights that determine the contribution of each variate v to the variate token $\mathbf{e}_t$. The resulting tokens $\mathbf{e}_t$ encapsulate the interactions between different variates, as well as their temporal contexts.

B. Correspondence Between Tokens and Variates

The datasets used are ETT2 [13], Weather [1] and Electricity [27]. These three datasets are standard benchmarks in time-series prediction, and used to cover a wide range of seasonality, number of features and scale of data. We start by focusing on the ETTh2 dataset, with detailed layer-wise representations. We then propose alternative use cases of the relevance mechanism with the Weather and Electricity datasets. The embedding process results in a set of tokens, each corresponding to a specific variate or temporal feature. The iTransformer and its modified counterpart generate a total of V+F=11 tokens in the experiments conducted on the ETT2 dataset.

Token	Feature	Description	Units
0	HUFL	High UseFul Load	kW
1	HULL	High UseLess Load	kW
2	MUFL	Middle UseFul Load	kW
3	MULL	Middle UseLess Load	kW
4	LUFL	Low UseFull Load	kW
5	LULL	Low UseLess Load	kW
6	OT	Oil temperature of the transformer	°C
7	hourDay	hour of the day	-
8	dayWeek	day of the week	-
9	dayMonth	day of the month	-
10	dayYear	day of the year	-

TABLE I: Tokenized features of the ETT2 dataset.

Table I presents the correspondence between tokens and features in the experiments. The temporal features are transformed as value between $[-0.5, 0.5]$ before being embedded into tokens 7 to 10. The direct embedding of a variate as a token allows for a better understanding of the relationship between features. The goal is to predict future values of token 6 OT, and as such the embedded past values are expected to have a high relevancy.

VI. Results

The experimental setting uses a modified version of the official iTransformer implementation [28]. Prototyping and testing were done using a Nvidia 3090 FE. The code used is available at Github [29]. In the following section, the

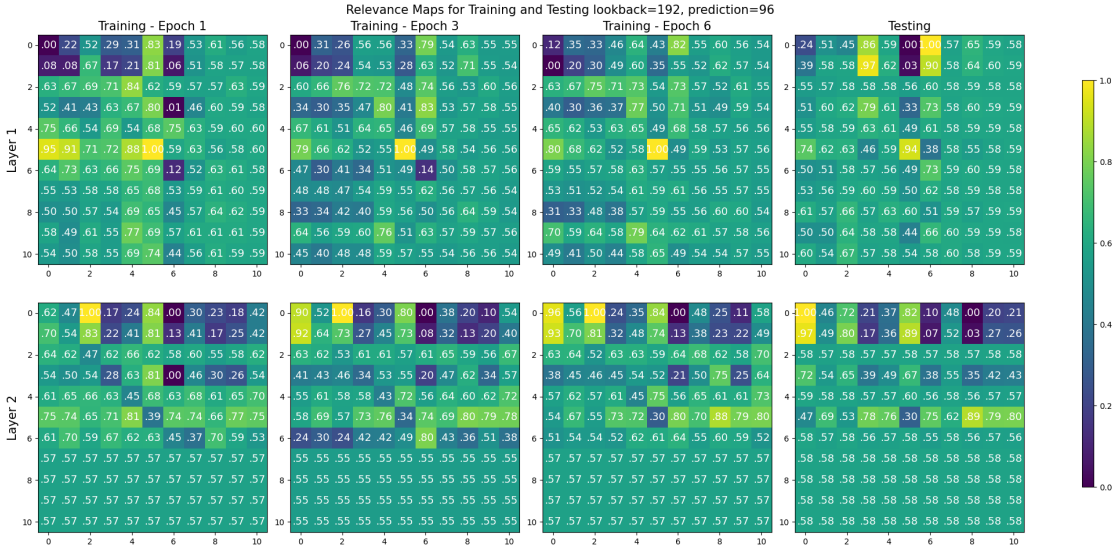


Fig. 2: Relevance maps for training and testing with large lookback window (192) and short predictive horizon (96) for the ETT2 dataset. The model has more data available to forecast the same number of tokens, making it an ideal use case for the iTransformer architecture.

notation CT_<lookbackWindow>_<predictionLength> will be used to designate the experiments, and CT stands for CheferTransformer. All experiments use two encoder layers to represent the multivariate series.

A. Performance Metrics

We compare the performance of the modified iTransformer (CheferTransformer) to the original implementation. The lookback window is $T = 96$ for the performance comparison. The loss used throughout this study are the Mean Squared Error and the Mean Absolute Error.

Table II presents the Mean Squared Error (MSE), Mean Absolute Error (MAE), and the training time of the modified iTransformer vs the vanilla iTransformer. The experiment was ran four times for each model at different prediction length: 96, 192, 336 and 720. The goal of this experiment is to observe if there is any loss in performance or training time when interpreting the results. As our adaptation of Chefer's method for interpretability does not interfere with the original architecture, the MSE and MAE are the same for every experiment. The training time is however slightly faster for the vanilla iTransformer, which is explained by the storing and rearranging of the relevancy maps and gradients in the modified architecture. The following sections will focus on the CT at varying lookback windows and prediction lengths.

TABLE II: Performance of Modified vs. Original iTransformer, P is Prediction Length

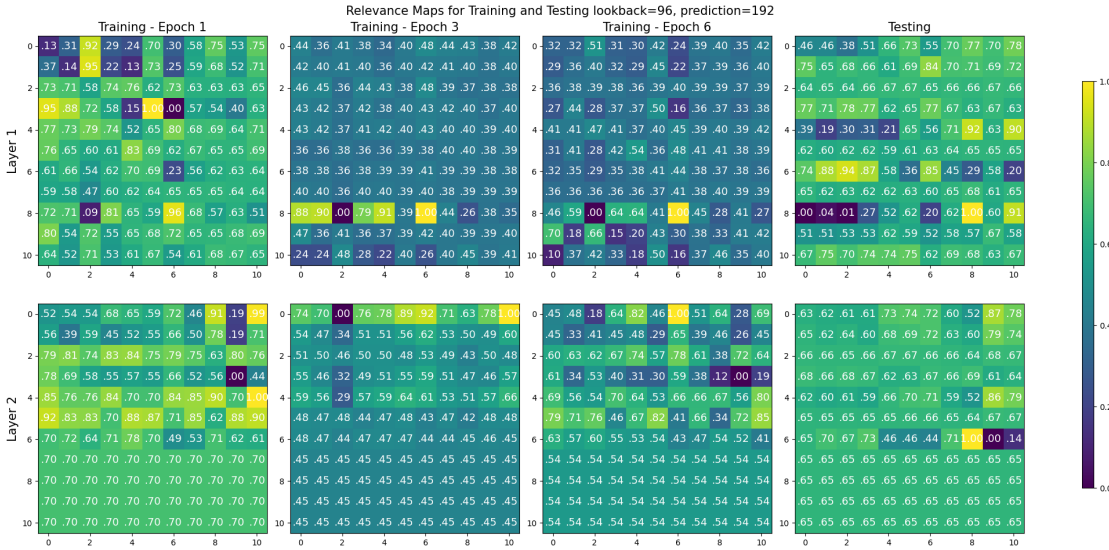
P	MSE	MAE	Training Time Chefer/Vanilla
96	0.300	0.349	937.8/923.8
192	0.381	0.399	563.8/556.9
336	0.423	0.432	745.4/742.1
720	0.426	0.445	464.2/463.8

B. Relevance Maps

Figure 1 presents the relevance maps obtained in experiment CT_96_96 during training and testing. At epoch 1, Layer 1 shows a scattered focus, with the token 6 receiving significant attention, indicating that early in training the shallowest layer of the model tend to focus on individual tokens. This is an encouraging sign indicating that the training is effective, as the model recognizes after one epoch that the past value of the predicted variate (token 6) is pivotal to make a prediction. Layer 2 proposes a more diffused attention, suggesting that the deeper layer is still early in the process of learning how to efficiently distribute attention.

By epoch 5, the distribution of relevance in layer 1 has shifted. The relevancy of tokens 6 and 8 attending to other tokens has plummeted. Layer 2 maintains the relevancy of tokens 4, 5 and 6. In layer 2, the smoothed out attention of tokens 7 to 10 gives crucial insights: token 7 to 10 are the embedded temporal features of the dataset, and the model recognizes and integrates those as a consistent part of its

Fig. 3: Relevance maps for training and testing with short lookback (96) and large prediction length (192) for the ETT2 dataset. The model has to forecast more data despite a shorter lookback window, which is a generally harder task.



forecasting process with different relevance when attended to by tokens 1 to 6. Layer 2 posits that the temporal features are equally relevant when attending to other tokens. Layer 2 also highlights that all the temporal tokens (7 to 10) have the same relevancy when attending to each other.

By epoch 10, the variation in relevance in layer 1 are more uniform, with fewer extremely low values. A good example of the non-commutativity of attention is token 6, which when attending to other tokens has a low relevancy, but is extremely relevant when attended to. Layer 2 presents relevance scores that are much lower, but specific relationships are highly emphasized, for instance token 0 attending to token 8.

The difference in relevance distribution throughout the epochs indicate that the two layers are attempting to capture different aspects of the input sequence. According to the hierarchy of the model, it is likely that layer 1 focuses on immediate dependency between features, while layer 2 is attempting to capture higher level trends. This would also explain the fewer variations in distribution of relevance in layer 2 for tokens 7 to 10.

The last column represents the relevance map of the fully trained model after 10 epochs, when presented with unseen data. A first observation is that the pattern of relevancy in layer 1 is quite similar to training in both layers. Token 6 for instance maintains a high relevancy when attended to in layer 1. Layer 2 adopts a more balanced strategy, indicating that the model is relying on more different tokens than training when presented with unseen data.

The balanced strategy is an encouraging sign that the model is not overfitting to the training set. Clearly, certain tokens have been identified as important for prediction, but the overall spread out attention indicates that the attention mechanism is robust towards new data. Layer 1 and 2 agree on the most relevant token being token 6, especially when attended to. A possible interpretation of the forecasting strategy could be that the model bases its prediction strongly on the past value of `OT` and utilizes the context clues given by the other tokens to fine-tune the predicted value.

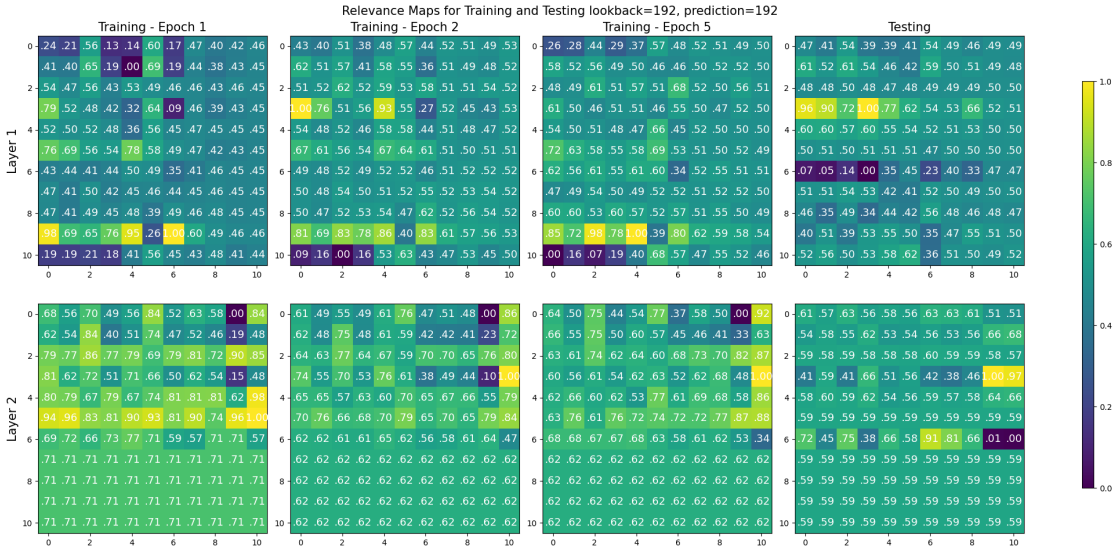
C. Increasing Lookback Window

Figure 2 presents the relevance maps obtained in experiment CT_192_96 (lookback window of 192 data points, predicting 96 ahead) during training and testing. One of the key improvements of the iTransformer model compared to original forecasting transformers is a more efficient use of larger lookback windows. CT_192_96 presents an extremely distributed relevance at epoch 1, with the high load tokens (0 and 1) presenting low relevance when attending to other tokens in both layers.

At epoch 3, token 5 appears to be the most relevant in layer 1. Interestingly, the two layers disagree on the relevance of several tokens, for instance the relevance of token 0 to itself and the relevance of token 1 to token 0. Layer 1 deems this relationship almost irrelevant whereas layer 2 advocates for a near-1 relevance.

At epoch 6 the model reaches early stopping and finishes training. Token 2 (middle useful load) has significant relevance when attending to the other tokens, and token 6 (`OT`, target)

Fig. 4: Relevance maps for training and testing with large lookback (192) and large prediction length (192) for the ETT2 dataset. The model has to forecast far in the future, but also disposes of an equally large lookback.



has the highest relevance when attended to in layer 1. In layer 2, Token 0 and 1, the high load features, are quite relevant to all other tokens but not nearly as much to the temporal tokens. Specifically, only the time of the year and the hour of the day appears to have relevancy. A possible explanation could be that layer 2 captures seasonality and the day of the year is a much better metric than the month to know which season it is currently due to lower/higher temperature and more/less sun exposure. Token 3, the middle useless load, is only slightly relevant to the day of the month, but highly relevant to the day of the week and year, revealing potential inefficiencies at specific periods of the year and the week.

During testing, layer 1 highly prioritizes token 6 when attended to. Layer 2 maintains a very similar distribution to training. These results indicate that the model generalizes well, and is capable of re-applying the forecasting strategies learnt in training.

D. Increasing Prediction Length

Figure 3 presents the relevance maps obtained in experiment CT_96_192 during training and testing. Unlike CT_96_96 or CT_96_192, in this experiment the model kept a relatively even distribution of relevance between the different tokens related to dataset features. This can be seen in layer 1 at epoch 3. The temporal features had much more emphasis during the training compared to shorter prediction length, with the relevance values changing dramatically between epoch 3 to 6 in layer 1. Token 8, the day of the week, appears to have had the widest range of values when attending the high load tokens, starting out really high in epoch 1 to 3 and plummeting in epoch 6 and testing. In layer 2, `OT` seems to be the token with the most developed relevancy. For instance, layer 2 deems completely irrelevant the day of the month and the year, but the day of the week essential.

E. Increasing Both Prediction Length and Lookback Window

Figure 4 presents the relevance maps obtained in experiment CT_192_192 during training and testing. This experiment presents a mix of strategies between CT_96_192 and CT_192_96. The temporal tokens are highly emphasized during training, especially day of the month and day of the year in layer 1. Interestingly, during testing token 6 is not very relevant when attending the high and middle load tokens in layer 1, but highly relevant in layer 2. This could indicate that the previous values are more suited to infer high level trends, according to the layer hierarchy.

VII. Other Datasets and Combined View

In dataset with a low number of features, it is convenient to visualize the relationship between each token and in each layer. However, when the number of features grows, this representation can become harder to read and less useful. We present combined relevance of the features by summing over by layers and by pairwise relevance and applying a normalization of the values between 0 and 1. This alternative visualization provides a more direct overview of which features are the most relevant in a prediction, and is more suited for datasets with a large number of features. To demonstrate this method, we

experiment on two other staple datasets from the Time-series library, Weather and Electricity.

A. Weather

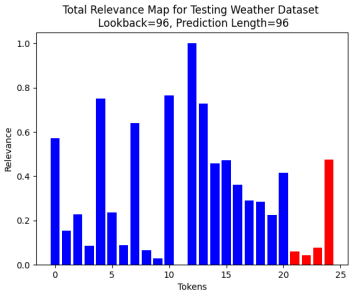
The Weather dataset contains 21 meteorological indicators recorded for the whole year of 2020. Table III presents the features of the dataset. The goal is to predict the CO2 concentration in ambient air, represented by token 20 `OT`.

TABLE III: Summary of Weather Dataset Features

Token	Feature Name	Units	Description
0	`p`	mbar	Air pressure
1	`T`	°C	Air temperature
2	`Tpot`	K	Potential temperature
3	`Tdew`	°C	Dew point temperature
4	`rh`	%	Relative humidity
5	`VPmax`	mbar	Maximum vapor pressure
6	`VPact`	mbar	Actual vapor pressure
7	`VPdef`	mbar	Vapor pressure deficit
8	`sh`	g/kg	Specific humidity
9	`H2OC`	mmol/mol	Water vapor concentration
10	`rho`	g/m^3	Air density
11	`wv`	m/s	Wind velocity
12	`max.`	m/s	Maximum wind velocity
13	`wd`	degrees	Wind direction
14	`rain`	mm	Precipitation
15	`raining`	s	Duration of precipitation
16	`SWDR`	W/m^2	Shortwave downward radiation
17	`PAR`	μmol/m^2/s	Active radiation
18	`max. PAR`	μmol/m^2/s	Maximum active radiation
19	`Tlog (degC)`	°C	Internal logger temperature
20	`OT`	ppm	CO2 concentration
21-24	`date`	–	Date

By combining Figure 6 and Table III, we can determine that the most relevant feature is the maximum wind velocity. In contrast, the current wind velocity appears to have no relevance. The 24th token is a highly relevant and corresponds to the day of the year, possibly indicating that the season could play a part in accurately predicting the CO2 concentration. Similarly, the maximum wind velocity, air density, vapor pressure deficit and air pressure are the next most relevant features.

Fig. 6: Total relevance map for Weather dataset. Dataset features are in blue, while temporal features are in red.



B. Electricity

The Electricity dataset contains the daily electricity consumption of 321 Portuguese clients in kW. The goal is to predict the consumption of client 321 based on the consumption of the other 320 clients. Figure 5 presents the relevance of each token in predicting client 321. The temporal features in green highlight that the time of the day is the most relevant temporal features, as it can be expected in electrical consumption. Perhaps surprisingly, the past values of the client do not constitute the most relevant token. Token 112 is the most relevant in predicting client 321, and upon plotting the target variable and client 112 in Figure 7, we observe a similar seasonality in the data.

Fig. 7: Client 112 consumption vs Target consumption.

Another application of the summed relevance is a similar process to Principal Component Analysis (PCA). If we wish to reduce the computational overhead of the model, one of the easiest way is to cut down the number of features. We want to keep the most relevant features, and can use Figure 8 to draw a threshold at the most relevant features. In this example the threshold is arbitrarly set at 0.1 relevance, and cuts about half of the tokens.

Fig. 8: Total sorted relevance map for Electricity dataset. Temporal features are in red, dataset features in blue.

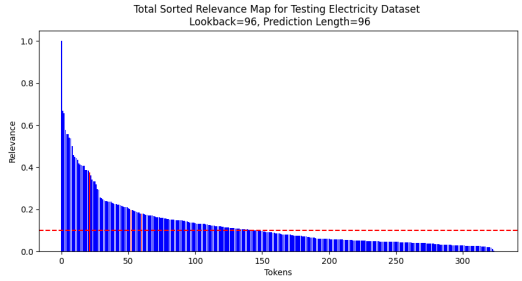
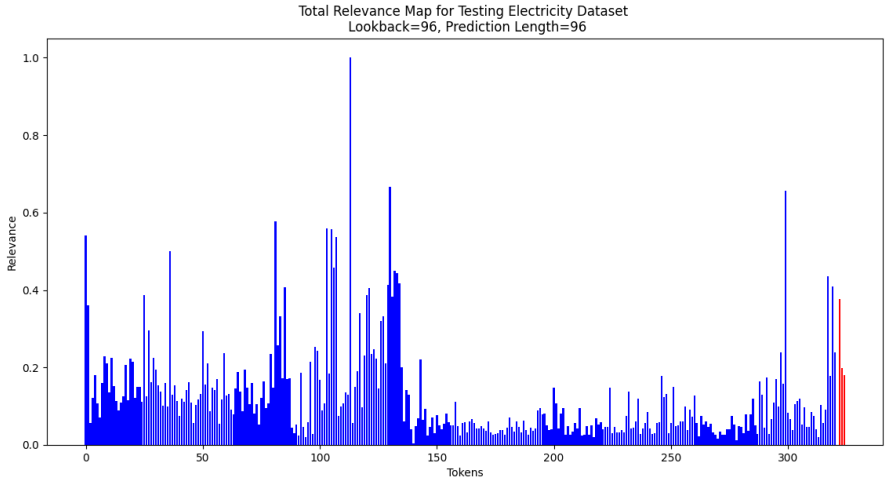


Fig. 5: Total relevance map for Electricity dataset. Dataset features are in blue, while temporal features are in red.



VIII. Future Developments

Relevance maps can be adapted to any Transformer-based architecture, and the iTransformer places itself in the literature as a basis to be improved upon. In timeseries prediction, several modifications to the attention mechanisms have proven to be efficient to improve the performance of the Transformer model [14], [16]. As the ecosystem around the iTransformer grows, there is an opportunity for interpretability by adapting the proposed method.

The datasets used in this article are benchmarks designed to challenge the predictive power of machine learning models at different scales. In a real life application, feature importance and relevance maps can contribute to the expert knowledge of the field, and inversely a deeper understanding of each features can help improve the performance of the model.

IX. Conclusions

The results of our initial experiments on ETT2 provide clear insights into how the model attends to different features over time, with significant focus on key tokens such as the embedded past values of `OT` for the ETT2 dataset, indicating that past values of specific variates play a critical role in prediction. The ability of Layer 1 to capture short-term dependencies and Layer 2 to focus on higher-level trends was consistent across multiple configurations, and the differences in relevance between training and testing phases suggest that the model generalizes well. Notably, as the prediction horizon increases, the relevance maps highlight the model's emphasis on temporal features in Layer 1.

In datasets with a larger amount of features, such as Weather or Electricity, summing over layers and pairwise relevance allows to distinguish the key features to the dataset. This can be crucial in real life applications to determine which features are worth measuring with greater accuracy. Through the relevance values we can also determine the least useful features and develop an analogous method to PCA to cut down the number of features in a dataset and subsequently reduce the overhead for computational complexity in large datasets.

This study extends and adapts generic methods for transformer interpretability to the inverted transformer architecture. With no loss in performance and only a negligible amount of training time, we extract relevance values that indicate the most important variates, which can then be extrapolated to feature importance in the dataset. By highlighting which features are key, we can also facilitate hyperparameter tuning and detect less useful features that might introduce noise. This is a significant result to maintain the interpretability offered by the then state-of-the-art linear models, and is crucial in real-life applications where intepretability meets accountability.

References

[1] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, pp. 11121–11128, 2023.

[2] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," *arXiv preprint arXiv:2310.06625*, 2023.

[3] Y. Wang, H. Wu, J. Dong, Y. Liu, M. Long, and J. Wang, "Deep time series models: A comprehensive survey and benchmark," 2024.

[4] H. Chefer, S. Gur, and L. Wolf, "Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 397–406, 2021.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[6] OpenAI, "Openai." https://openai.com/, 2024.

