School of science and technology

Postgraduate Course

Master in Mathematics

Postgraduate Dissertation

Continuous-Time Modelling of Investor Sentiment with Neural Differential Equations and Applications to Stock Price Forecasting

Ioannis Pararas

Supervisor: “Nikolaos Matzakos”

Patras, Greece, June 2026

Continuous-Time Modelling of Investor Sentiment with Neural Differential Equations and Applications to Stock Price Forecasting

Ioannis Pararas

Supervising Committee

Supervisor:	Co-Supervisor:
Nikolaos Matzakos	Nikolaos Tsitsas

Patras, Greece, June 2026

ABSTRACT

This thesis investigates the use of Neural Ordinary and Stochastic Differential Equations (Neural ODEs and Neural SDEs) for modelling the continuous-time evolution of investor sentiment and its impact on financial markets. The work begins by collecting sentiment data from financial news sources, which inherently exhibit irregular sampling and stochastic variability. Sentiment scores—derived using LLM models such as GPT-4.0 —are treated as time-dependent signals representing collective market mood. Neural ODEs are then employed to learn smooth latent trajectories that capture the underlying dynamics of these sentiment processes, while Neural SDEs extend this approach to model uncertainty and volatility in sentiment evolution. The learned latent representations are subsequently integrated into downstream forecasting models, including Neural SDE-based predictors, to assess their contribution to stock price and return prediction. Comparative analyses are conducted against conventional discrete-time and feature-based baselines to quantify the benefits of continuous-time modelling. Overall, this research aims to determine whether modelling sentiment as a continuous stochastic process enhances the interpretability and predictive performance of financial time series models.

List of Tables

Table 1: Definitions of Stock Price Dataset Features (OHLCV) ……………………………………17

Table 2: Sentiment NSDE test metrics vs Baseline ....…………………………………...32

Table 3: Baseline price NSDE performance metrics vs Naive (tommorow = today) ...…………..…33

Table 4: Loss ablation (baseline, price space)......................... …………………………………...…34

Table 5: Variation comparison (lag L=1, price space)......................………………………..……45

Table 6: Lag sensitivity (price space)..……………………………………46

Table 7 Variation 3 multi-horizon evaluation (price space)).................……………………………51

Table 8 Classical and deep-learning baselines vs. price NSDE (identical data, price space)………..52

List of Figures

Figure 1: Thesis Structure Flowchart…………………… ……………………………………15

Figure 2: Evolution of stock price of NVDA ………………. ……………………………………18

Figure 3: Example of a token based sentiment evaluation of a financial headline……………… .…18

Figure 4: Evolution of sentiment scores ………………… ……………………………………19

Figure 5: Per-article score distribution…………………… …………………………………...…20

Figure 6: Agentic sentiment extraction workflow……………… ………………………………21

Figure 7: Neural ODE fit for the sentiment series: in-sample training fit and out-of-sample one-step-ahead test forecasts……………………………………..……………………………………28

Figure 8: NODE vector field over time with the learned trajectory and observed sentiment points………………………………..…………………………………29

Figure 9: 3D extended phase space of the NODE trajectory, showing smooth deterministic dynamics…………………………………………………………………..………………………30

Figure 10: *Neural ODE error behavior for sentiment: clustered deviations highlight unmodeled stochastic noise*……………………………..…………………… ………………………………31

Figure 11: *Neural ODE residual behavior for sentiment: clustered deviations highlight unmodeled stochastic noise*………………………………..……………… ……………………..…………………31

Figure 12: Neural SDE fit for the sentiment series: in-sample training fit and out-of-sample one-step-ahead test forecasts with 95% intervals. ....…………………………………32

Figure 13: Neural SDE on stock price (train + test)……… ……………………………………33

Figure 14: Training loss curve – baseline (variation 0)…………………… ………………………35

Figure 15: Training loss curve –(variation 1) …………………… ………………………………35

Figure 16: Training loss curve –(variation 2) ……………… ……………………………………35

Figure 17: Training loss curve –(variation 3) …………………… ………………………………36

Figure 18: Training loss curve –(variation 4) …………………… ………………………………36

Figure 19: Training loss curve –(variation 5) …………………… ………………………………36

Figure 20: Pearson correlations between NVDA daily Open-to-Open returns and lagged sentiment NSDE proxies. …………………… …………………………....................................…………38

Figure 21: Spearman correlations between NVDA daily Open-to-Open returns and lagged sentiment NSDE proxies..................................…………………… ……………………………………39

Figure 22: Variation 0 (baseline: no sentiment) — in-sample training fit and out-of-sample one-step-ahead forecasts with 95% predictive interval.……………………….………………………………..40

Figure 23: Variation 1 (trend→drift) — in-sample training fit and out-of-sample one-step-ahead forecasts with 95% predictive interval.......……………… ……………………………………..41

Figure 24: Variation 2 (trend→drift+diffusion) — in-sample training fit and out-of-sample one-step-ahead forecasts with 95% predictive interval.……………… ………………………………………42

Figure 25: Variation 3 (vol→diffusion) — in-sample training fit and out-of-sample one-step-ahead forecasts with 95% predictive interval. …………………… ..………………………………………42

Figure 26: Variation 4 (trend→drift, vol→diffusion) — in-sample training fit and out-of-sample one-step-ahead forecasts with 95% predictive interval. ………… ……………………………………...43

Figure 27: Variation 5 (full→both) — in-sample training fit and out-of-sample one-step-ahead forecasts with 95% predictive interval. …………………… ..…………………………………………44