[7] Anthropic, "Claude." https://www.anthropic.com/claude, 2024.

[8] Mistral, "Mistral 7b." https://mistral.ai/, 2024.

[9] Meta, "Llama 3.1." https://llama.meta.com/, 2024.

[10] Z. Zeng, C. Liu, Z. Tang, K. Li, and K. Li, "Acctfm: An effective intra-layer model parallelization strategy for training large-scale transformer-based models," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 12, pp. 4326–4338, 2022.

[11] V. A. Korthikanti, J. Casper, S. Lym, L. McAfee, M. Andersch, M. Shoeybi, and B. Catanzaro, "Reducing activation recomputation in large transformer models," *Proceedings of Machine Learning and Systems*, vol. 5, pp. 341–353, 2023.

[12] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in neural information processing systems*, vol. 34, pp. 22419–22430, 2021.

[13] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, pp. 11106–11115, 2021.

[14] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *The eleventh international conference on learning representations*, 2023.

[15] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730*, 2022.

[16] Y. Liu, H. Wu, J. Wang, and M. Long, "Non-stationary transformers: Exploring the stationarity in time series forecasting," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9881–9893, 2022.

[17] A. Jha, O. Dorkar, A. Biswas, and A. Emadi, "itransformer network based approach for accurate remaining useful life prediction in lithium-ion batteries," in *2024 IEEE Transportation Electrification Conference and Expo (ITEC)*, pp. 1–8, IEEE, 2024.

[18] L. Wang, Z. Li, Y. Chen, J. Wang, and J. Fu, "Maxent seismosense model: Ionospheric earthquake anomaly detection based on the maximum entropy principle," *Atmosphere*, vol. 15, no. 4, p. 419, 2024.

[19] W. Jia, S. Guan, and Y. Xue, "Tl-itransformer: Revolutionizing sea surface temperature prediction through itransformer and transfer learning," *Earth Science Informatics*, pp. 1–11, 2024.

[20] S. Jain and B. C. Wallace, "Attention is not explanation," *arXiv preprint arXiv:1902.10186*, 2019.

[21] S. Serrano and N. A. Smith, "Is attention interpretable?," *arXiv preprint arXiv:1906.03731*, 2019.

[22] M. Rigotti, C. Miksovic, I. Giurgiu, T. Gschwind, and P. Scotton, "Attention-based interpretability with concept transformers," in *International conference on learning representations*, 2021.

[23] S. Kim, J. Nam, and B. C. Ko, "Vit-net: Interpretable vision transformers with neural tree decoder," in *International conference on machine learning*, pp. 11162–11172, PMLR, 2022.

[24] Y. Qiang, C. Li, P. Khanduri, and D. Zhu, "Interpretability-aware vision transformer," *arXiv preprint arXiv:2309.08035*, 2023.

[25] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov, "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned," *arXiv preprint arXiv:1905.09418*, 2019.

[26] A. Binder, G. Montavon, S. Lapuschkin, K.-R. Müller, and W. Samek, "Layer-wise relevance propagation for neural networks with local renormalization layers," in *Artificial Neural Networks and Machine Learning–ICANN 2016: 25th International Conference on Artificial Neural Networks, Barcelona, Spain, September 6-9, 2016, Proceedings, Part II 25*, pp. 63–71, Springer, 2016.

[27] A. Trindade, "ElectricityLoadDiagrams20112014." UCI Machine Learning Repository, 2015. DOI: https://doi.org/10.24432/C58C86.

[28] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "stable-baseline-3." https://github.com/thuml/Time-Series-Library/blob/main/models/iTransformer.py, 2024.

[29] Cazaux, "chefer-transformer." https://github.com/hugoiscracked/chefertransformer, 2024.

Appendix D

Article 4 - Non-Stationary iTransformer With Time2Vec Embeddings [41]

Non-Stationary Inverted Transformer with Time2Vec Embedding

Hugo Cazaux*†, Ralph Rudd*, Hlynur Stefánsson*, Sverrir Ólafsson*, Eyjólfur Ingi Ásgeirsson*
*Department of Engineering, Reykjavik University, Menntavegur 1, 102 Reykjavik, Iceland
†Corresponding author: hugot@ru.is, Menntavegur 1, 102 Reykjavik, Iceland

***Abstract*—Time series prediction has been recently dominated by linear forecasters, in spite of a general effort to modify the transformer architecture to increase performance. The new addition of the inverted transformer, that outperforms all previous transformer-based and linear forecasters without modifying any of the base components of the Transformer, significantly advanced the field. This new inverted framework allows for a novel modification of the base components to further improve performance.**

This paper presents a new attention mechanism based on the non-stationary relationship between variables and integrates time2vec embedding to better represent the temporal variables. We observed an improvement in accuracy in most benchmark datasets, with at worst equal performance to the vanilla integration. We also propose an efficient training policy based on sparse attention to limit the quadratic complexity of modelling inter-variable relationships, and tools for further interpretability in the inverted framework.

***Impact Statement*—Timeseries prediction is a task that leverages past data to infer future values for a given variable. This simple idea holds large stakes when implemented in key industries such as staffing, failure prediction or electricity consumption. Traditional statistical approaches have now been overtaken by neural network-based methods. As the top performing models bear a lot of similarities with the extremely public large language models, timeseries prediction is in an incredible second mover position to leap forward. The model we introduce in this study borrows techniques from previous models and other fields to upgrade the predictive power of one of the current top performing models. This model is ready to be implemented in any timeseries dataset, with a clear strategy for parameters tuning. It also uses interpretability techniques to provide clarity and security when used in key industries where responsibility is paramount.**

***Index Terms*—Attention Mechanism, Forecasting, Inverted Transformer, Machine learning, Time series, Transformer**

I. Introduction

Transformers have revolutionized the natural language processing with their self-attention mechanism and layered feed-forward networks. The self-attention mechanism enables the model to weigh the importance of each input token in relation to others, allowing it to capture long-range dependencies, while the feed-forward networks refine these relationships across layers. Their application to time series forecasting has been lagging behind, especially in the context of larger lookback windows.

This lag was highlighted by the surprising effectiveness of linear forecasters, which outperformed the previous attempts at adapting the transformer architecture to time series prediction [1]. With an affordable computation cost and a strong base of interpretability, the linear forecasters outperformed the modified transformer architectures, especially for long-term predictions.

The inverted transformer (iTransformer) architecture is among the state-of-the-art models in time series analysis [2]. By inverting the typical duties of the attention mechanism and feed-forward networks of the standard transformer architecture, the architecture is better equipped to forecast series with larger lookback windows. The iTransformer currently ranks first in the long-term forecasting task of the Time Series Analysis benchmarks [3]. As the iTransformer does not introduce any adaptation to the basic components, this architecture also benefits from the tools developed for the original Transformer architecture.

Despite the success of this models, effectively handling non-stationarity and computational complexity remains a challenge in multivariable time series forecasting [4]. The integration of de-stationary mechanisms, like those introduced in the NSTransformer to model inter-tokens relationships, into architectures such as the iTransformer offers a potential solution to these issues. By learning scaling and shifting factors for inter-variable relationships, models can better adapt to non-stationary behaviors in the data.

Moreover, computational efficiency is crucial when dealing with high-dimensional time series data. Introducing sparsity into attention mechanisms, such as using top-k sparse attention, can significantly reduce computational complexity from $O(N^2)$ to $O(Nk)$, where N is the number of variables and k is a small constant. This allows the model to focus on the most relevant inter-variable relationships without incurring prohibitive computational costs.

This paper leverages the similarity between the original Transformer architecture and the iTransformer to adapt and extend Transformer-specific interpretability methods. Specifically, we explore the application of Chefer's generic method for transformer interpretability [5]. This paper reformulates the original method and adapts it to the inverted transformer architecture. The result is a continuous relevance map highlighting critical variables that are most influential in the predictive power of the model.

The core research questions in this article are the following: can de-stationary attention be extended to the iTransformer architecture? Can Time2Vec embedding improve the performance of the iTransformer? What improvements in forecasting performance and efficiency can be achieved through this integration?

The paper is structured as follow: Section 2 includes a contextualization and a review of the existing literature. Section 3 introduces the mathematical definition of the method. Section 4 displays the results of the forecasting using the NSiTransformer. Section 5 proposes an analysis of several components and mechanisms of the model. Section 6 presents the hardware used for training. Section 7 and 8 are the conclusion and future work of the study.

II. Background

The transformer architecture [6] has become a cornerstone of deep learning, particularly in natural language processing tasks. The self-attention mechanism allows the model to weight the importance of different tokens in a sequence relative to one another. This architecture is the foundation behind most of the mainstream models, such as ChatGPT [7], Mistral [8], and Llama [9]. The surge in research has provided fast improvement in parallelization [10] and diverse optimizations [11].

Various modification paradigms have been proposed to improve the accuracy of Transformer-based forecasters. Autoformer [12] and Informer [13] propose to replace the attention component respectively with an autocorrelation and sparse attention mechanisms. Crossformer [14] focuses on modeling the cross-time and cross-dimension dependency using a two-stage attention and modified hierarchical encoder-decoder architecture. Finally, PatchTST [15] and Non-Stationary Transformer (NSTransformer) [16] focused on the processing of time series using patching and stationarization respectively.

The inverted transformer (iTransformer) introduces no modification to the original Transformer components [6]. By inverting the duties of the attention mechanism and the feed-forward network, this architecture aims to reduce performance degradation and computation explosion in larger lookback windows. This recent model has been succesfully used to predict the useful life of Lithium-Ion batteries [17], earthquake detection [18] and predict sea surface temperature [19]

Interpretability methods for Transformers are sparse in the literature. Exploiting the raw attention weights to draw attention maps has been criticized for a limited contribution to the interpretability [20] [21]. The ConceptTransformer [22] proposed to modify the architecture for better explainability. Vision Transformer is the most prolific source of interpretability attempts, with neural tree decoder [23] and interpretability-aware training objectives [24]. The method used to get insights in this study is centered around an adaptation of a general interpretability technique to iTransformer [25]. This technique consists of building a relevancy maps of the different tokens using the gradients of the feed forward networks.

In this paper, we build upon these ideas by integrating the de-stationary attention mechanism and variable projector from the NSTransformer into the iTransformer framework. We further enhance computational efficiency by incorporating a sparse attention mechanism that computes scaling and shifting factors only for the top-k most relevant variable pairs. This approach aims to capture non-stationary inter-variable relationships more effectively while maintaining scalability for large-scale time series forecasting tasks.

III. Non-Stationary Inverted Transformer

A. Preliminaries

iTransformer: Given historical observations $X = \{x_1, x_2, \ldots, x_T\} \in \mathbb{R}^{T \times N}$ with T time steps and N variables, the goal is to predict the future S time steps $Y = \{x_{T+1}, x_{T+2}, \ldots, x_{T+S}\} \in \mathbb{R}^{S \times N}$. In the iTransformer, each time series of a variable is embedded into variable tokens, which are then utilized by the attention mechanism to capture multivariable correlations. The feed-forward network is applied to each variable token to learn nonlinear representations, and the final output is generated by projecting these representations back to the time series domain.

The process can be formulated as follows:

$$h_0^n = \text{Embedding}(X_{:,n}) \tag{1}$$

$$H^{l+1} = \text{TrmBlock}(H^l), \quad l = 0, \ldots, L-1 \tag{2}$$

$$\hat{Y}_{:,n} = \text{Projection}(h_L^n) \tag{3}$$

where $H = \{h_1, h_2, \ldots, h_N\} \in \mathbb{R}^{N \times D}$ contains N embedded tokens of dimension D. The functions Embedding : $\mathbb{R}^T \to \mathbb{R}^D$ and Projection : $\mathbb{R}^D \to \mathbb{R}^S$ are implemented by multi-layer perceptrons (MLP). The self-attention and feed-forward network operations in each Transformer block (TrmBlock) enable the model to learn complex dependencies across variables and time steps.

NSTransformer: addresses the challenges posed by non-stationary time series data, where statistical properties such as mean and variance change over time. It introduces a de-stationary attention mechanism that adjusts the attention computations to account for these changes, enhancing the model's ability to capture evolving patterns in the data.

In the NSTransformer, per-time-step scaling (τ_t) and shifting (δ_t) factors are learned to adjust the input:

$$\tilde{\mathbf{x}}_t = \tau_t \odot \mathbf{x}_t + \delta_t, \tag{4}$$

where $\mathbf{x}_t$ is the input at time step t, $\odot$ the Hadarmard product (element-wise multiplication) and $\tilde{\mathbf{x}}_t$ is the adjusted input.

NSiTransformer: The inverted architecture combined with the de-stationary factors proposed lead us to the name of Non-stationary Inverted Transformer, or NSiTransformer.

B. Components

This section details the components of the NSiTransformer.

Overall Model Architecture: Figure III-B presents the architecture of the model. The overall process of the proposed model can be summarized as follows:

1) **Normalization**: Normalize each variable time series using its mean and standard deviation.
2) **Embedding**:
   - Concatenate the normalized variable time series with the Time2Vec embeddings at each time step.
   - Apply a linear transformation to project the concatenated vectors into the model dimension D.
3) **Attention with De-Stationary Factors**:

- Compute preliminary attention scores between variable tokens.
- Select the top-k variable pairs for each variable based on these scores.
- Compute scaling and shifting factors $\tau_{i,j}$ and $\delta_{i,j}$ using the variable projector network for the selected pair of variables (i, j) for each of the top-k pairs.
- Adjust the attention scores using $\tau_{i,j}$ and $\delta_{i,j}$.
- Apply the attention mechanism to update variable tokens.

4) **Feed-Forward Network**: Apply position-wise feed-forward networks to the updated variable tokens.
5) **Projection**: Project back from the embedded dimension back to the projection length.
6) **De-Normalization**: Reintroduce the original scale and mean to the variable tokens to obtain the final output.

Normalization: To stabilize training and improve convergence, the input time series is normalized before being fed into the model. For each variable i, the mean μ_i and standard deviation σ_i are computed over the sequence length T:

$$\mu_i = \frac{1}{T}\sum_{t=1}^{T} X_{t,i}, \quad \sigma_i^2 = \frac{1}{T}\sum_{t=1}^{T}(X_{t,i} - \mu_i)^2 + \epsilon, \tag{5}$$

where $X_{t,i}$ is the $i-th$ variable at time t, and ϵ is a small constant to prevent division by zero. The normalized input $\tilde{\mathbf{X}}_{:,i}$ is then obtained by:

$$\tilde{\mathbf{X}}_{:,i} = \frac{\mathbf{X}_{:,i} - \mu_i}{\sigma_i}. \tag{6}$$

This normalization ensures that each variable has zero mean and unit variance, reducing the impact of scale differences between variables.

Embedding: Time2Vec [26] extends the concept of positional encoding by learning a vector representation of time that captures both linear and periodic patterns.

Given an hyperparameter d_{time}, for each time step t, the Time2Vec embedding $\mathbf{t}_t \in \mathbb{R}^{d_{\text{time}}}$ is defined as:

$$\begin{aligned}\mathbf{t}_t = \mathbf{w}_0 t + \mathbf{b}_0 + [&\sin(\mathbf{w}_1 t + \mathbf{b}_1), \sin(\mathbf{w}_2 t + \mathbf{b}_2), \\ &\ldots, \sin(\mathbf{w}_k t + \mathbf{b}_k)]\end{aligned} \tag{7}$$

where $\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_k \in \mathbb{R}$ and $\mathbf{b}_0, \mathbf{b}_1, \ldots, \mathbf{b}_k \in \mathbb{R}$ are learnable parameters, and $k = d_{\text{time}} - 1$ is the number of sine components.

The Time2Vec embedding captures both linear trends and periodic patterns, enhancing the model's ability to learn temporal dynamics.

Modeling Inter-variable Non-Stationary Relationships using τ and δ: To capture non-stationary relationships between variables, we introduce learned scaling ($\tau_{i,j}$) and shifting ($\delta_{i,j}$) factors for each pair of variables (i, j). These factors adjust the attention scores between variable tokens, allowing the model to adapt to changes in inter-variable relationships over time.

Attention Mechanism with De-Stationary Factors: The attention scores between variable tokens are computed as:

$$\text{scores}_{i,j} = \frac{\mathbf{q}_i \mathbf{k}_j^\top}{\sqrt{d_k}} \times \tau_{i,j} + \delta_{i,j}, \tag{8}$$

where $\mathbf{q}_i, \mathbf{k}_j \in \mathbb{R}^{d_k}$ are the query and key vectors for variable tokens i and j, respectively, and d_k is the dimension of the key vectors.

Variable Projector Network: The scaling and shifting factors $\tau_{i,j}$ and $\delta_{i,j}$ are computed using a single linear transformation of the concatenated embeddings of variable tokens i and j:

$$[\tau_{i,j}, \delta_{i,j}] = \mathbf{W}[\mathbf{h}_i; \mathbf{h}_j] + \mathbf{b}, \tag{9}$$

where $\mathbf{h}_i, \mathbf{h}_j \in \mathbb{R}^D$ are the embeddings of variable tokens i and j, $[\mathbf{h}_i; \mathbf{h}_j] \in \mathbb{R}^{2D}$ denotes their concatenation, $\mathbf{W} \in \mathbb{R}^{2\times 2D}$ is a learnable weight matrix, and $\mathbf{b} \in \mathbb{R}^2$ is a bias vector.

Sparse Computation with Top-k Selection: To reduce computational complexity from $O(N^2)$ to $O(Nk)$, we compute $\tau_{i,j}$ and $\delta_{i,j}$ only for the top-k most relevant variable pairs for each variable. The top-k variables are selected based on the preliminary attention scores:

$$\text{scores}_{i,j}^{\text{pre}} = \sum_{h=1}^{H} \mathbf{q}_i^{(h)} \left(\mathbf{k}_j^{(h)}\right)^\top, \tag{10}$$

where H is the number of attention heads, and $\mathbf{q}_i^{(h)}$, $\mathbf{k}_j^{(h)}$ are the query and key vectors for head h. For each variable i, we select the indices of the top-k variables j with the highest $\text{scores}_{i,j}^{\text{pre}}$.

De-Normalization: After the attention mechanism and updates to the variable tokens, we reintroduce the original scale and mean to obtain the final output. This de-normalization step ensures that the model's predictions are in the same scale as the original data and are interpretable.

The de-normalization is performed as:

$$\mathbf{h}_i^{\text{final}} = \mathbf{h}_i' \odot \sigma_i + \mu_i, \tag{11}$$

where $\mathbf{h}_i' \in \mathbb{R}^D$ is the updated variable token after the attention and feed-forward layers, and $\odot$ denotes element-wise multiplication.

IV. Experiments

Models: We harness the Time-Series-Library [27] and propose seven of the best performing models as our benchmark: iTransformer [2], PatchTST [15], Crossformer [14], TimesNet [28], DLinear [1], NSTransformer [16].

Datasets: We use the ETT, Weather, ECL and Traffic datasets included in Autoformer for long term forecasting. Table II details the features of the datasets.

TABLE II
DETAILED DATASET DESCRIPTIONS.

Task	Dataset	Dim	Dataset Size	Frequency
Forecasting (long-term)	ETT	7	(8545, 2881, 2881)	15min
	Weather	21	(36792, 5271, 10540)	10min
	ECL	321	(18317, 2633, 5261)	Hourly
	Traffic	862	(12185, 1757, 3509)	Hourly

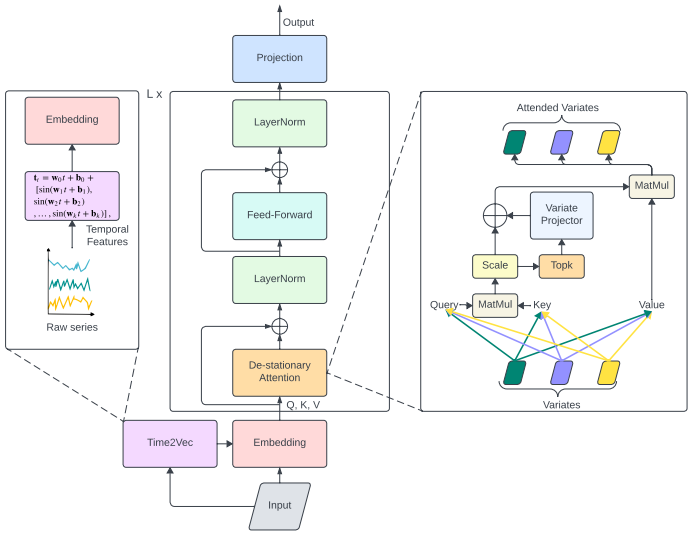


Fig. 1. Architecture of the proposed model. MatMul is the matrix multiplication. The temporal features are embedded using Time2vec, and the series is embedded. De-stationary attention is then applied. We then apply Layer Normalization and the Feed-Forward Network. The result is then projected to the prediction length.

Forecasting: Table I presents the full results in long-term forecasting of the NSiTransformer against the six benchmark models. NSiTransformer achieves state-of-the-art or near state-of-the-art performance in all 4 benchmark datasets.

V. Analysis

Ablation: We propose an ablation study to determine the contributing mechanisms in the NSiTransformer. Table III displays the result of this ablation study. We first remove Time2vec, then remove the de-stationary attention mechanism.

The ablation study highlights the mechanism that contributes the most in each experiment. In the ETT dataset, both Time2Vec and the De-stationary attention contribute depending on the prediction length. At $T = 96$ and $T = 336$, the de-stationary attention is driving the MSE down, while at $T = 192$ the combination of both mechanisms is best performing. The ECL dataset benefits the most from Time2Vec embedding, as the default self-attention performs as well as the NSiTransformer at most prediction length. The Weather dataset performs best when using the combination of both mechanisms at all prediction length, highlighting the relevancy of de-stationary attention and Time2Vec for predicting this dataset.

TABLE III
Ablation results for the NSiTransformer. The input sequence length is set to 96, and T is the prediction length. Avg is the average result of all four prediction lengths.

Models	T	NSiTransformer	W/o Time2Vec	W/o DSAttention
Metric		MSE	MSE	MSE
ETT	96	**0.292**	0.296	0.299
	192	**0.374**	0.382	0.378
	336	0.426	**0.420**	0.427
	720	0.420	0.423	**0.410**
	Avg	0.378	0.380	**0.377**
ECL	96	**0.147**	0.148	0.147
	192	**0.162**	0.162	0.162
	336	**0.175**	0.179	0.176
	720	**0.208**	0.213	0.208
	Avg	**0.173**	0.175	0.173
Weather	96	0.172	0.174	**0.171**
	192	**0.222**	0.225	0.224
	336	**0.278**	0.282	0.280
	720	**0.356**	0.359	0.360
	Avg	**0.257**	0.259	0.258

Mixed Floating Point Precision: Due to the possibly heavy computation at large k, we use mixed floating point computation for the ECL and Traffic dataset. Mixed floating point truncates Float16 to Float8 unless the supplementary precision is necessary.

TABLE I
FULL RESULTS FOR THE LONG-TERM FORECASTING TASK. THE INPUT SEQUENCE LENGTH IS SET TO 96 FOR ALL BASELINES, AND T IS THE PREDICTION LENGTH. AVG IS THE AVERAGE RESULT OF ALL FOUR PREDICTION LENGTHS. MSE STANDS FOR MEAN SQUARED ERROR AND MAE FOR MEAN ABSOLUTE ERROR.

Models	T	NSiTransformer		iTransformer		PatchTST		Crossformer		TimesNet		DLinear		Stationary	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETT	96	**0.292**	**0.347**	0.297	0.349	0.302	0.348	0.745	0.584	0.340	0.374	0.333	0.387	0.476	0.458
	192	**0.374**	**0.400**	0.380	0.400	0.388	0.400	0.877	0.656	0.402	0.414	0.477	0.476	0.512	0.493
	336	**0.426**	**0.432**	0.428	**0.432**	0.426	0.433	1.043	0.731	0.452	0.452	0.594	0.541	0.552	0.551
	720	**0.420**	**0.440**	0.427	0.445	0.431	0.446	1.104	0.763	0.462	0.468	0.831	0.657	0.562	0.560
	Avg	**0.378**	**0.404**	0.383	0.407	0.387	0.407	0.942	0.684	0.414	0.427	0.559	0.515	0.526	0.516
ECL	96	**0.147**	**0.238**	0.148	0.240	0.195	0.285	0.219	0.314	0.168	0.272	0.197	0.282	0.169	0.273
	192	**0.162**	**0.253**	**0.162**	**0.253**	0.199	0.289	0.231	0.322	0.184	0.289	0.196	0.285	0.182	0.286
	336	**0.175**	**0.268**	0.178	0.269	0.215	0.305	0.246	0.337	0.198	0.300	0.209	0.301	0.200	0.304
	720	**0.208**	**0.298**	0.225	0.317	0.256	0.337	0.280	0.363	0.220	0.320	0.245	0.333	0.222	0.321
	Avg	**0.173**	**0.264**	0.178	0.270	0.216	0.304	0.244	0.334	0.192	0.295	0.212	0.300	0.193	0.296
Traffic	96	**0.393**	**0.267**	0.395	0.268	0.544	0.359	0.522	0.290	0.593	0.321	0.650	0.396	0.612	0.338
	192	**0.414**	**0.275**	0.417	0.276	0.540	0.354	0.530	0.293	0.617	0.336	0.598	0.370	0.613	0.340
	336	**0.428**	**0.281**	0.433	0.283	0.551	0.358	0.558	0.305	0.629	0.336	0.605	0.373	0.618	0.328
	720	**0.459**	**0.300**	0.467	0.302	0.586	0.375	0.589	0.328	0.640	0.350	0.645	0.394	0.653	0.355
	Avg	**0.423**	**0.280**	0.428	0.282	0.555	0.362	0.550	0.304	0.620	0.336	0.625	0.383	0.624	0.340
Weather	96	**0.171**	**0.211**	0.174	0.214	0.177	0.218	0.158	0.230	0.172	0.220	0.196	0.255	0.173	0.223
	192	0.224	0.256	**0.221**	**0.254**	0.225	0.259	0.206	0.277	0.219	0.261	0.237	0.296	0.245	0.285
	336	0.281	0.297	**0.278**	**0.296**	0.278	0.297	0.272	0.335	0.280	0.306	0.283	0.335	0.321	0.338
	720	**0.356**	**0.348**	0.358	0.349	0.354	0.348	0.398	0.418	0.365	0.359	0.345	0.381	0.414	0.410
	Avg	**0.258**	**0.278**	**0.258**	0.279	0.259	0.281	0.259	0.315	0.259	0.287	0.265	0.317	0.288	0.314

TABLE IV
DIFFERENCE IN PERFORMANCE FOR ETT AND WEATHER WITH AND WITHOUT MIXED PRECISION.

Models	T	FP16	Mixed
Metric		MSE	MSE
ETT	96	**0.292**	0.293
	192	**0.374**	0.384
	336	**0.426**	0.427
	720	**0.420**	0.423
	Avg	**0.378**	0.381
Weather	96	**0.172**	**0.172**
	192	**0.222**	**0.222**
	336	**0.278**	0.282
	720	**0.356**	0.359
	Avg	**0.257**	0.259

Table IV displays the difference in performance for ETT and Weather with and without mixed precision. The MSE is equal or slightly higher in Mixed precision, indicating that mixed floating point is a valid option for larger datasets when computation can be a bottleneck.

Hyperparameters Sensitivity: We experiment with different values of d_{time} and top-k. Figure 2 presents the influence of top-k on the MSE of the Weather dataset. As the top-k grows, the MSE diminishes.

Fig. 2. Influence of top-k hyperparameter on MSE for Weather dataset.

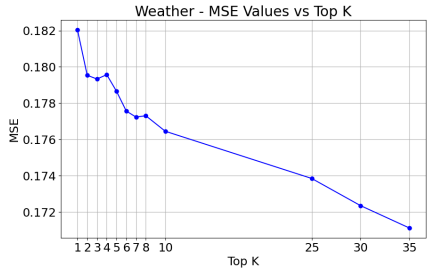


Figure 3 presents the influence of top-k on the MSE of the Weather dataset. There is a local minima at $k = 6$, indicating that a top-k value too high can also be detrimental to the performance of the model.

Fig. 3. Influence of top-k hyperparameter on MSE for ETT dataset.

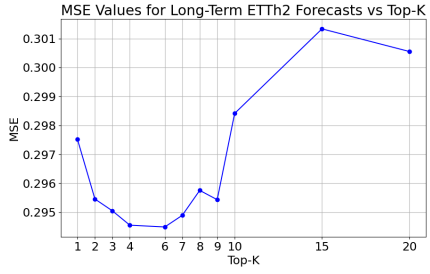


Figure 4 presents the influence of the hyperparameter d_{time} on the MSE for the dataset ECL. It appears that the model experiences an initial loss in performance, before reaching a

its best performance at 32 dimensions. As the number of dimensions increases, the model experiences worse performance.

Fig. 4. Influence of d_{time} hyperparameter on MSE for ECL dataset.

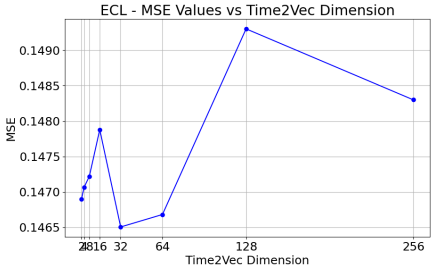
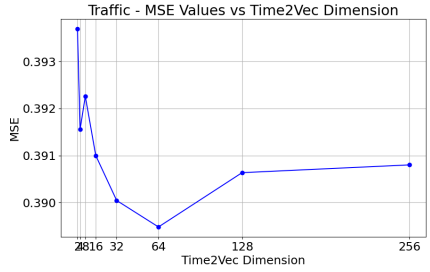


Figure 5 presents the influence of the hyperparameter d_{time} on the MSE for the dataset Traffic. The trend is more pronounced on the Traffic dataset, as increasing the number of Time2vec dimensions dimishes the loss up to 64 dimensions, but further increasing it leads to diminishing returns and a worse MSE.

Fig. 5. Influence of d_{time} hyperparameter on MSE for Traffic dataset.



These results lead to believe that efficient tuning of d_{time} is related to the number of features in the dataset. As the number of features increases, the best value for d_{time} increases.

Depth of the variable projector: We experiment with different depths for the variable projector network. Table V displays the forecasting result when using a simple linear projector network versus a deeper network with 128 hidden layers. We find that using a high number of hidden dimensions considerably increases computational overhead but can be beneficial, especially at higher prediction length. Numerous hidden layers can also lead the model to overfit the training dataset.

De-stationary Factors: We sample the tensors of τ and δ during testing and represent it as heatmaps. The de-stationary factors represent the evolving relationship between the features.

TABLE V
VARIABLE PROJECTOR DEPTH. THE INPUT SEQUENCE LENGTH IS SET TO 96, AND T IS THE PREDICTION LENGTH. AVG IS THE AVERAGE RESULT OF ALL FOUR PREDICTION LENGTHS.

Models	T	NSiTransformer No hidden layers		NSiTransformer 128 Hidden Layers	
Datasets		MSE	MAE	MSE	MAE
ETT	96	**0.292**	**0.345**	0.294	0.347
	192	**0.374**	**0.396**	0.382	0.400
	336	0.426	0.435	**0.419**	**0.432**
	720	0.420	0.443	**0.415**	**0.440**
	Avg	0.378	**0.404**	**0.377**	**0.404**
Weather	96	**0.171**	**0.211**	0.172	**0.211**
	192	0.224	0.256	**0.222**	**0.255**
	336	0.281	0.297	**0.278**	**0.296**
	720	**0.356**	**0.348**	0.357	0.349
	Avg	**0.258**	**0.278**	0.257	**0.278**

Fig. 6. De-stationary factors for ETT dataset in testing.

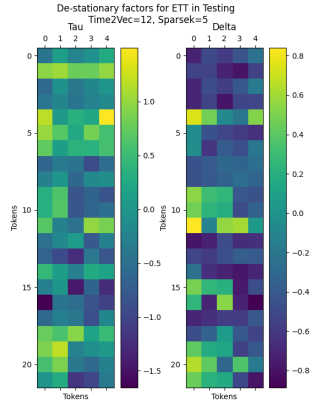


Figure 6 presents the de-stationary factors sampled during testing for the ETT dataset. The Tau shows the scaling of the attention scores, and the Delta the shifting of the attention scores. Token 1 is noteworthy as it is scaled up but shifted down. On the other hand, token 4 is both scaled and shifted up, indicating that the variable projector believes this token to have a strong relationship relative to other tokens.

Fig. 7. De-stationary factors for Weather dataset in testing.

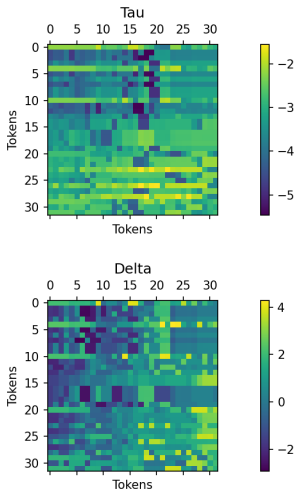


Figure 7 presents the de-stationary factors sampled during testing for the Weather dataset. Tokens 0, 4, 10, 20, 23 and 29 exhibit the same pattern, with a higher tau and delta than most. In both figures, the distribution of tau and delta appears to be similar, as evidenced by the different zones in the heatmaps.

Relevance maps: We provide supplemental interpretability by calculating the relevance of each token using Chefer et al [5] general technique adapted to the iTransformer [25]. Broadly, we initialize the relevancy map:

$$R_{vd} = I_{v \times d} \tag{12}$$

$$R_{vt} = 0_{v \times t} \tag{13}$$

$$R_{vv} = Concat(R_{vd}, R_{vt}) \tag{14}$$

For each layer, update the maps with the attention weights:

$$R_{vv} \leftarrow R_{vv} + \overline{A}_{vv} \odot R_{vv} \tag{15}$$

where $\overline{A}_{vv}$ represents the averaged attention weights for variable tokens, computed as:

$$\overline{A}_{vv} = \frac{1}{H} \sum_{h=1}^{H} \text{ReLU}\left(\nabla A^h_{vv} \odot A^h_{vv}\right) \tag{16}$$

And finally apply the learned transformation:

$$R_{\text{ff}} = \text{ReLU}(\nabla y \odot y) \tag{17}$$

$$R_{vv} \leftarrow R_{vv} + R_{\text{ff}} \tag{18}$$

All the scores are then normalized between 0 and 1. In figures 8 to 10, the figure on the left presents the total relevance for tokens with regular full attention ($k = 0$), with the total relevance sorted by descending order. The figure on the right presents the relevance of tokens with the k used in the results section, in their original order.

Fig. 8. Total relevance of tokens for ETT Dataset. Dataset features in blue, Time2vec features in red.

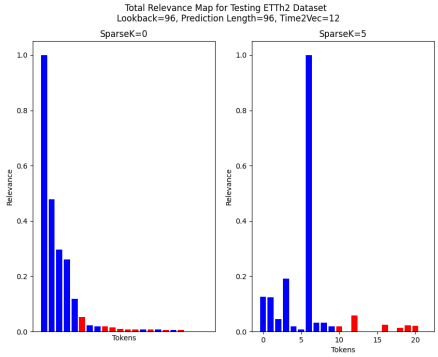


Figure 8 presents the total relevance of tokens for the ETTh2 dataset. Token 6 corresponds to the feature OT, which is also the predicted feature. The past values contribute the most to the prediction. Tokens 0, 1, 3, and 12 are also particularly relevant.

Fig. 9. Total relevance of tokens for Weather Dataset. Dataset features in blue, Time2vec features in red.

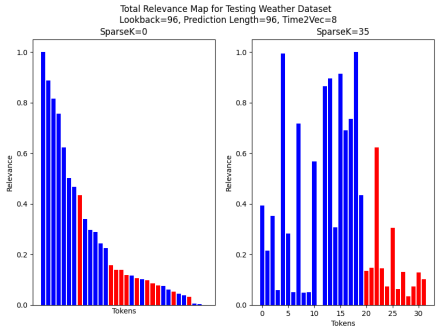


Figure 9 presents the total relevance of tokens for the Weather dataset. The relevance of the different tokens is significantly scattered, and 9 tokens reach a relevancy score ≥ 0.5. Notably, 2 of the embedded time features are prevalent, tokens 22 and 25. Those tokens are sine components representing the temporality dependencies of the dataset. The high relevancy of multiple tokens also corroborate the use of a higher k. In the ETT dataset, only a few tokens are relevant, and the model performs best at $k = 5$. On the other hand, the Weather dataset performs best at a significant $k = 35$, with only about 10 more features after Time2Vec. These results demonstrate that by observing the relevancy of the tokens we can determine experimentally the local best k for a given dataset.

Fig. 10. Total relevance of tokens for ECL Dataset. Dataset features in blue, Time2vec features in red.

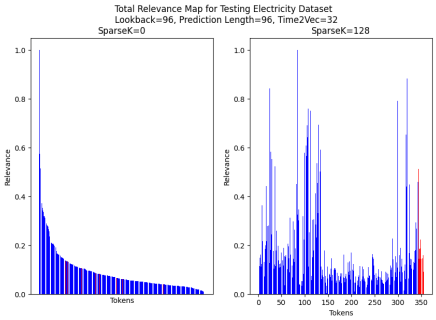


Figure 10 presents the total relevance of tokens for the ECL dataset. The large number of features of this dataset smooths out the distribution of relevance. The Time2Vec features are particularly prevalent in this dataset, as corroborated by Figure 4. We chose $k = 128$ as a compromise between highlighting the relevant features and computation time.

VI. Future work

Recent studies proposed alternative architectures for foundational models such as the multi-layer perceptrons used in this paper. Kolmogorov-Arnold networks (KAN) [29] in particular stand out as an ideal candidate for better interpretability and possibly better performance. A potential avenue for work would be to modify the original TrmBlock from the iTransformer and replace it with KANs. It is not well determined how this changed would alter the computational requirements. An initial idea could be to replace the variable projector linear layer with a KAN. Another approach could be to approximate the frozen MLP layers using KAN layers and try to deduce a closed-form expression of the model. Other types of data augmentation are also a promising avenue for work. Fundamentally, Time2Vec is a form of data augmentation designed for temporal features. Domain-specific techniques combined with autoencoders [30] could further refine the model.

The high interpretability that comes with the inverted framework is extremely valuable. Future work could use this model to demonstrate the contribution of a variable to the prediction, analogous to a variable to variable correlation. This transparency also broadens the field of applications to more critical industries. Law and finance, for instance, might value the accountability offered by more interpretable models while maintaining state-of-the-art performance. Other fields that suffer from the curse of dimensionality could use the innate interpretability to remove the variables that are not relevant enough, similar to a principal components analysis.

VII. Conclusion

This study proposes the NSiTransformer, an alternative architecture that places itself in the inverted transformer framework, and implements a custom attention mechanism and time embedding. The model performs at state-of-the-art level on the dataset benchmarks for long-term forecasting. The experiments highlight the efficiency of the attention mechanism and Time2vec in the different datasets and proposes a relevant use-case for each. The ETT, ECL and Traffic datasets proposed the largest gain in performance, with an average gain of 0.005 over the current state-of-the-art (vanilla iTransformer). The performance on the Weather dataset was equivalent on average to the iTransformer.

Specific interpretability techniques are also implemented to gain insights in the inner workings of the model. De-stationary factors are sampled during testing to keep track of the most important relationships between variables. Supplemental interpretability is provided through the use of token relevance, which helps determine which tokens are the most influential in the prediction. This information allows for an effective tuning strategy for the new hyperparameters d_{time} and k. The gains in longer prediction length with a deeper network indicate an increased flexibility for the NSiTransformer.

References

[1] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, pp. 11121–11128, 2023.

[2] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," *arXiv preprint arXiv:2310.06625*, 2023.

[3] Y. Wang, H. Wu, J. Dong, Y. Liu, M. Long, and J. Wang, "Deep time series models: A comprehensive survey and benchmark," 2024.

[4] R. Manuca and R. Savit, "Stationarity and nonstationarity in time series analysis," *Physica D: Nonlinear Phenomena*, vol. 99, no. 2-3, pp. 134–161, 1996.

[5] H. Chefer, S. Gur, and L. Wolf, "Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 397–406, 2021.

[6] A. Vaswani *et al.*, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[7] OpenAI, "Openai." https://openai.com/, 2024.

[8] Mistral, "Mistral 7b." https://mistral.ai/, 2024.

[9] Meta, "Llama 3.1." https://llama.meta.com/, 2024.

[10] Z. Zeng, C. Liu, Z. Tang, K. Li, and K. Li, "Acctfm: An effective intra-layer model parallelization strategy for training large-scale transformer-based models," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 12, pp. 4326–4338, 2022.

[11] V. A. Korthikanti, J. Casper, S. Lym, L. McAfee, M. Andersch, M. Shoeybi, and B. Catanzaro, "Reducing activation recomputation in large transformer models," *Proceedings of Machine Learning and Systems*, vol. 5, pp. 341–353, 2023.

[12] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in neural information processing systems*, vol. 34, pp. 22419–22430, 2021.

[13] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, pp. 11106–11115, 2021.

[14] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *The eleventh international conference on learning representations*, 2023.

[15] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730*, 2022.

[16] Y. Liu, H. Wu, J. Wang, and M. Long, "Non-stationary transformers: Exploring the stationarity in time series forecasting," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9881–9893, 2022.

[17] A. Jha, O. Dorkar, A. Biswas, and A. Emadi, "itransformer network based approach for accurate remaining useful life prediction in lithium-ion batteries," in *2024 IEEE Transportation Electrification Conference and Expo (ITEC)*, pp. 1–8, IEEE, 2024.

[18] L. Wang, Z. Li, Y. Chen, J. Wang, and J. Fu, "Maxent seismosense model: Ionospheric earthquake anomaly detection based on the maximum entropy principle," *Atmosphere*, vol. 15, no. 4, p. 419, 2024.

[19] W. Jia, S. Guan, and Y. Xue, "Tl-itransformer: Revolutionizing sea surface temperature prediction through itransformer and transfer learning," *Earth Science Informatics*, pp. 1–11, 2024.

[20] S. Jain and B. C. Wallace, "Attention is not explanation," *arXiv preprint arXiv:1902.10186*, 2019.

[21] S. Serrano and N. A. Smith, "Is attention interpretable?," *arXiv preprint arXiv:1906.03731*, 2019.

[22] M. Rigotti, C. Miksovic, I. Giurgiu, T. Gschwind, and P. Scotton, "Attention-based interpretability with concept transformers," in *International conference on learning representations*, 2021.

[23] S. Kim, J. Nam, and B. C. Ko, "Vit-net: Interpretable vision transformers with neural tree decoder," in *International conference on machine learning*, pp. 11162–11172, PMLR, 2022.

[24] Y. Qiang, C. Li, P. Khanduri, and D. Zhu, "Interpretability-aware vision transformer," *arXiv preprint arXiv:2309.08035*, 2023.

[25] S. Cazaux, Rudd, "Inverted transformers interpretability beyond attention visualization," in *International Joint Conference on Neural Networks*, 2025.

[26] S. M. Kazemi, R. Goel, S. Eghbali, J. Ramanan, J. Sahota, S. Thakur, S. Wu, C. Smyth, P. Poupart, and M. Brubaker, "Time2vec: Learning a vector representation of time," *arXiv preprint arXiv:1907.05321*, 2019.

[27] Y. Wang, H. Wu, J. Dong, Y. Liu, M. Long, and J. Wang, "Deep time series models: A comprehensive survey and benchmark," 2024.

[28] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," *arXiv preprint arXiv:2210.02186*, 2022.

[29] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, "Kan: Kolmogorov-arnold networks," *arXiv preprint arXiv:2404.19756*, 2024.

[30] B. Ning, S. Jaimungal, X. Zhang, and M. Bergeron, "Arbitrage-free implied volatility surface generation with variational autoencoders," *SIAM Journal on Financial Mathematics*, vol. 14, no. 4, pp. 1004–1027, 2023.

Appendix E

Article 5 - Controlled Log Returns Prediction Using NSiTransformer on ESG Enhanced Timeseries [43]

ARTICLE TEMPLATE

Controlled Log Returns Prediction Using NSiTransformer on ESG Enhanced Time Series

Hugo Cazaux[a,b] , Ralph Rudd[a], Hlynur Stefánsson[a], Sverrir Ólafsson[a], and Eyjólfur Ingi Ásgeirsson[a]

[a]Reykjavik University, Department of Engineering, Menntavegur 1, Reykjavik, 102, Iceland
[b]Corresponding author, email: hugot@ru.is

ARTICLE HISTORY
Compiled June 4, 2025

ABSTRACT
Environmental, social, and governance (ESG) ratings have been at the forefront of investment in the past decade. There are however several providers and discrepancies that lead to an absence of a standard and undermine the credibility of the ratings. In this study, we demonstrate not only the helpfulness of ESG ratings to bolster the predictive power of machine learning models, but also argue that the multiplicity of providers is actually an asset. Through the lens of the NSiTransformer, a state-of-the-art time series prediction model, we determine that adding ESG ratings not only improves the accuracy of the model, but the gain is further emphasized when adding more providers. Finally, using relevance, we determine the key scores used by the model to make a prediction, isolating the most important ESG metrics from the dataset.

KEYWORDS
ESG ratings, Finance, Machine Learning, Time Series Prediction.

1. Introduction

The finance sector is familiar with the implementation of emerging technologies to gain an edge over the rest of the market. High frequency trading (HFT) systems for example leveraged progress in computation and lower latency to edge the market through sheer speed [1]. Blockchain technology caught worldwide attention by proposing decentralized transactions and tokenized assets [2]. Machine learning and artificial intelligence are no exceptions, and have been implemented in finance as far back as the 1980s with projects such as the Fifth Generation Computer System in Japan [3]. In more recent years, artificial intelligence (AI) has seen a widespread adoption in virtually every domain. The integration of AI models and agents covers a broad number of use cases, including but not limited to: invoice generation, customer service, automated trading, portfolio balancing. Time series prediction is one of the most direct application that consists of training a model to learn past patterns in the data to infer future values based on a given number of features. The non-stationary inverted transformer (NSiTransformer) positions itself ideally for this task, as financial data can often be noisy and non-stationary.

Environmental, social and governance (ESG) ratings have been at the center of mod-

ern investment strategies, as new regulations are increasing the pressure on companies for sustainable long-term strategies. Their integration in predictive frameworks has been so far focused as a predicted variable, especially using natural language processing to measure the sentiment toward a company. Through the lens of interpretability, in this work we integrate the ESG ratings with key financial data and indicators to create a competitive model that also reveals insight about the predictive power of ESG ratings. Using the relevance maps, de-stationary factors and Shapley values we can estimate if the integration of ESG ratings is beneficial for the prediction, and to what scale. We also compare the performance to a model that does not integrate any extra-financial data to include a baseline.

The specific research questions developed in this study are: Can the NSiTransformer provide accurate predictions of company returns? Are ESG ratings beneficial for the predictive power of the model? Can interpretability highlight the role of the financial and extra financial parts of the dataset? Are there discrepancies in the predictive power of different providers?

This study is structured as follows: Section 2 presents a literature review of financial and ESG time series forecasting, Section 3 develops the methodology used to conduct the experiments, Section 4 evaluates the model in an exclusively financial dataset and sets the baseline, Section 6 uses relevance maps to harness insights from the results, and Section 7 is the conclusion of this study.

2. Literature Review

Time series are a classic data input for statistical analysis and their study has been formalized for decades [4]. One of the earliest method employed was the Gaussian process, which uses prior covariance function to model the behavior of the series [5] . A first improvement was exponential smoothing, which produces weighted averages of the past values that decay exponentially as the model gets further in the look-back window [6]. This model was then improved upon using state space to constrain the model non-linearly [7]. Finally, auto-regressive integrated moving average (ARIMA) stands out as the most commonly used statistical method to model univariate and multivariate time series [8]. ARIMA was used in a multitude of fields, including stock price forecasting [9], next-day electricity prices [10] and more recently on the COVID-2019 dataset [11]. Specialized statistical models are also developed for key industries: for instance, fractional Brownian motions and Hurst exponents [12] for foreign exchange rates.

Machine learning based approaches for financial time series forecasting have been extensively studied [13]. Artificial neural networks are the most dominant machine learning technique in this field. Support Vector Machines (SVM) also emerges as a popular algorithm, both in predicting future direction of stock price index [14] and future contracts evaluation [15]. More recent applications have been harnessing popular deep learning methods with great success [16]: Recurrent Neural Networks [17] [18] [19], Convolutional Neural Networks [20] [21] [22], Long-Short Term Memory Neural Networks [23] [24] [25] and Deep Reinforcement Learning [26] [27] [28]. The transformer architecture [29] has become the forefront of deep learning research: in finance, transformers have mostly been used for natural language processing and sentiment analysis using BERT models [30] [31] [32]. Other studies model stock volatility using modified transformer layers to improve forecasting models [33], use iterative dropout tests and batch size optimization [34], or harness multiplexed attention to increase the inference speed of the model [35].

Various modification paradigms have been proposed to improve the accuracy of Transformer-based forecasters. Crossformer [36] focuses on modeling the cross-time and cross-dimension dependency using a two-stage attention and modified hierarchical encoder-decoder architecture. PatchTST [37] and Non-Stationary Transformer (NSTransformer) [38] focused on the processing of time series using patching and stationarization respectively. The inverted transformer (iTransformer) introduces no modification to the original Transformer components [29]. By inverting the duties of the attention mechanism and the feed-forward network, this architecture aims to reduce performance degradation and computation explosion in larger lookback windows. This recent model has been succesfully used to predict the useful life of Lithium-Ion batteries [39], earthquake detection [40] and predict sea surface temperature [41]

The search for a relationship between sustainability and corporate performance can be traced back to the 1970s [42]. Scholars have studied the impact on branding [43], market longevity [44], and equity valuation [45]. Today, through integrated reporting [46] and a higher access to finance [47], companies understand that the market react to corporate ESG news [48]. There are however large discrepancies between how providers calculate ESG ratings [49] and even how investors use this information at the end of the chain [50]. Scholars have established a link between ESG performance and cost of debt [51], long-term financial performance [52] and higher potential for cumulative abnormal returns in specific markets [53]. As the more sustainably involved youth achieves higher levels of wealth, there is now a financial incentive to signal long-term commitment to future-proof policies [54].

ESG ratings have been integrated in machine learning and deep learning models in diverse studies in the literature [55] [56]. On one hand, studies attempted to predict the ESG performance of a given company using using random forests [57] [58], deep neural networks [59] [60], regression [61], and ensemble methods [62]. On the other hand, studies tried to evaluate financial data based on the ESG ratings using natural language processing for volatility prediction [63], deep learning with ESG and technical indicators [64], and ensemble methods [65] [66]. The use of transformers was centered around natural language processing [67] and sentiment analysis [68]. To the best of our knowledge, we found no transformer-based time series prediction model using ESG ratings as a feature.

3. Methodology

We need to design a training and evaluation framework for the NSiTransformer. To this, end we start by constituting a dataset based on recognized financial providers. To combine data from different stocks at the same timestamp, we define a walk-forward evaluation framework that learns iteratively on each stock.

3.1. *Dataset*

The data providers used in this study are the following:

- **Refinitiv** is a global leader in financial data and analytics, and serves as one of the primary sources for this study [69]. Refinitiv is also a significant ESG provider, covering over 80% of the global market capitalization with over 450 different ESG metrics. Refinitiv Eikon was extensively used throughout this study for both financial and ESG data. The three pillar scores: Environmental, Social

and Governance were used in the dataset, as well as the combined overarching score.

- **Sustainalytics** is a company that rates the sustainability of listed companies, based on their ESG performance. Their ratings are used by a variety of asset managers, asset owners and banks to define a sustainable investment strategy and create portfolios with strong ESG performers [70]. The ESG Risk Ratings are their flagship product and what we implemented in the dataset.
- **Sustainability Accounting Standards Board (SASB)** identify the most critical sustainability-related issues to investor in a wide array of industries. Since August 2022, the SASB standards have taken an important role by becoming the basis for the first two International Financial Reporting Standards (IFRS) dedicated to sustainability, IFRS S1 *General requirements for Sustainability-related Disclosures* [71] and S2 *Sustainability-related Disclosures* [72]. The first publication of the SASB standards date back to 2018, using a project-based model. The SASB issues are one-hot encoded based on each unique combination of material issues.

Table 1 shows a sample of the financial data extracted for Apple from 2005-12-05 to 2005-12-13. The initial dataset consists of financial data dating from **2005-12-05** to **2024-08-07**.

Date	Open	Low	High	Close	Volume
2005-12-05	2.17	2.15	2.19	2.16	5.84e8
2005-12-06	2.23	2.21	2.25	2.23	8.57e8
2005-12-07	2.24	2.20	2.24	2.23	6.79e8
2005-12-08	2.21	2.19	2.23	2.23	7.90e8
2005-12-09	2.24	2.21	2.25	2.24	5.55e8
2005-12-12	2.26	2.25	2.27	2.26	5.25e8
2005-12-13	2.25	2.24	2.27	2.26	4.94e8

Table 1.: Sample financial data for AAPL

Raw data can be enriched through calculated features and adjustments to better model financial dynamics and support robust predictions. In this study, data augmentation involves calculating log returns, controlling the returns using Fama-French 5 [73], and deriving technical indicators such as RSI [74], MACD, [75], and Bollinger Bands [76] from historical price and volume data to capture patterns, momentum, and volatility. These indicators are vital for machine learning models, providing features that encapsulate complex financial behaviors.

The predicted variable are the log returns controlled used Fama-French 5. The log returns are calculated and prepared in the same way as [77]. The goal of Fama French 5 is to isolate firm-specific characteristics and remove broader market effects. This model accounts for market-wide influences and fundamental financial drivers.

A central issue that needs to be addressed is the fundamental difference in granularity between financial data and ESG ratings. Financial data can easily be extracted down to the minute, and varies between the span of two queries. ESG ratings, on the other hand, are refreshed annually by Reuters [69] and, "regularly" by Sustainalytics [70]. We explored four different options to deal with this issue: regression, interpolation, autoencoders and forward fill. We decided to use forward fill which respects the methodology of the providers the most, as a query about the ESG ratings of a given company a quarter after financial disclosure would return the same value as

the start of the financial year, granted the company has not been affiliated to any scandal meaningful enough to warrant a change. Forward fill also does not introduce compoundable error that comes with an autoencoder or a regression model, or any interpolated assumption that the transition between ESG data points is smooth and linear.

3.2. *Model*

The model used is the Non-Stationary inverted Transformer (NSiTransformer) [78]. This model bases itself upon the inverted transformer architecture and provides a balance between interpretability and performance. The core idea is to use learned de-stationary factors to model the changing relationships between tokens. The model also implements Time2Vec [79] embedding to efficiently model time dependencies into higher dimensional spaces.

3.3. *Walk-Forward Time Series Evaluation*

One of the first challenge we faced when evaluating the models was the large discrepancy between older and the latest market data. In a classic time series prediction task, a proportion of the dataset (usually 70%) is used for training, another one for validation (10%) and the rest becomes unseen testing data (20%). This division respects the temporality of the time series, meaning that the 30% dedicated to validation and testing are the latest observations in the time series.

This is a fundamental problem in time series data, as significant events such as the Covid-19 pandemic or the rise of artificial intelligence fall into the validation or testing dataset, leading to worse performances in choppy markets. The first solution that was developed to counter this issue was to randomly sample the segments of data in the training, validation and testing datasets. This solution improved the performance of the model considerably but posed a major problem when the sequence and prediction length grew, as either data from multiple segments would leak onto each other and bias the model, or the safeguards put in place would significantly reduce the number of samples available. In order to conserve a high number of samples and take into account more recent data, we used a rolling time series evaluation mechanism in the training loop.

Consider a time series:

$$\{x_t\}_{t=1}^{T},$$

where T is the total number of time steps.

Let $L_{\text{train}}, L_{\text{val}}, L_{\text{test}}$ denote the (fixed) lengths of the training, validation, and test windows, respectively. We also define a *step size* Δ by which we will slide the window for each new iteration. For iteration $k = 1, 2, \ldots, K$, we define s_k (start index of iteration k) The training, validation, and test windows for iter-

ation k are given by:

$$W_{\text{train}}^{(k)} = \{\, t \mid s_k \le t < s_k + L_{\text{train}} \},$$

$$W_{\text{val}}^{(k)} = \{\, t \mid s_k + L_{\text{train}} \le t < s_k + L_{\text{train}} + L_{\text{val}} \},$$

$$W_{\text{test}}^{(k)} = \{\, t \mid s_k + L_{\text{train}} + L_{\text{val}} \le t < s_k + L_{\text{train}} + L_{\text{val}} + L_{\text{test}} \}.$$

After processing iteration k, we shift the start index by Δ, i.e. in each iteration k:

(1) **Training:** Fit the model using data $\{x_t \mid t \in W_{\text{train}}^{(k)}\}$.
(2) **Validation:** Tune hyperparameters or perform early stopping using $\{x_t \mid t \in W_{\text{val}}^{(k)}\}$.
(3) **Testing:** Evaluate final performance on $\{x_t \mid t \in W_{\text{test}}^{(k)}\}$.

We then collect the test metrics across iterations $k = 1, \ldots, K$ to obtain an estimate of the model's performance under various temporal regimes:

$$\text{Score} = \frac{1}{K} \sum_{k=1}^{K} \text{Metric}\big(\text{predictions on } W_{\text{test}}^{(k)},\ \text{ground truth on } W_{\text{test}}^{(k)}\big).$$

3.4. *Integration of Multiple Stocks*

Another benefit of the Walk-Forward method is the capacity to bundle several stocks in the training sets. Let us assume we have M distinct stocks, each represented by a time series $\{x_t^{(m)}\}_{t=1}^{T_m}$ for $m = 1, 2, \ldots, M$. When performing a walk-forward evaluation across multiple stocks, we start by adding to the dataset a one-hot encoded column representing the stocks' ticker. We then perform the walk-forward procedure on each stock's time series separately. For stock m, we define $W_{\text{train}}^{(k,m)}, W_{\text{val}}^{(k,m)}, W_{\text{test}}^{(k,m)}$ as the respective train, validation, and test windows for iteration k. We repeat the rolling-window approach from $k = 1$ to K for each stock m, creating $M \times K$ sets of evaluation results. A final performance measure can be computed by averaging (or otherwise aggregating) across all stocks and all iterations:

$$\text{Score} = \frac{1}{M\,K} \sum_{m=1}^{M} \sum_{k=1}^{K} \text{Metric}\Big(\hat{y}_t^{(k,m)},\ y_t^{(k,m)};\ t \in W_{\text{test}}^{(k,m)}\Big).$$

Alternatively, the model can be tested on any given stock's test segment for performance evaluation or comparison with a single stock model.

3.5. *Evaluation Metrics*

In this study, we quantify predictive performance using two standard error metrics: Mean Squared Error (MSE) and Mean Absolute Error (MAE). Let $\{\hat{y}_t\}$ denote the predictions on a test window W, and $\{y_t\}$ the corresponding ground-truth values:

$$\text{MSE}\big(\{\hat{y}_t\}, \{y_t\}\big) = \frac{1}{|W|} \sum_{t \in W} \big(\hat{y}_t - y_t\big)^2,$$

$$\mathrm{MAE}\big(\{\hat{y}_t\},\{y_t\}\big) = \frac{1}{|W|}\sum_{t\in W}|\hat{y}_t - y_t|.$$

3.6. *Benchmark Models*

We harness the time series-Library [80] and propose seven of the best performing models as our benchmark: iTransformer [81], PatchTST [37], Crossformer [36], TimesNet [82], DLinear [83], NSTransformer [38].

4. Predictive Performance on Financial Time Series

This section evaluates the performance of the NSiTransformer compared to the benchmark models on the financial dataset. Table 3 presents the results of the forecasting task for three distinct sectors—Finance, Industrials, and Technology, using the walk-forward approach and P the prediction length. The MSE and MAE shown are the average of testing results across all stocks **after** the model has completed pre-training. In the Finance and Industrials sectors, the NSiTransformer and its competitors achieve similar error levels. For instance at $P = 1$ in the Finance sector, the top three models are within ± 0.002 of MSE suggesting that the models all efficiently capture the dynamics present in the dataset. DLinear and Stationary on the other hand perform significantly worse than on a single stock compared to the other models. Stationary remains competitive at $P = 1$, but experiences a massive drop in performance at higher prediction length, being the worst performer for the Industrials and Technology sectors at $P = 14$ by a significant margin. A clear pattern that emerges is the uneven difficulty of sectors: Finance and Industrials appear to be significantly easier than Technology, especially at higher prediction length. This could indicate that this sector does not benefit from pre-training as much as the two others, or that predicting the erratic movement of Tech stocks at longer horizons is particularly difficult. We found that the median MSE for the NSiTransformer at $P = 14$ was 0.407, indicating very strong outliers on the tail end of the dataset. Two main outliers were identified: Nvidia's (NVDA) 1,789.12% growth over the past five years was especially hard to predict by all the models, with MSEs around 8.001 (NSiTransformer) or 10.53 (TimesNet), but the biggest culprit was Super Micro Computer Inc (SMCI), which hit an all time high four months before the cut-off of the dataset.

Table 2.: Results of benchmark models on Super Micro Computer Inc, NVDA and AMD. On day-after prediction, the model maintains a decent understanding of the upward pattern, but when the prediction length rises, the MSE increases drastically.

Models	P	SMCI		NVDA		AMD	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
NSiTransformer	1	23.74	2.17	2.21	0.517	0.296	0.229
	7	145.20	5.19	3.88	0.951	0.557	0.382
	14	257.84	7.14	8.001	1.32	0.851	0.501
iTransformer	1	23.83	2.20	2.19	0.525	0.302	0.236
	7	118.75	4.73	3.68	0.952	0.553	0.381
	14	254.88	7.17	8.13	1.27	0.852	0.496
PatchTST	1	23.39	0.19	2.17	0.562	0.294	0.238
	7	125.50	4.81	3.96	0.993	0.551	0.379
	14	271.90	7.06	8.91	1.33	0.909	0.507
TimesNet	1	25.33	2.35	2.23	0.553	0.303	0.242
	7	120.40	4.77	4.57	0.944	0.535	0.374
	14	307.52	7.41	10.53	1.38	0.760	0.475
DLinear	1	65.74	3.96	2.77	0.882	0.429	0.343
	7	208.36	6.71	7.49	1.36	0.761	0.484
	14	387.67	9.38	13.53	1.87	1.05	0.597
Stationary	1	31.82	2.55	2.29	0.553	0.298	0.235
	7	492.87	9.17	16.45	0.147	0.618	0.405
	14	520.49	10.35	16.09	1.47	0.820	0.492

Table 2 presents the MSE and MAE of all models when predicting SMCI. The MSEs and MAEs are exceedingly high and too polarizing to be included and as such SMCI will be excluded in the next tables referencing the Technology sector. For comparison, NVDA is the second highest MSE for all models within the Technology sector and AMD is in the top 10% of highest error for all models. Table 3 indicates that the NSiTransformer is competitive when predicting financial time series compared to the state-of-the-art models. Further comparison using ESG enhanced time series are available in Appendix B. In order to get comparable metrics between financial and ESG-enhanced time series, we need to define the largest subset of available data for both providers. This subset needs to be defined first temporally with the oldest possible data, and by availability with the highest possible amount of stocks.

Table 3.: Full results for the long-term forecasting task by industrial sectors. The input sequence length is set to 96 for all baselines, and P is the prediction length. Avg is the average result of all four prediction lengths. MSE stands for Mean Squared Error and MAE for Mean Absolute Error (Lower is better). In **bold**, the best performing model for each prediction length.

Models	P	NSiTransformer		iTransformer		PatchTST		TimesNet		DLinear		Stationary	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Finance	1	0.195	0.163	0.196	0.163	0.193	0.173	**0.191**	**0.167**	0.289	0.282	0.194	0.166
	7	0.336	0.278	0.357	0.289	**0.316**	**0.263**	0.331	0.274	0.426	0.357	0.377	0.296
	14	0.486	0.366	0.500	0.366	**0.447**	**0.348**	0.455	0.350	0.635	0.494	0.596	0.416
Industrials	1	0.175	0.179	0.178	0.191	**0.174**	**0.184**	0.178	0.190	0.252	0.250	0.177	0.182
	7	0.366	0.311	0.347	0.304	**0.343**	**0.304**	0.384	0.323	0.499	0.415	0.426	0.337
	14	0.563	0.404	0.546	0.403	**0.548**	**0.399**	0.695	0.451	0.729	0.530	0.809	0.490
Tech	1	0.380	0.201	0.381	0.203	**0.376**	**0.206**	0.395	0.209	0.809	0.320	0.440	0.202
	7	1.50	0.354	**1.29**	**0.347**	1.35	0.352	1.31	0.342	2.16	0.461	4.20	0.414
	14	2.57	0.469	**2.52**	**0.463**	2.69	0.471	2.94	0.465	3.77	0.609	4.64	0.527
Tech (No SMCI)	1	0.209	0.400	0.209	0.189	**0.207**	**0.192**	0.211	0.193	0.331	0.294	0.209	0.185
	7	0.442	0.320	**0.430**	**0.315**	0.437	0.319	0.433	0.310	0.630	0.415	0.610	0.350
	14	0.698	0.420	**0.673**	**0.414**	0.711	0.422	0.700	0.414	0.950	0.544	0.855	0.455

5. Towards More Comparable Metrics

In order to provide more comparable metrics between providers, we need to ensure that the data available to each model is at the same granularity. To this end we introduce a temporal and stock cut-off to provide comparable metrics.

We start by defining a time cut-off that includes the largest subset of available data for both providers. Sustainalytics ratings started in 2018, while Reuters ratings started as far back as 2006 for certain companies. Table 4 shows the number of data points removed from the Reuters dataset to align with Sustainalytics. We place our cut-off at December 1st, 2018. The average number of data points is slightly higher for Reuters due to certain companies not being covered by Sustainalytics at the exact cut-off. To ensure that the number of data points is even, we also propose a stock cut-off to align both datasets.

Table 4.: Average number of data points per stock pre- and post- time cut-off (01-12-2018).

Provider	Pre-cut	Post-cut
Sustainalytics	1420	1420
Reuters	4635	1530

We experiment on the intersection of available stocks for each companies. This process reduces the total number of stocks from 216 down to 184. Table 5 shows the number of stocks removed for each sector. The proportion of stocks removed is close to even for each sector.

Table 5.: Number of stocks removed due to missing or incomplete data in at least one other dataset.

Sector	Pre-cut	Post-cut
Finance	71	59
Industrials	77	68
Tech	68	57
Total	216	184

Table 6 shows the total number of total points for each sector after the datasets of both providers have been aligned to the same number of stocks.

Table 6.: Total number of data points.

Sector	Data points
Finance	83 780
Industrials	96 560
Tech	80 940
Total	261 280

Table 7 evaluates the NSiTransformer's performance using a consistent subset of 184 stocks (down from 216) across three different types of ESG datasets—financial-only, Sustainalytics, and Reuters—in the Finance, Industrials, and Technology sectors (excluding SMCI). By harmonizing the sample, this comparison isolates the impact of ESG data quality on forecasting accuracy. We also train a model (S+R in Table

7) using both providers' metrics, since the harmonization of the sample allows for no missing data.

The analysis reveals consistent forecasting gains when ESG data supplements financial metrics. Across all sectors and prediction windows, ESG integration reduces errors relative to financial-only baselines. In Finance, for instance, $P = 1$ forecasts improve from an MSE of 0.322 (no ESG) to 0.255 with Sustainalytics and 0.226 with Reuters, all the way down to 0.203 with S+R. This hierarchy also scales at longer horizons: at $P = 14$, S+P forecasts achieve an MSE of 0.396, outperforming Reuters (0.401) , Sustainalytics (0.463) and the baseline (0.550).

In Industrials, Reuters emerges as the superior ESG source. its $P = 1$ MSE (0.123) undercuts both S+R (0.128), Sustainalytics (0.159) and the baseline (0.175). This pattern maintains at $P = 7$ and $P = 14$, where the Reuters trained model maintains the top spot. Although Sustainalytics improves the No ESG approach, this sector exemplifies the necessity to test multiple combinations of data when implementing predictive models in real life scenario, as the addition of new features does not necessarily equate better performance.

In Technology, the same pattern as Finance emerges: No ESG is the baseline model and performs the worst. Individually, Sustainalytics and Reuters both help reduce the MSE, despite Reuters clearly promoting the model more, but S+P emerges as the clear top performer in all sectors, for all prediction lengths. These results indicate that the promotion provided by each ESG provider are different, as there is a compounding effect when used in conjunction.

Table 7.: Results with forecasting of NSiTransformer for Finance, Industrials and Technology sector with a temporal cut-off in 2018 and stock cut-off at P=1, 7 and 14. S+R stands for Sustainalytics+Reuters and was trained with both ESG metrics available. In **bold**, the best performing model for each prediction length.

Models	P	No ESG		Sustainalytics		Reuters		S+R	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Finance	1	0.322	0.189	0.255	0.163	0.226	0.136	**0.203**	**0.118**
	7	0.449	0.292	0.359	0.220	0.332	0.200	**0.317**	**0.182**
	14	0.550	0.357	0.463	0.280	0.401	0.252	**0.396**	**0.228**
Industrial	1	0.175	0.187	0.159	0.155	**0.123**	**0.132**	0.128	0.121
	7	0.311	0.305	0.268	0.243	**0.210**	**0.211**	0.214	0.192
	14	0.457	0.396	0.410	0.320	**0.311**	**0.276**	0.326	0.255
Tech (No SMCI)	1	0.207	0.207	0.164	0.150	0.146	0.142	**0.131**	**0.119**
	7	0.391	0.335	0.328	0.257	0.282	0.237	**0.260**	**0.204**
	14	0.567	0.420	0.489	0.327	0.405	0.299	**0.400**	**0.266**

The results imply Sustainalytics and Reuters capture distinct but overlapping facets of ESG risk. Their combination appears to filter out some of the provider-specific biases—for instance, Sustainalytics' noisier Industrials metrics are counterbalanced by Reuters' cleaner signal, while Reuters' potential blind spots in Technology volatility are mitigated by Sustainalytics' complementary data. The differences in methodology exposed in 3.1 could be a determining factor in the compounding effect observed. This synergy elevates the NSiTransformer's predictive capacity, particularly in longer-horizon forecasts where the modeling challenges are amplified. This is a significant result indicating to practitioners trying to get an edge on the market that ESG signal is worth including in their model. Beyond the boost in predictive power, the inte-

gration of ESG metrics is in line with the new generation of retail investors that have been sensitized to environmental issue and wish to bring their capital to sustainable companies. The computational cost added is also minimal: only 6 features for each provider (see Appendix C) is enough to upgrade the NSiTransformer predictive power. ESG providers tend to offer a large catalog of data, which can be overwhelming and ultimately deter their integration in machine learning models to avoid the curse of dimensionality. By focusing on the high level ratings, we maintain a broad picture of the sustainability footprint of a company without over-saturating the dataset with exogenous variables. Collectively, the experiment demonstrates that ESG integration strategies transcend simple additive benefits. By merging providers, models like NSiTransformer can exploit inter-dataset correlations to dampen noise and amplify sectorally relevant signals—a critical advantage in multi-horizon forecasting where isolated data sources may inadequately capture complex market inter-dependencies. Multiple providers are also a hedge against biased methodology in a given sample, which due to a lack of standardization has been established to be an issue in ESG ratings.

6. Relevance Maps

We provide supplemental interpretability by calculating the relevance of each token using Chefer et al [84] general technique adapted to the iTransformer [85]. Tables detailing the correspondence between tokens and features for each experiment can be found in Appendix C. Further interpretability is provided in Appendix D using de-stationary factors.

Figure 1a and 1b present the relevance maps for the models trained on Industrials at $P = 1$. In the No ESG experiment, the top relevant tokens are 5, 9 and 14. Token 5 corresponds to the Log returns before controlling, which is surprising considering that this experiment predicts token 6, which appears to be not as relevant. A possible explaining factor is the embedding of market movement in log returns: since the model is trained on multiple stocks, it could find beneficial when making a prediction to consider the uncontrolled returns to gauge the market. Token 9 is the MACDs signal line, and paired with token 8 showing a respectable relevance on its own, MACD clearly emerges as the most relevant financial indicator in the dataset. Token 14 is a special case: although it is highly relevant, this token represents the one-hot encoded ticker, and as such never changes throughout the sequence fed into the model. The most likely explanation for this relevance is the identification of a stock for the model. Although there are common patterns, when trained on all the stocks for a given sector (here Industrials) the model has to determine which stock it is predicting. In No ESG, the ticker serves as an anchor for the model to tailor the prediction to a given stock. The temporal tokens, although all slightly relevant, do not present a strong dominance between the raw data and the Time2vec embeddings.

When adding the Reuters ESG ratings, the relevance of the ticker plummets. The log returns and MACD tokens remain highly relevant, token 5 being even more dominating than in No ESG. Token 4, the volume, also increases in relevance. But the most interesting observation comes from token 20, a newly introduced token that encodes the SASB score. The SASB score is an encoded number that designates which issues are recognized to be material for a given company. Similarly to the ticker, this data point remains the same, and is likely to contributing to the recognition of the stock being predicted. This relevance score also means the model buckets SASB similar companies,

which is in line with the findings in [77]. The other Reuters tokens are also slightly relevant, especially the Environmental pillar score. In this study, Industrials was most correlated with the Environmental sector, which is coherent with what intuition would tell for a sector that includes Caterpillar (mining and construction), Union Pacific (freight hauling) or Boeing (aircraft manufacturer). This result echoes how simpler tools such as correlation can infer behaviors that will be found in far more complicated architectures such as the NSiTransformer.

Figure 1c and 1d present the relevance maps for the models trained on Finance at $P = 7$. As the prediction length increases, we find similarities and discrepancies compared to the previous maps. In 1c, the ticker is once again extremely relevant at first. Token 8 and 9, corresponding to the MACD and MACDs, maintain high relevance, with token 9 becoming the most relevant. The controlled returns, encoded in token 6, are however far more relevant than before. As the temporal horizon for prediction increases, it is likely that the model starts favoring historical data of the target value. As the relevance of the uncontrolled returns are still high, the hypotheses that the model uses uncontrolled returns to gauge the market still holds up.

As shown in 1d, The SASB code clearly remains pivotal for the predictive power, but the model uses a much broader array of tokens to fuel its prediction. For instance, the historical controlled log returns in token 6 gain relevance, similarly to the other components of MACD and RSI. Through the relevance map, we can discern the strategy used by the model: identify which group of stock is being predicted with the SASB code, and use the historical financial data to produce an adapted result. The Reuters ESG scores also start demonstrating sizable relevance, with the Environmental Pillar Score and Social Pillar Score displaying particularly high relevance. This broader approach could be the reason for the promotion in performance between the Sustainalytics-only and both providers model.

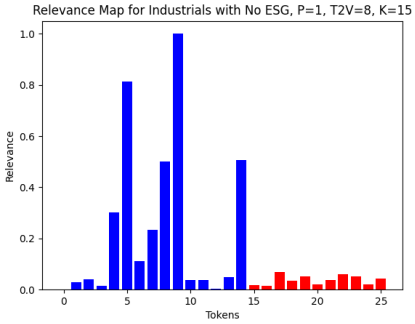
Figure 1e and 1f presents the relevance maps for the models trained on Technology at $P = 14$. At this prediction length, the ticker is no longer as relevant as in previous experiments. This could be an indication of a broader strategy that does not tailor the prediction as much as other model, but rather uses a combination of historical data and technical indicators to infer general behavior. This strategy can lead to the model underfitting, and explain why this model performed worse than the ones using ESG data. The historical data of controlled returns have taken over the log returns, but the most relevant token remains the MACD signal line (token 9). This consistent behavior throughout all experiments indicates that the models are extremely receptive to the MACD model, and this technical indicator is likely to yield improvements in other financial predictors.

The S + R model closely maintains the structure of No ESG for financial data. This further reinforces the thesis that ESG ratings are beneficial for the predictive power of the model, since the structure of financial tokens remains sensibly similar. The same pattern of close to equally relevant tokens for the ticker and SASB indicates that the S + R model adopts a more targeted strategy compared to No ESG based on the general inference strategy with more targeting. Sustainalytics' ESG Risk Score is slightly less relevant in this model, but it is to the benefit of Reuters' ESG Score and Governance score. The model appears to favor general scores over specific subcategories in the majority of cases, however Governance is an sensitive pillar to Technology companies, as the companies within this sector are under constant scrutinization from the public due to their high profile.

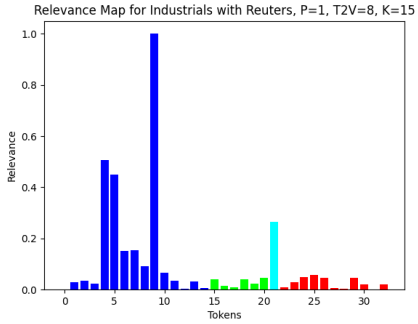
Finally, despite the model's performance being sensitive to the number of time embedded dimension, we found no significant difference in how the temporal tokens

were employed in longer prediction horizons. The embedded dimensions are in all cases slightly relevant without a particular dominance. The raw data of the days, which could indicate periodicity in the data, are not particularly relevant either regardless of the prediction horizon

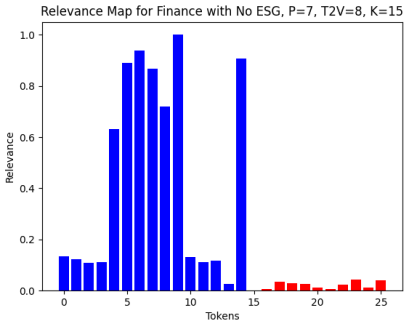
Relevance maps emerge as a key tool to not only identify the most relevant features but also infer the strategy used by the model to make a prediction. These strategies can then be assessed based on domain expertise and results on unseen data, in order to ensure that the model is not underfitting or overfitting. As such, this technique not only promotes interpretability in transformer-based models, but also informs practitioners on the key metrics to maximize the performance of their model.



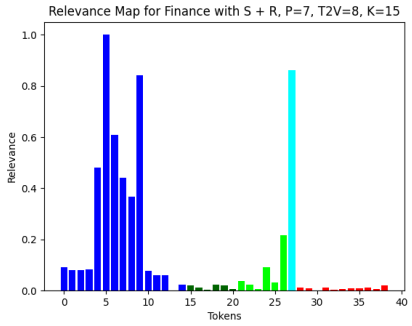
(a) No ESG



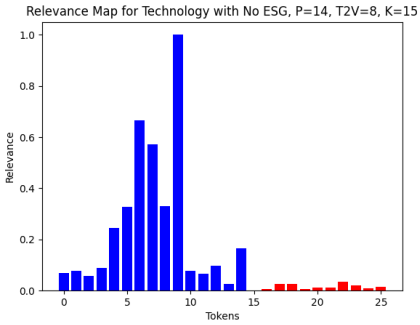
(b) Reuters



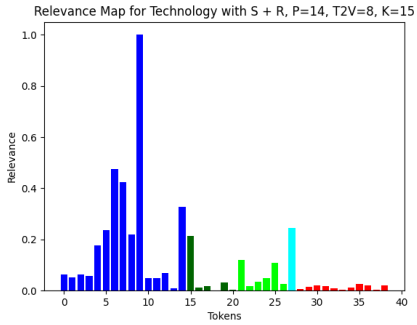
(c) No ESG



(d) Sustainalytics + Reuters



(e) No ESG



(f) Sustainalytics + Reuters

Figure 1.: Relevance Maps for Industrials P=1 (top), Finance P=7 (middle) and Technology P=14 (bottom). In blue financial features, lime Reuters features, cyan SASB, red temporal features, T2V=8, K=15..

7. Conclusion

In this study, we established that incorporating ESG data into financial datasets enhances the predictive power of the NSiTransformer. The best performing model integrated ESG data from three different providers, reinforcing the idea that the subjectivity of ESG ratings can be alleviated through multiple rating agencies. This study further corroborates the discrepancies in ESG ratings between agencies, as the inclusion of different providers yielded different results, despite clearly defining a common ground for evaluation.

The NSiTransformer provided accurate predictions, and saw an improvement in performance when the dataset was augmented with ESG data, indicating their benefits for the predictive power of the model. Through the lens of interpretability, we determined the importance of the SASB categorization replacing the ticker as an identifier for companies. Interpretability also allowed us to study the strategy adopted by the NSiTransformer to make a prediction and highlighted the role of the financial and extra financial parts of the dataset. The model employs a general strategy based on the financial data and targets specifically the company using the categorical variables.

Future work should try to propose different embeddings of the EGS ratings, and include other providers. A possible angle could be to study the interaction of different predicted variables with ESG ratings. From a model point of view, building larger models that encompass more stocks and financial sectors could lead to better performance. New model-specific techniques can also be explored, notably with Kolmogorov-Arnold networks to replace multi-layer perceptrons.

References

[1] I. Aldridge, *High-frequency trading: a practical guide to algorithmic strategies and trading systems*. John Wiley & Sons, 2013.

[2] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, "Blockchain," *Business & information systems engineering*, vol. 59, pp. 183–187, 2017.

[3] E. Feigenbaum and H. Shrobe, "The japanese national fifth generation project: Introduction, survey, and evaluation," *Future Generation Computer Systems*, vol. 9, no. 2, pp. 105–117, 1993, FGCS Conference, ISSN: 0167-739X. DOI: https://doi.org/10.1016/0167-739X(93)90003-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0167739X93900038.

[4] T. W. Anderson, *The statistical analysis of time series*. John Wiley & Sons, 2011.

[5] S. Brahim-Belhouari and A. Bermak, "Gaussian process for nonstationary time series prediction," *Computational Statistics & Data Analysis*, vol. 47, no. 4, pp. 705–712, 2004.

[6] E. S. Gardner Jr, "Exponential smoothing: The state of the art," *Journal of forecasting*, vol. 4, no. 1, pp. 1–28, 1985.

[7] R. Hyndman, A. B. Koehler, J. K. Ord, and R. D. Snyder, *Forecasting with exponential smoothing: the state space approach*. Springer Science & Business Media, 2008.

[8] S. L. Ho and M. Xie, "The use of arima models for reliability forecasting and analysis," *Computers & industrial engineering*, vol. 35, no. 1-2, pp. 213–216, 1998.

[9] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, "Stock price prediction using the arima model," in *2014 UKSim-AMSS 16th international conference on computer modelling and simulation*, IEEE, 2014, pp. 106–112.

[10] J. Contreras, R. Espinola, F. J. Nogales, and A. J. Conejo, "Arima models to predict next-day electricity prices," *IEEE transactions on power systems*, vol. 18, no. 3, pp. 1014–1020, 2003.

[11] D. Benvenuto, M. Giovanetti, L. Vassallo, S. Angeletti, and M. Ciccozzi, "Application of the arima model on the covid-2019 epidemic dataset," *Data in brief*, vol. 29, p. 105 340, 2020.

[12] M. Garcin, "Estimation of time-dependent hurst exponents with variational smoothing and application to forecasting foreign exchange rates," *Physica A: statistical mechanics and its applications*, vol. 483, pp. 462–479, 2017.

[13] B. Krollner, B. Vanstone, and G. Finnie, "Financial time series forecasting with machine learning techniques: A survey," in *European Symposium on Artificial Neural Networks: Computational Intelligence and Machine Learning*, 2010, pp. 25–30.

[14] K.-j. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1-2, pp. 307–319, 2003.

[15] F. E. Tay and L. Cao, "Application of support vector machines in financial time series forecasting," *omega*, vol. 29, no. 4, pp. 309–317, 2001.

[16] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," *Applied soft computing*, vol. 90, p. 106 181, 2020.

[17] J. Wang, S. Hong, Y. Dong, Z. Li, and J. Hu, "Predicting stock market trends using lstm networks: Overcoming rnn limitations for improved financial forecasting," *Journal of Computer Science and Software Applications*, vol. 4, no. 3, pp. 1–7, 2024.

[18] S. Hansun and J. C. Young, "Predicting lq45 financial sector indices using rnn-lstm," *Journal of Big Data*, vol. 8, no. 1, p. 104, 2021.

[19] K. Pawar, R. S. Jalem, and V. Tiwari, "Stock market price prediction using lstm rnn," in *Emerging Trends in Expert Applications and Security: Proceedings of ICETEAS 2018*, Springer, 2019, pp. 493–503.

[20] E. Hoseinzade and S. Haratizadeh, "Cnnpred: Cnn-based stock market prediction using a diverse set of variables," *Expert Systems with Applications*, vol. 129, pp. 273–285, 2019.

[21] M. U. Gudelek, S. A. Boluk, and A. M. Ozbayoglu, "A deep learning based stock trading model with 2-d cnn trend detection," in *2017 IEEE symposium series on computational intelligence (SSCI)*, IEEE, 2017, pp. 1–8.

[22] Y.-C. Chen and W.-C. Huang, "Constructing a stock-price forecast cnn model with gold and crude oil indicators," *Applied Soft Computing*, vol. 112, p. 107 760, 2021.

[23] S. Siami-Namini and A. S. Namin, "Forecasting economics and financial time series: Arima vs. lstm," *arXiv preprint arXiv:1803.06386*, 2018.

[24] J. Cao, Z. Li, and J. Li, "Financial time series forecasting model based on ceemdan and lstm," *Physica A: Statistical mechanics and its applications*, vol. 519, pp. 127–139, 2019.

[25] A. H. Bukhari, M. A. Z. Raja, M. Sulaiman, S. Islam, M. Shoaib, and P. Kumam, "Fractional neuro-sequential arfima-lstm for financial market forecasting," *Ieee Access*, vol. 8, pp. 71 326–71 338, 2020.

[26] X.-Y. Liu, H. Yang, Q. Chen, *et al.*, "Finrl: A deep reinforcement learning library for automated stock trading in quantitative finance," *arXiv preprint arXiv:2011.09607*, 2020.

[27] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 3, pp. 653–664, 2016.

[28] K. Lei, B. Zhang, Y. Li, M. Yang, and Y. Shen, "Time-driven feature-aware jointly deep reinforcement learning for financial signal representation and algorithmic trading," *Expert Systems with Applications*, vol. 140, p. 112 872, 2020.

[29] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[30] K. Mishev, A. Gjorgjevikj, I. Vodenska, L. T. Chitkushev, and D. Trajanov, "Evaluation of sentiment analysis in finance: From lexicons to transformers," *IEEE access*, vol. 8, pp. 131 662–131 682, 2020.

[31] D. Othan, Z. H. Kilimci, and M. Uysal, "Financial sentiment analysis for predicting direction of stocks using bidirectional encoder representations from transformers (bert)

and deep learning models," in *Proc. int. conf. innov. intell. technol*, vol. 2019, 2019, pp. 30–35.

[32] R. Pan, J. A. García-Díaz, F. Garcia-Sanchez, and R. Valencia-García, "Evaluation of transformer models for financial targeted sentiment analysis in spanish," *PeerJ Computer Science*, vol. 9, e1377, 2023.

[33] E. Ramos-Pérez, P. J. Alonso-González, and J. J. Núñez-Velázquez, "Multi-transformer: A new neural network-based architecture for forecasting s&p volatility," *Mathematics*, vol. 9, no. 15, p. 1794, 2021.

[34] C. Yañez, W. Kristjanpoller, and M. C. Minutolo, "Stock market index prediction using transformer neural network models and frequency decomposition," *Neural Computing and Applications*, pp. 1–21, 2024.

[35] C. Xu, J. Li, B. Feng, and B. Lu, "A financial time-series prediction model based on multiplex attention and linear transformer structure," *Applied Sciences*, vol. 13, no. 8, p. 5175, 2023.

[36] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *The eleventh international conference on learning representations*, 2023.

[37] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730*, 2022.

[38] Y. Liu, H. Wu, J. Wang, and M. Long, "Non-stationary transformers: Exploring the stationarity in time series forecasting," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9881–9893, 2022.

[39] A. Jha, O. Dorkar, A. Biswas, and A. Emadi, "Itransformer network based approach for accurate remaining useful life prediction in lithium-ion batteries," in *2024 IEEE Transportation Electrification Conference and Expo (ITEC)*, IEEE, 2024, pp. 1–8.

[40] L. Wang, Z. Li, Y. Chen, J. Wang, and J. Fu, "Maxent seismosense model: Ionospheric earthquake anomaly detection based on the maximum entropy principle," *Atmosphere*, vol. 15, no. 4, p. 419, 2024.

[41] W. Jia, S. Guan, and Y. Xue, "Tl-itransformer: Revolutionizing sea surface temperature prediction through itransformer and transfer learning," *Earth Science Informatics*, pp. 1–11, 2024.

[42] M. Friedman, "The social responsibility of business is to increase its profits," in *Corporate ethics and corporate governance*, Springer, 1970, pp. 173–178.

[43] M. T. Lee, R. L. Raschke, and A. S. Krishen, "Signaling green! firm esg signals in an interconnected environment that promote brand valuation," *Journal of Business Research*, vol. 138, pp. 1–11, 2022, ISSN: 0148-2963. DOI: https://doi.org/10.1016/j.jbusres.2021.08.061. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0148296321006287.

[44] I. Fafaliou, M. Giaka, D. Konstantios, and M. Polemis, "Firms' esg reputational risk and market longevity: A firm-level analysis for the united states," *Journal of Business Research*, vol. 149, pp. 161–177, 2022, ISSN: 0148-2963. DOI: https://doi.org/10.1016/j.jbusres.2022.05.010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0148296322004337.

[45] G. Giese, L.-E. Lee, D. Melas, Z. Nagy, and L. Nishikawa, "Foundations of esg investing: How esg affects equity valuation, risk, and performance," *The Journal of Portfolio Management*, vol. 45, no. 5, pp. 69–83, 2019.

[46] G. Serafeim, "Integrated reporting and investor clientele," *Journal of Applied Corporate Finance*, vol. 27, no. 2, pp. 34–51, 2015.

[47] B. Cheng, I. Ioannou, and G. Serafeim, "Corporate social responsibility and access to finance," *Strategic management journal*, vol. 35, no. 1, pp. 1–23, 2014.

[48] G. Serafeim and A. Yoon, "Which corporate esg news does the market react to?" *Financial Analysts Journal*, vol. 78, no. 1, pp. 59–78, 2022.

[49] D. M. Christensen, G. Serafeim, and A. Sikochi, "Why is corporate virtue in the eye of the beholder? the case of esg ratings," *The Accounting Review*, vol. 97, no. 1, pp. 147–175, 2022.

[50] A. Amel-Zadeh and G. Serafeim, "Why and how investors use esg information: Evidence from a global survey," *Financial analysts journal*, vol. 74, no. 3, pp. 87–103, 2018.

[51] S. Boccaletti and G. G. and, "Esg performance, institutional factors, and the cost of debt," *Journal of Sustainable Finance & Investment*, vol. 0, no. 0, pp. 1–29, 2025. DOI: 10.1080/20430795.2025.2489386. eprint: https://doi.org/10.1080/20430795.2025.2489386. [Online]. Available: https://doi.org/10.1080/20430795.2025.2489386.

[52] C. Shrestha, P. Andrikopoulos, and N. P. A. and, "Does it matter to be a part of the sustainability index?" *Journal of Sustainable Finance & Investment*, vol. 0, no. 0, pp. 1–22, 2024. DOI: 10.1080/20430795.2024.2401357. eprint: https://doi.org/10.1080/20430795.2024.2401357. [Online]. Available: https://doi.org/10.1080/20430795.2024.2401357.

[53] G. Arévalo, M. González, A. Guzmán, and M.-A. T. and, "The value effect of sustainability: Evidence from latin american esg bond market," *Journal of Sustainable Finance & Investment*, vol. 14, no. 3, pp. 516–537, 2024. DOI: 10.1080/20430795.2024.2344527. eprint: https://doi.org/10.1080/20430795.2024.2344527. [Online]. Available: https://doi.org/10.1080/20430795.2024.2344527.

[54] K. A. and, "Are retail investors willing to buy green bonds? a case for japan," *Journal of Sustainable Finance & Investment*, vol. 0, no. 0, pp. 1–15, 2024. DOI: 10.1080/20430795.2024.2349723. eprint: https://doi.org/10.1080/20430795.2024.2349723. [Online]. Available: https://doi.org/10.1080/20430795.2024.2349723.

[55] J. Xu, "Ai in esg for financial institutions: An industrial survey," *arXiv preprint arXiv:2403.05541*, 2024.

[56] A. Alonso-Robisco, J. Bas, J. M. Carbo, A. de Juan, and J. M. M. and, "Where and how machine learning plays a role in climate finance research," *Journal of Sustainable Finance & Investment*, vol. 0, no. 0, pp. 1–42, 2024. DOI: 10.1080/20430795.2024.2370325. eprint: https://doi.org/10.1080/20430795.2024.2370325. [Online]. Available: https://doi.org/10.1080/20430795.2024.2370325.

[57] V. D'Amato, R. D'Ecclesia, and S. Levantesi, "Esg score prediction through random forest algorithm," *Computational Management Science*, vol. 19, no. 2, pp. 347–373, 2022.

[58] R. Hisano, D. Sornette, and T. Mizuno, "Prediction of esg compliance using a heterogeneous information network," *Journal of Big Data*, vol. 7, no. 1, p. 22, 2020.

[59] Y. Zou, "Predicting future esg performance using past corporate financial information: Application of deep neural networks," in *Proceedings of the 5th International Conference on Computer Information and Big Data Applications*, 2024, pp. 284–289.

[60] H. N. Bhandari, N. R. Pokhrel, R. Rimal, K. R. Dahal, and B. Rimal, "Implementation of deep learning models in predicting esg index volatility," *Financial Innovation*, vol. 10, no. 1, p. 75, 2024.

[61] V. D'Amato, R. D'Ecclesia, and S. Levantesi, "Fundamental ratios as predictors of esg scores: A machine learning approach," *Decisions in Economics and Finance*, vol. 44, no. 2, pp. 1087–1110, 2021.

[62] M. A. F. Chowdhury, M. Abdullah, M. A. K. Azad, Z. Sulong, and M. N. Islam, "Environmental, social and governance (esg) rating prediction using machine learning approaches," *Annals of Operations Research*, pp. 1–25, 2023.

[63] T. Guo, N. Jamet, V. Betrix, L.-A. Piquet, and E. Hauptmann, "Esg2risk: A deep learning framework from esg news to stock volatility prediction," *arXiv preprint arXiv:2005.02527*, 2020.

[64] H. Lee, J. H. Kim, and H. S. Jung, "Deep-learning-based stock market prediction incorporating esg sentiment and technical indicators," *Scientific Reports*, vol. 14, no. 1, p. 10 262, 2024.

[65] F. Ghallabi, B. Souissi, A. M. Du, and S. Ali, "Esg stock markets and clean energy prices prediction: Insights from advanced machine learning," *International Review of Financial Analysis*, p. 103 889, 2024.

[66] J. Park, H. J. Na, and H. Kim, "Development of a success prediction model for crowdfunding based on machine learning reflecting esg information," *IEEE Access*, 2024.

[67] C. Li, A. R. Keeley, S. Takeda, D. Seki, and S. Managi, "Investor's esg tendency probed by pre-trained transformers," *Corporate Social Responsibility and Environmental Management*, 2024.

[68] B. Sandwidi and S. P. Mukkolakal, "Transformers-based approach for a sustainability term-based sentiment analysis (stbsa)," in *Proceedings of the Second Workshop on NLP for Positive Impact (NLP4PI)*, 2022, pp. 157–170.

[69] Reuters, *Reuters*, `https://www.reuters.com/`, 2024.

[70] Sustainalytics, *Sustainalytics*, 2022. [Online]. Available: `https://www.sustainalytics.com/`.

[71] IFRS, *Ifrs s1*, `https://www.ifrs.org/issued-standards/ifrs-sustainability-standards-navigator/ifrs-s1-general-requirements/`, 2023.

[72] N. S. Soderstrom and K. J. Sun, "Ifrs adoption and accounting quality: A review," *European accounting review*, vol. 16, no. 4, pp. 675–702, 2007.

[73] E. F. Fama and K. R. French, "A five-factor asset pricing model," *Journal of Financial Economics*, vol. 116, no. 1, pp. 1–22, 2015.

[74] P. C. Belafsky, G. N. Postma, and J. A. Koufman, "Validity and reliability of the reflux symptom index (rsi)," *Journal of voice*, vol. 16, no. 2, pp. 274–277, 2002.

[75] T. T.-L. Chong and W.-K. Ng, "Technical analysis and the london stock exchange: Testing the macd and rsi rules using the ft30," *Applied Economics Letters*, vol. 15, no. 14, pp. 1111–1114, 2008.

[76] J. Bollinger, "Using bollinger bands," *Stocks & Commodities*, vol. 10, no. 2, pp. 47–51, 1992.

[77] H. Cazaux, R. Rudd, H. Stefánsson, S. Ólafsson, M. Raberto, and E. I. Ásgeirsson, "Correlation study between returns and esg ratings.," *Journal of Impact & ESG Investing*, vol. 5, no. 1, 2024.

[78] H. Cazaux, R. Rudd, H. Stefánsson, S. Ólafsson, and E. I. Ásgeirsson, "Non-stationary inverted transformer with time2vec embedding," *In Review at IEEE Transactions on Artificial Intelligence*, vol. 0, no. 0, 2025.

[79] S. M. Kazemi, R. Goel, S. Eghbali, *et al.*, "Time2vec: Learning a vector representation of time," *arXiv preprint arXiv:1907.05321*, 2019.

[80] Y. Wang, H. Wu, J. Dong, Y. Liu, M. Long, and J. Wang, "Deep time series models: A comprehensive survey and benchmark," 2024.

[81] Y. Liu, T. Hu, H. Zhang, *et al.*, "Itransformer: Inverted transformers are effective for time series forecasting," *arXiv preprint arXiv:2310.06625*, 2023.

[82] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," *arXiv preprint arXiv:2210.02186*, 2022.

[83] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" In *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, 2023, pp. 11 121–11 128.

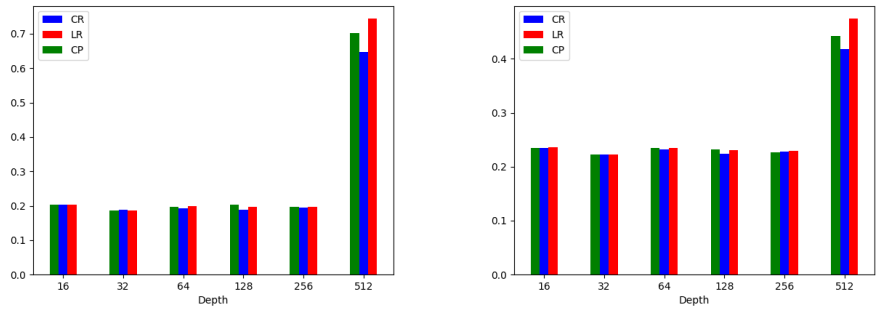
[84] H. Chefer, S. Gur, and L. Wolf, "Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 397–406.

[85] H. Cazaux, R. Rudd, H. Stefánsson, S. Ólafsson, and E. I. Ásgeirsson, "Inverted transformers interpretability beyond attention visualization," in *International Joint Conference on Neural Networks*, 2025.

[86] J. M. Cebrian, L. Natvig, and M. Jahre, "Scalability analysis of avx-512 extensions," *The Journal of supercomputing*, vol. 76, no. 3, pp. 2082–2097, 2020.

Appendix A. Depth of the network

We trained the model with progressively deeper networks, as shown in Figure A1a and A1b on AAPL. We use depths of the power of 2 to fully take advantage of bit-wise operations and AVX-512 computing [86]. We want to use the depths that yields the lowest MSE and MAE while also minimizing the computation overhead. Depths from 16 to 256 have similar results, with a small gain in performance at 32 hidden layers. At 512, the model massively overfits and the MSE/MAE increase dramatically. Consequently, the results of this preliminary depth tuning indicate that the optimal dimension of the model is 32, with a good compromise between fitting the model and reasonable computational overhead.



(a) MSE (b) MAE

Figure A1.: MSE/MAE per depth and per predicted variable on AAPL (Lower is better).

Appendix B. Predictive Performance on ESG-Enhanced Time Series

This section evaluates the performance of the NSiTransformer compared to the benchmark models on the financial dataset enhanced with data from Reuters and Sustainalytics.

B.1. *Sustainalytics*

Table B1 shows how adding Sustainalytics ESG data affects forecasting performance across financial models. While including ESG ratings shrinks the dataset size, the results reveal clear patterns in how different models handle this expanded set of features across three sectors—Finance, Industrials, and Technology.

The NSiTransformer performs strongest in the Finance sector, with MSE scores of 0.253 ($P = 1$), 0.384 ($P = 7$), and 0.512 ($P = 14$). Both iTransformer and PatchTST deliver nearly identical results, while TimesNet and DLinear show slightly higher er-

rors. The Stationary model keeps pace on short-term forecasts but loses accuracy as predictions extend beyond a week.

All models show higher error rates on the Industrials sector compared to Finance. The NSiTransformer achieves MSEs of 1.18 ($P = 1$), 1.47 ($P = 7$), and 1.86 ($P = 14$). While the NSiTransformer, iTransformer, PatchTST and TimesNet cluster closely in performance, the Stationary model struggles—its MSE jumps to 3.90 for $P = 14$ forecasts. The DLinear model places itself between the leading pack and Stationary, getting close to the top performer in Finace at $P = 1$ but never beating out any of the top 4 models. This pattern remains in the Technology sector, that appears to be simpler to predict at $P = 1$ but sharply increasing in difficulty as P rises.

Incorporating Sustainalytics ESG data creates a mixed picture: The initial top models like NSiTransformer, iTransformer, PatchTST and TimesNet maintain strong performance despite a smaller datasets, while DLinear and Stationary suffer, especially at higher P. These results indicate a predominant problem in using ESG ratings: depending on the provider, the history of data available might be more beneficial than the added dimensions.

Table B1.: Full results for the long-term forecasting task. The input sequence length is set to 96 for all baselines, and T is the prediction length. MSE stands for Mean Squared Error and MAE for Mean Absolute Error.

Models	T	NSiTransformer		iTransformer		PatchTST		TimesNet		DLinear		Stationary	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Finance	1	0.253	0.158	0.268	0.158	0.272	0.157	0.267	0.186	0.303	0.200	0.260	0.175
	7	0.384	0.245	0.382	0.245	0.405	0.240	0.404	0.271	0.429	0.274	0.435	0.290
	14	0.512	0.312	0.526	0.324	0.524	0.333	0.549	0.346	0.580	0.382	0.610	0.377
Industrials	1	1.18	0.185	1.21	0.185	1.13	0.180	1.21	0.216	1.28	0.230	1.56	0.213
	7	1.47	0.281	1.47	0.294	1.40	0.277	1.56	0.332	1.62	0.355	1.73	0.392
	14	1.86	0.375	1.98	0.375	1.76	0.349	2.06	0.400	2.22	0.505	3.90	0.496
Tech (No SMCI)	1	0.427	0.194	0.399	0.187	0.398	0.183	0.566	0.220	0.629	0.257	1.021	0.240
	7	1.240	0.306	1.222	0.306	1.196	0.296	1.600	0.360	1.544	0.382	1.989	0.350
	14	2.010	0.398	2.234	0.405	2.016	0.384	3.018	0.482	2.505	0.451	5.610	0.610

B.2. *Reuters*

Table B2 compares long-term forecasting performance using Reuters ESG ratings alongside financial data. Unlike the earlier Sustainalytics analysis, Reuters' ESG integration causes less severe data reduction—this smaller trimming of the dataset helps preserve more training examples while still adding sustainability metrics.

In Finance, NSiTransformer, iTransformer, and PatchTST dominate short-term predictions: their 1-day forecasts show nearly identical MSE scores (0.151–0.152) and MAE values (0.148–0.153). TimesNet performs slightly worse, only beating the NSiTransformer on Industrials at $P = 7$ and $P = 14$. While DLinear trails with a 0.265 MSE at this horizon, the Stationary model holds its own initially. Predictably, errors grow for all models as forecasts stretch to 7 and 14 days.

The Industrials sector tells a similar story. All models start strong with 1-day MSEs between 0.159–0.181. As predictions extend to $P = 7$, the Stationary model surprisingly closes the gap slightly at the $P = 14$, despite the previous trend trend of massive compounding error in longer time predictions.

In Technology, the first four leading models achieve $P = 1$ MSEs of 0.201–0.233, matching the Stationary model's short-term performance. But beyond a week, their lead widens dramatically: by the $P = 14$ horizon, the top models clearly outpace both DLinear and Stationary models. This divergence implies these newer architectures handle Technology inherent instability better.

Reuters' ESG integration appears to support steadier performance across sectors and timeframes compared to Sustainalytics. The larger quantity of available training data is a definite benefiting factor, which also makes the results between the two models incomparable in this state since the test segments differ.

Table B2.: Full results for the long-term forecasting task. The input sequence length is set to 96 for all baselines, and T is the prediction length. (All numbers are rounded to 3 s.f.) MSE stands for Mean Squared Error and MAE for Mean Absolute Error.

Models	P	NSiTransformer		iTransformer		PatchTST		TimesNet		DLinear		Stationary	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Finance	1	0.151	0.148	0.151	0.153	0.152	0.153	0.163	0.168	0.265	0.233	0.149	0.154
	7	0.458	0.780	0.439	0.274	0.425	0.267	0.519	0.302	0.655	0.392	0.532	0.303
	14	0.780	0.365	0.779	0.365	0.773	0.361	0.918	0.410	0.995	0.437	0.959	0.409
Industrials	1	0.168	0.168	0.162	0.161	0.159	0.159	0.181	0.187	0.291	0.242	0.171	0.168
	7	0.506	0.305	0.481	0.300	0.455	0.290	0.484	0.303	0.657	0.371	0.566	0.313
	14	0.895	0.411	0.813	0.403	0.811	0.392	0.878	0.419	1.03	0.479	0.927	0.421
Tech (No SMCI)	1	0.201	0.179	0.202	0.181	0.212	0.191	0.233	0.202	0.510	0.297	0.226	0.191
	7	1.004	0.369	0.908	0.355	0.911	0.355	0.943	0.363	1.403	0.453	1.212	0.372
	14	1.69	0.482	1.663	0.485	1.732	0.480	1.762	0.497	2.206	0.563	2.732	0.594

Appendix C. Correspondence Between Tokens And Variables

In the next sections, we will be referring to the tokens used by the model to encode the features of the dataset. This section serves as a reference for the subsequent figures in subsection 6 and Appendix D. Table C1 describes the relationship between tokens and features. This dataset serves as the base for the ESG-enhanced dataset, which all use the same first 14 tokens that encode the financial features. Tokens 15 to 26 are used to encode the temporal context of the time series.

Table C1.: Description of tokens for the encoded network without ESG ratings.

Token	**Feature**	**Description**
0	Open	Opening price
1	Low	Lowest price of the period
2	High	Highest price of the period
3	Close	Closing price
4	Volume	Trading volume
5	Log_Returns	Logarithmic returns
6	Controlled_Returns	Returns with control adjustments **(Target variable)**
7	RSI	Relative Strength Index
8	MACD	Moving Average Convergence Divergence
9	MACDs	MACD signal line
10	MACDh	MACD histogram
11	BBL_5_2.0	Lower Bollinger Band (5-day, 2.0 std)
12	BBM_5_2.0	Middle Bollinger Band (5-day, 2.0 std)
13	BBU_5_2.0	Upper Bollinger Band (5-day, 2.0 std)
14	Ticker	Stock ticker one-hot encoded
15	dayWeek	Day of the week
16	dayMonth	Day of the month
17	dayYear	Day of the year
18 - 25	date	Time2Vec Embedded Features

Table C2 presents the tokens corresponding to the encoding of the Sustainalytics ESG ratings. Due to the Time2Vec embedding happening after the dataset features, the last 11 tokens always are the temporal tokens.

Table C2.: Description of tokens for the encoded network with Sustainalytics ESG ratings.

Token	Feature	Description
0 - 14	Financial features	Same features as Table C1
15	ESG Risk score	ESG risk score
16	Overall Management Score	Overall management score
17	Overall Exposure Score	Overall exposure score
18	Overall Manageable Risk Score	Overall manageable risk score
19	Overall Unmanageable Risk Score	Overall unmanageable risk score
20	Overall Managed Risk Score	Overall managed risk score
21	SASB	SASB rating
22	dayWeek	Day of the week
23	dayMonth	Day of the month
24	dayYear	Day of the year
25 - 32	date	Time2Vec Embedded Feature

Table C3 presents the tokens corresponding to the encoding of the Reuters ESG ratings. The Reuters scores correspond more directly to the idea of ESG ratings by directly integrating the values as pillar score.

Table C3.: Description of tokens for the encoded network with Reuters ESG ratings.

Token	Feature	Description
0 - 14	Financial features	Same features as Table C1
15	ESG Score	Overall ESG score
16	ESG Combined Score	Combined ESG score
17	ESG Controversies Score	ESG controversies score
18	Social Pillar Score	Social pillar score
19	Governance Pillar Score	Governance pillar score
20	Environmental Pillar Score	Environmental pillar score
21	SASB	SASB rating
22	dayWeek	Day of the week
23	dayMonth	Day of the month
24	dayYear	Day of the year
25 - 32	date	Time2Vec Embedded Features

Table C4 presents the tokens corresponding to the encoding of the ESG ratings from both providers. This experiment alongside the relevance maps and de-stationary factors aims at determining the influence of each token in the prediction.

Table C4.: Description of tokens for the encoded network with both providers of ESG ratings.

Token	Feature	Description
0 - 14	Financial features	Same features as Table C1
15	ESG Risk score	ESG risk score
16	Overall Management Score	Overall management score
17	Overall Exposure Score	Overall exposure score
18	Overall Manageable Risk Score	Overall manageable risk score
19	Overall Unmanageable Risk Score	Overall unmanageable risk score
20	Overall Managed Risk Score	Overall managed risk score
21	ESG Score	Overall ESG score
22	ESG Combined Score	Combined ESG score
23	ESG Controversies Score	ESG controversies score
24	Social Pillar Score	Social pillar score
25	Governance Pillar Score	Governance pillar score
26	Environmental Pillar Score	Environmental pillar score
27	SASB	SASB rating
28	dayWeek	Day of the week
29	dayMonth	Day of the month
30	dayYear	Day of the year
31 - 38	date	Time2Vec Embedded Feature

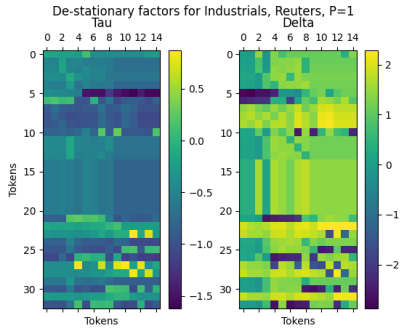
Appendix D. De-stationary Factors

We sample the tensors of τ and δ during testing and represent it as heatmaps. The de-stationary factors represent the evolving relationship between the features. We used a top-k of 15, meaning that de-stationary attention is applied to the top 15 features. This visualization allows us to gain insights on how the de-stationary factors from the NSiTransformer are used within the model to represent the changing relationships between the features.

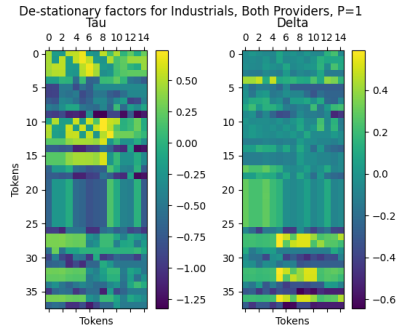
Figure D1 presents the de-stationary factors for the model trained on Industrials at $P = 1$, with both providers and only Reuters. Horizontally, the tokens are ordered according to the tables in subsection C. Vertically, the tokens are the top 15 with the highest attention at the time of the sampling. In Reuters, a clear horizontal pattern emerges alongside tokens 13 to 20, with both tau and delta being equal for those tokens regarding the top 15 tokens. A possible explanation for this result is the low periodicity of these tokens, which also often change at the same timestamps when the ratings are updated. This idea is further backed up by the S + R model, which demonstrates the same behavior with Sustainalytics and Reuters tokens. Another observation is the relationship between the de-stationary factors and the relevance maps. It appears that the tokens that are highly relevant in previous figures often present lower de-stationary factors. For instance, on the S+P model, token 5 is by far the most relevant, but has some of the lowest tau and delta. Token 9 and 27 also show the same behavior, while the ticker (token 14) displays the opposite: extremely low relevance but high de-stationary factors.

Figure D2 presents the de-stationary factors for the model trained on Finance at $P = 7$, and the inverted relationship between relevance and de-stationary factors persists. In the S+R model in particular, we can clearly see the groups of tokens 0 to 4 with low relevancy and high factors, as opposed to tokens 5 to 9 with high relevancy and low factors. There are however counter examples, for instance with token 9 in No ESG maintaining both high relevancy and high factors. A likely explanation is that since the de-stationary factors are learned, the model could be compensating high relevancy with low factors to balance out the prediction and avoid overfitting.

Visualizing the de-stationary factors allows to gain insight about the model and how it tries to balance the evolving relationship between features. While the vertical lecture of the maps are imperfect when using a k inferior to the number of features, the horizontal lecture still gives insight as to which token are given increased or decreased attention at a given time. In this case, the tendency of de-stationary factors to counterbalance the relevance and the low periodicity of data having an influence on their attention are important signs that the model can generalize well on unseen data.

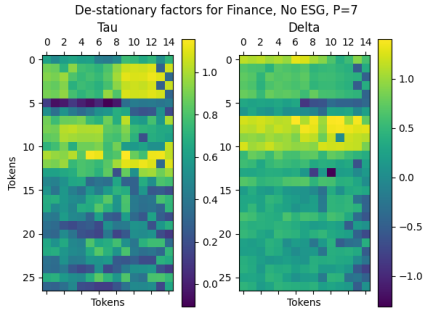


(a) Reuters

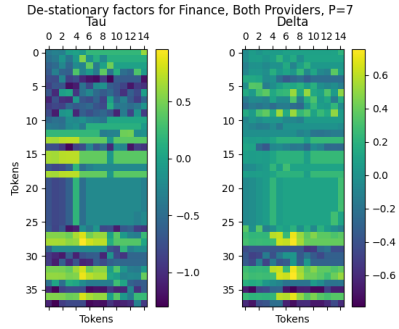


(b) S + R

Figure D1.: De-stationary factors for Industrials, P=1, Reuters (left), S + R (right).



(a) No ESG



(b) S + R

Figure D2.: De-stationary factors for Finance, P=7, No ESG (left), S + R (right).

Appendix F

Article 6 - Fine-tuning Timeseries Predictors Using Reinforcement Learning [44]

CHAPTER TEMPLATE

Fine-tuning Timeseries Predictors Using Reinforcement Learning

Hugo Cazaux[a,b] , Ralph Rudd[a], Hlynur Stefánsson[a], Sverrir Ólafsson[a] and Eyjólfur Ingi Ásgeirsson[a]

[a]Reykjavik University, Department of Engineering, Menntavegur 1, Reykjavik, 102, Iceland
[b]Corresponding author, email: hugot@ru.is

ARTICLE HISTORY
Compiled June 4, 2025

ABSTRACT
This chapter presents three major reinforcement learning algorithms used for fine-tuning financial forecasters. We propose a clear implementation plan for backpropagating the loss of a reinforcement learning task to a model trained using supervised learning, and compare the performance before and after the fine-tuning. We find an increase in performance after fine-tuning, and transfer learning properties to the models, indicating the benefits of fine-tuning. We also highlight the tuning process and empirical results for future implementation by practitioners.

KEYWORDS
Fine-tuning, Proximal Policy Optimization, Reinforcement Learning, Attention

1. Introduction

Timeseries predictors are generally trained using supervised learning on datasets. The standard setup divides the dataset into three segments: training, validation and testing. The model is initially fit on training data, then evaluated on the validation set to tune hyper-parameters and assess the predictive power. Finally, the test set is used by the final model to determine the accuracy on unseen data. These steps are well understood and constitute the backbone of supervised learning in timeseries prediction.

This methodology draws a strong parallel with large language models (LLMs), which are generally transformer-based models and use supervised learning for pre-training. The pre-training is a common step to all LLMs that starts with a large amount of raw data, compressed in the network. Once the pre-training is complete, alignment aims to tune the model to create an user-friendly experience. This step incentivizes answering and asking questions to contextualize requests, teaches the LLMs how to use external tools, or censors potentially harmful information that might lie within the embeddings. The novelty of this research lies in the extension of alignment to time-series prediction model.

Fine-tuning in large language models was initially based on human feedback. A standard setup consists of a human operator prompting a question and grading the answer. Later, practitioners aimed at removing subjectivity from the fine-tuning pipeline by instead proposing two answers to a prompt and have a human operator select the best one. These two methods fall under the umbrella of Reinforcement Learning with Human Feedback (RLHF), and although effective recent models have shown pure Re-

inforcement Learning (RL) approaches outperforming RLHF for a fraction of the cost.

The central idea of this chapter is to leverage the predictive power of supervised pre-training and to use RL algorithms to align the model with diverse constraints. These constraints can be domain specific, such as risk management or operational constraints, but can also be purely mathematical, such as incentivizing bolder out-of-sample predictions. The reward function is at the center of the tuning, and will determine which direction the model is pushed towards. This approach is more adapted to time-series prediction compared to RLHF, as it completely removes the human operator and the need to reduce subjectivity in the feedback. RL is also extremely cost effective, since it removes the need of a coordinated effort of human operators giving feedback on a large number of samples.

In the context of time-series prediction, RL for fine-tuning is novel. The standard implementations of reinforcement learning in time-series prediction consist of a completely untrained agent learning a policy over a simulated environment. In the case of finance, this environment might be a portfolio or an ensemble of assets. In this chapter, we used a pre-trained model from [1] that serves as a backbone for the RL implementation. The environment is set up to reflect the training data closely, with the main tuning tool available being the reward structure. The loss is back-propagated through the backbone, updating the weights according to the policy.

The research questions investigated in this chapter are: Can we fine-tune pre-trained models to enhance time-series predictions using reinforcement learning? What state-of-the-art reinforcement algorithms work best for fine-tuning?

This chapter is structured as follow: Section 2 presents a literature review, section 3 the data used to train/test the models, Section 4 the framework used to fine-tune and evaluate the models, Section 5 benchmarks the models on standard reinforcement learning tasks, Section 6 the results of the fine-tuning, Section 7 the tuning of the specific hyperparameters and finally Section 8 is the conclusion to the chapter.

2. Background

Fine-tuning has become an emerging trend since large pre-trained model became more widely available to the public [2]. Fine-tuning is a technique that intends to specialize a pre-trained backbone model, often to increase performance on selected benchmarks [3] or to benefit from previously acquired knowledge through transfer learning [4]. The democratization of open source models with available weights in natural language processing [5], [6] and image processing [7] enabled researchers and enthusiasts to propose their own fine-tuned version of an advanced model without the high computational cost of pre-training. Fine-tuning was leveraged to propose fine art classification [8], fine-tuning large language models for better medical care [9], biomedical tasks in different languages [10], and malware detection in images [11].

As the size of models and the parameter number grow exponentially, fine-tuning the entire model for each downstream tasks was replaced with a sparser approach called parameter-efficient fine-tuning [12], [13]. Methods such as Adapter [14], [15], LoRA [16] and Prefix-tuning [17] propose to modify the architecture of the original model to benefit from higher order patterns learned during supervised learning while also specializing in a downstream task. Supervised fine-tuning uses labeled data after pre-training to align the model towards a downstream task. This method has grown in popularity as large language models hit the public sphere and adapted for more intuitive or safer usage [18], [19], [20].

As the cost of computation carried over to efficient data labeling [21], alternative techniques for fine-tuning were explored. Reinforcement learning, one of the major paradigms in machine learning, has become one of the prime candidate for efficient fine-tuning. Adversarial networks had previously shown promising results [22], and policy learning has been employed in text-to-image [23] and multi-modal models [24]. Perhaps the most impressive implementation of reinforcement learning based fine-tuning comes from the DeepSeek-v3 report [25], which implements group proximal policy optimization to fine-tune a pre-trained model and implement chain-of-thoughts reasoning.

Within time-series prediction, fine-tuning has been focused on domain adaption. In a similar fashion to text and image generation, large pre-trained models are becoming available to researchers [26]. The models can then be fine-tuned for domain specific predictions and receive the same benefit as large language models [27], [28]. However, these methods involve supervised fine-tuning, which in the case of time-series prediction consists of adding data form the specific domain the model needs to be fine-tuned on. As large language models have proven in the past, this method of fine-tuning can quickly become unsustainable due to the increasing cost of data labeling. In this study, we follow the way paved by LLMs by proposing reinforcement learning to tune time-series predictors.

PPO is a policy gradient method developed by John Schulman et al. in 2017 [29]. The key innovation of this algorithm over older methods such as TRPO [30] or ACER [31] is the clip function that constrains policy updates of the agent. PPO has been used in a wide variety of applications: Atari games [32], track racing games [33], suspension monitoring for cars [34], and image captioning [35]. A number of articles have proposed innovations to the base algorithm, for instance an alternative minimization target [36], [37] introduced policy feedback; specifically improving early learning stages, which are recognized as a potential weak point of PPO [38]. Recently proposed improvements include a shift in learning to offline policy optimization [39] and including conservatism [40].

Multi-agent methods have gained significant attention in the field of reinforcement learning, particularly for their capability to simulate complex systems involving interactive agents. A notable early work in multi-agent systems is [41] which explored the dynamics of cooperative and competitive agents in a shared environment. Recent advancements have integrated PPO into multi-agent applications: [42] applied multi-agent PPO to competitive and cooperative tasks, [43] successfully employed multi-agent reinforcement learning in the complex environment of the Dota 2 game. The integration of PPO into multi-agent systems has also been explored in real-world scenarios such as traffic light control [44], and collaborative robotics [45]. Innovations specific to multi-agent PPO include [46] which introduced a meta-learning approach to enhance adaptability across different tasks and agent configurations and [47], which presented the concept of leniency in multi-agent learning, mitigating the non-stationary issue commonly faced in such environments.

Attention is a machine learning mechanism designed to imitate human awareness. Attention was brought to the forefront of the field with the transformer architecture, a self-attention-based architecture that enabled the recent breakthroughs in large language models [48]. It has since seen many implementations including in recurrent neural networks for search results customization [49], missing data imputation [50], and in computer vision [51]. In reinforcement learning, attention models have been developed within theoretical frameworks [52] and diverse applications such as source code summarizing [53], dynamic graph problems [54], and road networks management

[55].

The novelty of the framework presented lies in the combination of staple reinforcement learning models with time-series predictors. This chapter also creates an opportunity for further applications of the framework in simulated environment encompassing diverse fields.

3. Data

To contextualize the fine-tuning we detail the financial datasets used to train the backbone and to build the fine-tuning environment. We also present the MuJoCo framework, which we use to benchmark pure reinforcement learning performance between algorithms.

3.1. *Financial and ESG Data*

The financial and ESG data used in this chapter span from intraday market prices to annual sustainability ratings. Our primary sources are:

- **Refinitiv** [56]: a global leader in financial data and analytics, covering over 80% of global market capitalization with more than 450 ESG metrics. We extract daily price and volume data via Refinitiv Eikon, together with the three ESG pillar scores (Environmental, Social, Governance) and the combined ESG score.
- **Sustainalytics** [57]: provides ESG Risk Ratings for listed firms, widely used by asset managers and banks to construct sustainable portfolios. We incorporate their flagship ESG Risk Ratings into our dataset.
- **SASB Standards** [58], [59]: the Sustainability Accounting Standards Board identifies material sustainability issues by industry. Since August 2022, SASB standards underlie IFRS S1 and S2 disclosures. We one-hot encode each firm's material SASB issue set based on the 2018 publication.

Table 1 shows a snippet of Apple's daily price data from 2005-12-05 to 2005-12-13. The full time span of the dataset is 2005-12-05 through 2024-08-07.

Date	Open	Low	High	Close	Volume
2005-12-05	2.17	2.15	2.19	2.16	5.84e8
2005-12-06	2.23	2.21	2.25	2.23	8.57e8
2005-12-07	2.24	2.20	2.24	2.23	6.79e8
2005-12-08	2.21	2.19	2.23	2.23	7.90e8
2005-12-09	2.24	2.21	2.25	2.24	5.55e8
2005-12-12	2.26	2.25	2.27	2.26	5.25e8
2005-12-13	2.25	2.24	2.27	2.26	4.94e8

Table 1.: Sample daily financial data for AAPL

To enrich the raw price and volume data, we compute:

- *Log returns*, controlling for market effects via the Fama–French 5 factors [60].
- Technical indicators from historical prices and volumes:
  - Relative Strength Index (RSI) [61],
  - Moving Average Convergence Divergence (MACD) [62],

- Bollinger Bands [63].

The target variable is the FF5-adjusted log return, following the methodology of [64]. Financial data are available at sub-daily frequency, whereas ESG scores refresh annually (Refinitiv) or "regularly" (Sustainalytics). We evaluated regression, interpolation, autoencoders and forward-fill strategies. To respect provider methodologies and avoid compounding model error, we adopt a forward-fill approach for ESG values between update dates.

3.2. *MuJoCo Benchmarking Environments*

Multi-Joint dynamics with Contact, commonly called MuJoCo [65], proposes several standard environments to train and benchmark models on. To evaluate pure reinforcement learning performance, we employ three standard MuJoCo tasks:

- **HalfCheetah-v4**,
- **Hopper-v4**,
- **Humanoid-v4**.

MuJoCo provides a high-fidelity physics simulator for continuous-control benchmarks, where:

- *State* $s_t \in \mathbb{R}^d$ consists of joint angles, velocities and (for Humanoid) contact forces.
- *Action* $a_t \in \mathbb{R}^m$ represents torque inputs to each joint.
- *Reward* combines forward progress, control costs, and (where applicable) healthy posture and contact penalties.

Environment	**Reward**
HalfCheetah-v4	$R = w_f F - w_{\text{ctrl}}\, C$
Hopper-v4	$R = w_f F + w_h H - w_{\text{ctrl}}\, C$
Humanoid-v4	$R = w_f F + w_h H - w_{\text{ctrl}}\, C - w_{\text{ctct}}\, C_{\text{tct}}$

Table 2.: MuJoCo environment reward functions (forward reward F, healthy reward H, control cost C, contact cost C_{tct})

Here, $w_f, w_h, w_{\text{ctrl}}, w_{\text{ctct}}$ are environment-specific weights. We use the default observation and action spaces as defined in OpenAI Gym's MuJoCo suite.

4. Framework Details

As mentioned in [66], implementation is key in deep policy gradient algorithms. As such, the framework below is implemented using the clean-rl library [67]. We evaluate three state-of-the-art algorithms for fine-tuning: Proximal Policy Optimization (PPO), Centralized Multi-Agent PPO (CMAPPO), and Group Relative Policy Optimization (GRPO). In this section, we also detail the environment used during training and the integration of the pre-trained transformer in the algorithms.

4.1. *Proximal Policy Optimization (PPO)*

- **Policy Function:** For an agent x, its policy at time t is a probability density function denoted as $\pi_\theta(a_t|o_t)$, where θ are the parameters of the policy, o_t is the observation for agent x at time t, and a_t are the actions that can be taken. The policy is then sampled to obtain the action taken $\alpha_t \sim \pi_\theta(a_t|o_t)$.
- **Objective Function:** The PPO objective function is defined as:

$$L^{PPO}(\theta) = \mathbb{E}_t \left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t) \right]$$

  where $r_t(\theta) = \frac{\pi_\theta(a_t|o_t)}{\pi_{\theta_{\text{old}}}(a_t|o_t)}$ is the probability ratio, ϵ an hyperparameter and $\hat{A}_t$ is an estimator of the advantage at time t, typically computed using Generalized Advantage Estimation (GAE).
- Advantage Estimation: The advantage $\hat{A}_t$ is computed as:

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \ldots + (\gamma\lambda)^{T-t+1}\delta_{T-1} \tag{1}$$

  with $\delta_t = r_t + \gamma V(o_{t+1}) - V(o_t)$ and V a learned state-value function.
- **Training Process:** The agent is trained by iteratively updating its policy parameters. This involves:
  (1) Collecting trajectories by interacting with the environment using the current policy.
  (2) Estimating the advantages using GAE.
  (3) Calculating the surrogate objective function.
  (4) Optimizing the surrogate objective function using gradient ascent while ensuring the updates stay within a specified clipping range to maintain policy stability.

4.2. *Centralized Multi-Agent PPO (CMAPPO)*

- **Subagent Policy & Training:** Each subagent x_i observes its local state $o_{t,i}$, samples an action $\alpha_{t,i} \sim \pi_{\theta_i}(a_{t,i} \mid o_{t,i})$, and learns via its own reward $R_i(o_t, a_{t,i})$ using PPO:
  (1) *Collect trajectories:* Interact with environment to gather $\{(o_{t,i}, \alpha_{t,i}, r_{t,i})\}_{t=1}^{T}\}$.
  (2) *Advantage estimation:* Compute $\hat{A}_{t,i}$ via GAE: $\hat{A}_{t,i} = \delta_{t,i} + (\gamma\lambda)\delta_{t+1,i} + +(\gamma\lambda)^2\delta_{t+2,i} + \cdots$, with $\delta_{t,i} = r_{t,i} + \gamma V(o_{t+1,i}) - V(o_{t,i})$.
  (3) *Surrogate objective:*

$$L_i^{\text{PPO}}(\theta_i) = \mathbb{E}_t\big[\min(r_{t,i}\hat{A}_{t,i},\ \text{clip}(r_{t,i}, 1-\epsilon, 1+\epsilon)\hat{A}_{t,i})\big],$$

  where $r_{t,i} = \frac{\pi_{\theta_i}}{\pi_{\theta_i^{\text{old}}}}$.
  (4) *Policy update:* Perform gradient ascent on L_i^{PPO}, clipping updates to maintain stability.
- **Attention-Enhanced Aggregation:** Encode the global state e_t and subagent actions $\{\alpha_{t,i}\}$ via linear layers, compute attention weights $[w_{\text{env}}, w_1, \ldots, w_n] =$

$\mathrm{softmax}([f_{\mathrm{env}}(e_t),\ f_{\mathrm{sub}}(\{\alpha_{t,i}\})])$, then aggregate:

$$d_t = w_{\mathrm{env}}\, e_t \ + \ \sum_{i=1}^{n} w_i\, \alpha_{t,i}.$$

- **Superagent Decision:** The superagent samples its final action $\alpha_t^f \sim \pi_{\theta_f}(a_t^f \mid d_t)$, allowing coordinated, adaptive decisions across all agents.

4.3. *Group Relative Policy Optimization (GRPO)*

- **Policy Function:** As in PPO, we parameterize a stochastic policy $\pi_\theta(a \mid o)$ with parameters θ. At each step t, given observation o_t, we sample a group of G candidate actions

$$a_{t,i} \sim \pi_\theta(\cdot \mid o_t), \quad i = 1, \ldots, G.$$

- **Group Rewards and Relative Advantage:** Each candidate action $a_{t,i}$ is scored by a reward function $r(a_{t,i}, o_t)$, yielding

$$r_{t,i} = r(a_{t,i}, o_t).$$

We compute the group baseline (mean) and standard deviation:

$$\bar{r}_t = \frac{1}{G}\sum_{i=1}^{G} r_{t,i}, \quad \sigma_t = \sqrt{\frac{1}{G}\sum_{i=1}^{G}\big(r_{t,i} - \bar{r}_t\big)^2} + \epsilon.$$

The *relative advantage* of candidate i is then:

$$A_{t,i} \ = \ \frac{r_{t,i} - \bar{r}t}{\sigma_t}.$$

- **Surrogate Objective:** Defining the probability ratio for each candidate,

$$\rho_{t,i}(\theta) = \frac{\pi_\theta(a_{t,i} \mid o_t)}{\pi_{\theta_{\mathrm{old}}}(a_{t,i} \mid o_t)},$$

the GRPO loss uses the same clipped surrogate as PPO but averages over the group:

$$L^{\mathrm{GRPO}}(\theta) = \mathbb{E}_t\left[\frac{1}{G}\sum_{i=1}^{G} \min\big(\rho_{t,i}(\theta)\, A_{t,i},\ \mathrm{clip}\big(\rho_{t,i}(\theta), 1-\epsilon, 1+\epsilon\big)\, A_{t,i}\big)\right].$$

Optionally, one may add a KL-penalty term $\beta\, D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid o_t) \| \pi_{\mathrm{ref}}(\cdot \mid o_t)\big)$ to constrain policy drift.

- **Training Process:** GRPO proceeds in iterative updates:
  (1) *Sample Groups:* For each observation o_t in a batch, sample G actions $\{a_{t,i}\}$.
  (2) *Evaluate Rewards:* Compute $r_{t,i} = r(a_{t,i}, o_t)$ for $i = 1 \ldots G$.

(3) *Compute Advantages:* Form relative advantages $A_{t,i} = (r_{t,i} - \bar{r}_t)/\sigma_t$.
(4) *Surrogate Update:* Optimize θ by ascending the clipped surrogate $L^{\mathrm{GRPO}}(\theta)$ (plus optional KL term), using minibatch gradient steps.
(5) *Repeat:* Collect new groups under the updated policy and continue until convergence.

4.4. *Design of the Reinforcement Learning Environment*

The RL environment is designed to facilitate the fine-tuning of forecasting policies:

- **State:** At time t, the state $s_t \in \mathbb{R}^{T \times N}$ is a matrix containing historical observations.
- **Agent Action:** The agent produces a forecast $a_{t,i} \in \mathbb{R}^{P \times 1}$ based on its local observation $o_{t,i}$.
- **Transition Dynamics:** Following the agents' actions, the true future $y_t \in \mathbb{R}^{P \times 1}$ is revealed, and the state is updated (via a sliding window mechanism).
- **Reward:** The reward r_t is computed based on the forecast error and any additional domain-specific criteria:

$$r_t = -\ell(a_t, y_t) - \psi(a_t), \tag{2}$$

where $\ell(\cdot)$ is an error metric (e.g., absolute or squared error) and $\psi(\cdot)$ encapsulates further constraints or penalties.

In practice, the reward function used was $r_t = 2 \times e^{(-MSE(a_t, y_t)} - 1$. This implementation constrains the reward between $[-1, 1]$, and is driven up as the MSE converges towards 0.

4.5. *Latent Representation versus Actor Network*

In practice, the probability distribution each of the algorithms sample from is a neural network. In a classic reinforcement learning approach, a new network is created to learn the latent representation between observations and actions (the action network). In the case of PPO and CMAPPO, networks are also created to learn the value function (the critic network). To fine-tune a pre-trained backbone model, we need to integrate the trained network in the framework. There are two main paradigms for fine-tuning the network:

- **The backbone outputs a latent representation of the observation space.** The action network takes the latent representation as input and outputs a probability distribution over actions, which when sampled outputs the forecast. The critic network estimates the state value for advantage estimation and the gradients flow back through the action network, critic network, and the backbone, which leads to fine-tuning.
- **The backbone is connected to a projection layer that converts the latent representation to a forecast directly.** This is what commonly happens when the backbone is used independently as a predictor. In this paradigm, the backbone takes the place of the actor network. The critic network estimates the state value and the gradients flow back through the backbone and the critic network.

Using a separate action network can improve the flexibility since the actor network has the opportunity to learn from the latent features. Decoupling the backbone and the action network also allows us to adjust the hyperparameters for the action network individually. An actor network is also more likely to explore and better adapt to the reward structure of the environment, performing significantly better in the reinforcement learning environment. We can also delay the fine-tuning by temporarily freezing all the backbone layers. This can be beneficial to performance as it gives the opportunity for the action and value networks to learn about the environment before inducing changes in the backbone network. This process can help avoid catastrophic forgetting during the early stages of interacting with the environment.

By replacing the actor network with the backbone, we ensure that a new actor network will not corrupt the original predictor. This approach is simpler and more direct, as the actor network introduces new hyperparameters but directly using the backbones only involves a minor projection. With no actor network involved, there is also less risk of overfitting the reinforcement learning task, thus maintaining a good degree of generalization. However, without an intermediary network to adapt the learned features, the backbone might struggle to perform and learn in the reinforcement learning environment. This can lead to repeated poor performance which in turn can flow through the gradient and cause catastrophic forgetting. The environment also needs to be carefully designed to avoid a mismatch between the observations at each step of the training and the encoder size of the backbone.

Both methods are compared in Table 3 using standard PPO. The reference scores are the scores of the backbone without any fine-tuning. The latent paradigm performs significantly worse, with only a small improvement in the Financial sector and massive loss in Industrials and Technology. The Actor paradigm improves upon the reference on all datasets. As such, we implemented the actor paradigm when possible. The only latent representation used was in CMAPPO with the superagent, as the aggregation of the subagents action does not correspond to the encoder accepted size of the backbone.

Table 3.: Latent vs Actor paradigms comparison. The backbone is fine-tuned using PPO on Financial, Industrials and Technology. Reference is the base model without fine-tuning. Lower is better, in bold the best metric.

Dataset	Latent		Actor		Reference	
Metric	MSE	MAE	MSE	MAE	MSE	MAE
Financial	0.202	0.206	**0.200**	0.271	0.203	**0.118**
Industrials	0.274	0.251	**0.119**	**0.116**	0.128	0.121
Technology	0.341	0.264	**0.126**	**0.119**	0.131	0.119

5. Benchmarking

Three MuJoCo environments were selected as experimental settings. The three environments are: Hopper-v4, Half-Cheetah-v4 and Humanoid-v4. In this experiment, we use standard 64 hidden dimensions networks for the action and value heads. Table 4 presents the results of the three algorithms tested on each MuJoCo environment. CMAPPO wins out on all three environments, followed closely by default PPO. The GRPO algorithm, which does not use a critic network, underperforms slightly in the pure reinforcement learning task, especially in the Hopper-v4 environment.

Table 4.: Results of MuJoCo environment training. Higher is better, best value in bold.

Model Environment	PPO Reward	CMAPPO Reward	GRPO Reward
HalfCheetah-v4	-150.54	**-111.10**	-137.18
Hopper-v4	1185.06	**1960.75**	624.86
Humanoid-v4	2897.81	**3201.09**	2659.32

6. Results

Fine-tuning is by definition local and its performance is measurable on a case-by-case basis. To cover as many use cases as possible, we propose to examine the results through the use of two common techniques in fine-tuning: layers freezing and transfer learning.

6.1. *Fine-tuning and Frozen Layers*

In order to retain high level patterns learned during supervised training, we can freeze parts of the model to stop the loss propagation through the network. This technique is common in large language models alignment and is employed to build the results in Table 5. We fine-tune the model with no frozen layers, 25%, 50% and 75% frozen layers.

Table 5.: Results of fine-tuning models on Financial, Industrials and Technology dataset compared to the original model. In rows, the model's layers are progressively frozen. In columns, each sector represents the testing set of the model. Lower is better, best value in bold.

Frozen %	Model	Financial		Industrials		Technology	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
0%	PPO	0.200	0.271	0.119	0.116	0.126	0.119
	CMAPPO	0.324	0.208	0.146	0.160	0.203	0.189
	GRPO	0.198	0.109	0.118	0.113	0.124	0.115
25%	PPO	0.199	0.114	0.120	0.116	0.125	0.118
	CMAPPO	0.300	0.204	0.202	0.211	0.525	0.341
	GRPO	0.198	0.108	**0.118**	**0.112**	0.124	0.113
50%	PPO	0.202	0.113	0.119	0.117	0.124	0.117
	CMAPPO	0.237	0.155	0.257	0.248	0.151	0.151
	GRPO	**0.195**	**0.108**	**0.118**	**0.112**	0.124	0.113
75%	PPO	0.200	0.114	0.119	0.117	0.124	0.117
	CMAPPO	0.270	0.183	0.289	0.272	0.137	0.135
	GRPO	**0.195**	**0.109**	0.118	0.113	**0.123**	**0.113**
Original	Backbone	0.202	0.111	0.120	0.115	0.124	0.116

GRPO performed the best overall, either improving or leaving the backbone model unchanged. Notably, freezing at least 50% of the encoder layers gave consistently the best performance when using GRPO. PPO proposed a minor improvement in some categories, for instance in Financial at 25%, but mostly left the model unchanged. CMAPPO performed the worst in the fine tuning, provoking large negative changes

to the model even with 75% of the encoder frozen. The source of the performance of GRPO in fine-tuning is the same reason it was the worst performer in the pure reinforcement learning task: the absence of a value function. While this is mostly a disadvantage learning control tasks, in the case of fine-tuning the difference of complexity between the value network and the backbone severely hinders the performance of PPO and CMAPPO. In the case of CMAPPO, the latent representation offered by the subagents are also reconciled using an action network. This design is coherent with the original implementation of CMAPPO but also adds another layer of abstraction the model needs to learn. A possible improvement for PPO and CMAPPO would be to run the model without propagating the loss back to the backbone to train the value network. By delaying the learning, the value network could learn a proper representation of the advantage in the task and nudge the backbone in the right direction.

6.2. *Transfer Learning*

Transfer learning is a machine learning technique through which a model learns general concepts applicable across multiple datasets. We experiment on transfer learning by fine-tuning and testing the model on the three datasets.

Table 6.: Reference values before fine-tuning.

Trained on	Financial		Industrials		Technology	
Tested on	MSE	MAE	MSE	MAE	MSE	MAE
Financial	**0.203**	0.118	0.207	0.113	0.203	**0.110**
Industrials	0.224	0.227	0.128	0.121	**0.122**	**0.114**
Technology	0.256	0.229	0.132	0.118	**0.131**	**0.117**

Table 6 presents the results of the model on the Finance, Industrials and Technology datasets before fine-tuning. Instead of training the backbone model on all three datasets and fine-tuning for one, we train the backbone on a single dataset and test the MSE/MAE on all three. The Financial appears as the most challenging dataset, performing quite worse than the baseline when tested on Industrials and Technology. The model trained on Industrials manages to nearly match the performance of the models trained on Financial and Technology. Finally, the Technology model is by far the best, outperforming Industrials even when tested on Industrials. This metric could be interpreted as the degree of high level patterns present in the dataset. These high level patterns can be applied to any similar dataset, and ultimately are more powerful predictive tools than the past history for a given example.

Table 7.: Results of fine-tuning models on Financial, Industrials and Technology dataset compared to the original model. The model is fine-tuned and tested on the specified sector for each row. In columns, each sector represents the original training set of the model. Lower is better, best value in bold.

Trained on	Model	Financial		Industrials		Technology	
Fine-tuned on	Metric	MSE	MAE	MSE	MAE	MSE	MAE
Financial	PPO	0.279	**0.196**	**0.213**	0.219	0.247	0.219
	CMAPPO	0.224	**0.143**	**0.145**	0.152	0.151	0.151
	GRPO	0.230	**0.156**	**0.155**	0.167	0.170	0.165
Industrials	PPO	0.201	0.**115**	**0.123**	0.119	0.131	0.121
	CMAPPO	0.204	**0.117**	**0.127**	0.121	0.137	0.125
	GRPO	0.197	**0.110**	**0.119**	0.113	0.125	0.114
Technology	PPO	0.198	**0.114**	**0.125**	0.117	0.127	0.120
	CMAPPO	0.202	**0.115**	**0.128**	0.119	0.129	0.120
	GRPO	0.189	**0.108**	**0.118**	0.112	0.123	0.113

Table 7 presents the results of the model on the Finance, Industrials and Technology datasets after fine-tuning. A first observation is the improvement in performance in all nearly all models from the baseline in Table 6. Some of the most substantial gains are found in the model trained on Financial, which improved its performance in MSE for both Industrials and Technology but moreover completely dominates the MAE benchmark. On the MSE front, the model trained on Industrials had the best results and beat out the best reference values for each sector.

Notable exceptions are the model trained and fine-tuned on Financial, and the model trained on Technology and fine-tuned on Industrials. In both cases, neither PPO, CMAPPO or GRPO managed to improve the performance, and testing on unseen data yielded a worse result. For the first case, the likely explanation is an overfitting to the train data: effectively, the model was trained twice on the same dataset, once with supervised learning, and again using reinforcement learning. The second case is different: the original Technology model already performed outstandingly well in Industrials, beating out even the models trained on the complete dataset. The fine-tuning failed to further improve that performance, marking the importance of establishing baselines before introducing fine-tuning to the pipeline.

The patterns noticed in Table 5 largely stand, with GRPO clearly distinguishing as the better option in nearly all cases. CCMAPPO performed exceptionally well on Financial, outperforming both PPO and GRPO. The superagent managed to reconcile the actions of the subagents despite the added complexity of the actor and critic network. PPO nearly always improves on the baseline and constitute a valid choice for fine-tuning. The recommended algorithm stands out as GRPO, which uses fewer computational resources and yields the best performance. Committing to the actor paradigm and removing the critic network greatly simplifies the fine-tuning architecture, allowing for direct backpropagation through the backbone without the need for intermediary networks.

These results also clearly indicate the value of transfer learning for timeseries predictor. One of the best use case for fine-tuning appears to be adapting models from their supervised training dataset to another. This is in line with the current state of fine-tuning in large language models, which often adapts model after pre-training to diverse specific tasks. This result also highlights two clear areas for improvement in timeseries predictors: firstly, large pre-trained models can be built, and later special-

ized to a given dataset. But the biggest challenge to generalize this method is to specify a model and a fine-tuning environment that allows for various observation space and exogenous features.

7. Key hyperparameters

PPO and its variants are known to be sensitive to hyperparameters. In order to compare each algorithm fairly, we show in this section specific and non-specific hyperparameters tuning. All models presented from this point onward use the backbone as the action network, and a value network with 2 layers and 256 hidden dimensions when relevant (PPO, CMAPPO).

7.1. *Training time*

Training time is common to PPO, CMAPPO and GRPO. A higher number of timesteps will lead to a better performance in the environment until the agent reaches a plateau, at the expense of a higher computational cost. We fine-tune the model on the Financial dataset using PPO at different timesteps and plot the MSE over time in Figure 1. We found 500 000 timesteps to be the best value as a balance between overfitting and underfitting.

Figure 1.: Training time vs MSE. Dotted line is the original model performance before fine-tuning. Training time is scaled down from 1e6 for readability.

7.2. *Number of subagents (CMAPPO)*

Number of subagents is specific to CMAPPO and controls how many subagents are trained before the superagent. We fine-tune a predictive model on the Financial dataset with an increasing number of subagents and test the MSE/MAE after fine-tuning. Table 8 presents the MSE and MAE with increasing numbers of subagents and compared to the backbone model. We found 10 subagents to be the best configuration, despite the backbone outperforming the fine-tuned model in all configurations.

Table 8.: The influence of the number of subagents when fine-tuning the backbone compared to the non fine-tuned backbone.

Number of Subagents	Financial	
Metric	MSE	MAE
2	0.925	0.401
4	0.461	0.275
6	0.394	0.256
8	0.337	0.211
10	**0.271**	**0.183**
12	0.283	0.191
Backbone	0.202	0.111

7.3. *Group size (GRPO)*

Group size is specific to GRPO and determines the size of the group used to calculate the advantage. Similarly to Subesection 7.2, we fine-tune a predictive model on the Financial dataset with an increasing group size and test the MSE/MAE after fine-tuning. Table 9 presents the results of the model at group sizes from 2 to 12. We found that a group size of 8 is optimal for both computational load and model performance.

Table 9.: The influence of the group size when fine-tuning the model on the Financial dataset.

Group Size	Financial	
Metric	MSE	MAE
2	0.200	0.204
4	0.198	0.199
6	0.197	0.199
8	**0.195**	0.196
10	0.199	0.201
12	0.201	0.203
Backbone	0.202	**0.111**

8. Conclusion

Fine-tuning timeseries predictors is emerging as an essential post-supervised training step to improve the performance of models. As the paradigm shifts from local models to larger, eclectic models harnessing the predictive power of many timeseries from diverse fields, fine-tuning becomes even more essential. At scale, is it far more cost effective to fine-tune a large model to a specific use case than retraining on large datasets. As the computational load for supervised learning gets higher and the models get larger, which has been the trend observed in LLMs and timeseries predictors, fine-tuning becomes even more attractive.

There are still several limitations, the most prominent being that pre-trained models use a fixed size input vector. This problem is not encountered in standard large language models, as the alphabet is tokenized to represent the entirety of the model output. But timeseries prediction is a continuous process, and further innovation is needed in foundational model to break out of fixed size vectors and scale up the mod-

els on large datasets, without relying on tricks such as projection layers. In the same spirit, architectural changes to foundational model allowing for variable output vector size would benefit the industry integration of timeseries predictors.

The environment definition and reward structure are key to the success of fine-tuning. Empirically, we noticed better results by bounding the reward to values between -1 and 1. The algorithm used is also a determining factor, and GRPO emerges as the clear winner in this chapter. This result is in line with the recent advances in LLMs, and further strengthens the conjecture that LLMs and timeseries predictors based on the same architecture share scaling features. If this conjecture reveals to be true, timeseries predictors are in a fantastic second mover position to implement even more innovations the thriving LLM community is building.

References

[1] H. Cazaux, R. Rudd, H. Stefánsson, S. Ólafsson, and E. I. Ásgeirsson, "Non-stationary inverted transformer with time2vec embedding," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 1, 2024.

[2] K. W. Church, Z. Chen, and Y. Ma, "Emerging trends: A gentle introduction to fine-tuning," *Natural Language Engineering*, vol. 27, no. 6, pp. 763–778, 2021.

[3] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, *et al.*, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1299–1312, 2016.

[4] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146*, 2018.

[5] Meta, *Llama 3.1*, `https://llama.meta.com/`, 2024.

[6] J. Devlin, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[7] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[8] E. Cetinic, T. Lipic, and S. Grgic, "Fine-tuning convolutional neural networks for fine art classification," *Expert Systems with Applications*, vol. 114, pp. 107–118, 2018.

[9] H. Xiong, S. Wang, Y. Zhu, *et al.*, "Doctorglm: Fine-tuning your chinese doctor is not a herculean task," *arXiv preprint arXiv:2304.01097*, 2023.

[10] L. Luo, J. Ning, Y. Zhao, *et al.*, "Taiyi: A bilingual fine-tuned large language model for diverse biomedical tasks," *Journal of the American Medical Informatics Association*, ocae037, 2024.

[11] D. Vasan, M. Alazab, S. Wassan, H. Naeem, B. Safaei, and Q. Zheng, "Imcfn: Image-based malware classification using fine-tuned convolutional neural network architecture," *Computer Networks*, vol. 171, p. 107 138, 2020.

[12] L. Xu, H. Xie, S.-Z. J. Qin, X. Tao, and F. L. Wang, "Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment," *arXiv preprint arXiv:2312.12148*, 2023.

[13] Z. Fu, H. Yang, A. M.-C. So, W. Lam, L. Bing, and N. Collier, "On the effectiveness of parameter-efficient fine-tuning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, 2023, pp. 12 799–12 807.

[14] R. Zhang, J. Han, C. Liu, *et al.*, "Llama-adapter: Efficient fine-tuning of language models with zero-init attention," *arXiv preprint arXiv:2303.16199*, 2023.

[15] R. He, L. Liu, H. Ye, *et al.*, "On the effectiveness of adapter-based tuning for pretrained language model adaptation," *arXiv preprint arXiv:2106.03164*, 2021.

[16] E. J. Hu, Y. Shen, P. Wallis, *et al.*, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.
[17] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," *arXiv preprint arXiv:2101.00190*, 2021.
[18] B. Gunel, J. Du, A. Conneau, and V. Stoyanov, "Supervised contrastive learning for pre-trained language model fine-tuning," *arXiv preprint arXiv:2011.01403*, 2020.
[19] Y. Zhou and V. Srikumar, "A closer look at how fine-tuning changes bert," *arXiv preprint arXiv:2106.14282*, 2021.
[20] T. Zhang, F. Wu, A. Katiyar, K. Q. Weinberger, and Y. Artzi, "Revisiting few-sample bert fine-tuning," *arXiv preprint arXiv:2006.05987*, 2020.
[21] T. Fredriksson, D. I. Mattos, J. Bosch, and H. H. Olsson, "Data labeling: An empirical investigation into industrial challenges and mitigation strategies," in *International Conference on Product-Focused Software Process Improvement*, Springer, 2020, pp. 202–216.
[22] T. Chen, S. Liu, S. Chang, Y. Cheng, L. Amini, and Z. Wang, "Adversarial robustness: From self-supervised pre-training to fine-tuning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 699–708.
[23] Y. Fan, O. Watkins, Y. Du, *et al.*, "Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 79 858–79 885, 2023.
[24] S. Zhai, H. Bai, Z. Lin, *et al.*, "Fine-tuning large vision-language models as decision-making agents via reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 37, pp. 110 935–110 971, 2025.
[25] A. Liu, B. Feng, B. Xue, *et al.*, "Deepseek-v3 technical report," *arXiv preprint arXiv:2412.19437*, 2024.
[26] Y. Liu, H. Zhang, C. Li, X. Huang, J. Wang, and M. Long, "Timer: Generative pre-trained transformers are large time series models," in *Forty-first International Conference on Machine Learning*, 2024.
[27] C. Chang, W.-C. Peng, and T.-F. Chen, "Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms," *arXiv preprint arXiv:2308.08469*, 2023.
[28] Y. Liang, H. Wen, Y. Nie, *et al.*, "Foundation models for time series analysis: A tutorial and survey," in *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, 2024, pp. 6555–6565.
[29] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
[30] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, PMLR, 2015, pp. 1889–1897.
[31] Z. Wang, V. Bapst, N. Heess, *et al.*, "Sample efficient actor-critic with experience replay," *arXiv preprint arXiv:1611.01224*, 2016.
[32] L. Kaiser, M. Babaeizadeh, P. Milos, *et al.*, "Model-based reinforcement learning for atari," *arXiv preprint arXiv:1903.00374*, 2019.
[33] M. S. Holubar and M. A. Wiering, "Continuous-action reinforcement learning for playing racing games: Comparing spg to ppo," *arXiv preprint arXiv:2001.05270*, 2020.
[34] S.-Y. Han and T. Liang, "Reinforcement-learning-based vibration control for a vehicle semi-active suspension system via the ppo approach," *Applied Sciences*, vol. 12, no. 6, p. 3078, 2022.
[35] L. Zhang, Y. Zhang, X. Zhao, and Z. Zou, "Image captioning via proximal policy optimization," *Image and Vision Computing*, vol. 108, p. 104 126, 2021.
[36] T. Kobayashi, "Proximal policy optimization with relative pearson divergence," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 8416–8421.

[37] Y. Gu, Y. Cheng, C. P. Chen, and X. Wang, "Proximal policy optimization with policy feedback," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 7, pp. 4600–4610, 2021.

[38] C. C.-Y. Hsu, C. Mendler-Dünner, and M. Hardt, "Revisiting design choices in proximal policy optimization," *arXiv preprint arXiv:2009.10897*, 2020.

[39] Q. Cai, Z. Yang, C. Jin, and Z. Wang, "Provably efficient exploration in policy optimization," in *International Conference on Machine Learning*, PMLR, 2020, pp. 1283–1294.

[40] T. Yu, A. Kumar, R. Rafailov, A. Rajeswaran, S. Levine, and C. Finn, "Combo: Conservative offline model-based policy optimization," *Advances in neural information processing systems*, vol. 34, pp. 28 954–28 967, 2021.

[41] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," *Proceedings of the Tenth International Conference on Machine Learning*, 1993.

[42] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in Neural Information Processing Systems*, 2017.

[43] C. Berner, G. Brockman, B. Chan, *et al.*, "Dota 2 with large scale deep reinforcement learning," *arXiv preprint arXiv:1912.06680*, 2019.

[44] X. Liang, X. Du, G. Wang, and Z. Han, "Deep reinforcement learning for traffic light control in vehicular networks," *arXiv preprint arXiv:1904.08117*, 2019.

[45] L. Matignon, G. Laurent, and N. Le Fort-Piat, "Coordinated multi-agent learning: The state of the art," *Artificial Intelligence Review*, vol. 37, no. 3, pp. 219–250, 2012.

[46] T. Yu, G. Qu, A. Singh, S. Levine, and C. Finn, "Meta-learning with latent embedding optimization in multi-agent systems," in *International Conference on Learning Representations*, 2020.

[47] G. Palmer, K. Tuyls, D. Bloembergen, and R. Savani, "Lenient multi-agent deep reinforcement learning," *arXiv preprint arXiv:1805.04566*, 2018.

[48] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[49] X. Guo, H. Zhang, H. Yang, L. Xu, and Z. Ye, "A single attention-based combination of cnn and rnn for relation classification," *IEEE Access*, vol. 7, pp. 12 467–12 475, 2019.

[50] R. Wu, A. Zhang, I. Ilyas, and T. Rekatsinas, "Attention-based learning for missing data imputation in holoclean," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 307–325, 2020.

[51] M. J. Er, Y. Zhang, N. Wang, and M. Pratama, "Attention pooling-based convolutional neural network for sentence modelling," *Information Sciences*, vol. 373, pp. 388–403, 2016, ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2016.08.084. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025516306673.

[52] L. Bramlage and A. Cortese, "Generalized attention-weighted reinforcement learning," *Neural Networks*, vol. 145, pp. 10–21, 2022.

[53] W. Wang, Y. Zhang, Y. Sui, *et al.*, "Reinforcement-learning-guided source code summarization using hierarchical attention," *IEEE Transactions on software Engineering*, vol. 48, no. 1, pp. 102–119, 2020.

[54] U. Gunarathna, R. Borovica-Gajic, S. Karunasekara, and E. Tanin, "Solving dynamic graph problems with multi-attention deep reinforcement learning," *arXiv preprint arXiv:2201.04895*, 2022.

[55] C. Liu and G. Liu, "Jointppo: Diving deeper into the effectiveness of ppo in multi-agent reinforcement learning," *arXiv preprint arXiv:2404.11831*, 2024.

[56] Reuters, *Reuters*, https://www.reuters.com/, 2024.

[57] Sustainalytics, *Sustainalytics*, 2022. [Online]. Available: https://www.sustainalytics.com/.

[58] IFRS, *Ifrs s1*, https://www.ifrs.org/issued-standards/ifrs-sustainability-standards-navigator/ifrs-s1-general-requirements/, 2023.

[59] N. S. Soderstrom and K. J. Sun, "Ifrs adoption and accounting quality: A review," *European accounting review*, vol. 16, no. 4, pp. 675–702, 2007.

[60] E. F. Fama and K. R. French, "A five-factor asset pricing model," *Journal of Financial Economics*, vol. 116, no. 1, pp. 1–22, 2015.

[61] P. C. Belafsky, G. N. Postma, and J. A. Koufman, "Validity and reliability of the reflux symptom index (rsi)," *Journal of voice*, vol. 16, no. 2, pp. 274–277, 2002.

[62] T. T.-L. Chong and W.-K. Ng, "Technical analysis and the london stock exchange: Testing the macd and rsi rules using the ft30," *Applied Economics Letters*, vol. 15, no. 14, pp. 1111–1114, 2008.

[63] J. Bollinger, "Using bollinger bands," *Stocks & Commodities*, vol. 10, no. 2, pp. 47–51, 1992.

[64] H. Cazaux, R. Rudd, H. Stefánsson, S. Ólafsson, M. Raberto, and E. I. Ásgeirsson, "Correlation study between returns and esg ratings.," *Journal of Impact & ESG Investing*, vol. 5, no. 1, 2024.

[65] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2012, pp. 5026–5033. DOI: 10.1109/IROS.2012.6386109.

[66] L. Engstrom, A. Ilyas, S. Santurkar, *et al.*, "Implementation matters in deep policy gradients: A case study on ppo and trpo," *arXiv preprint arXiv:2005.12729*, 2020.

[67] S. Huang, R. F. J. Dossa, C. Ye, *et al.*, "Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms," *Journal of Machine Learning Research*, vol. 23, no. 274, pp. 1–18, 2022. [Online]. Available: http://jmlr.org/papers/v23/21-1342.html.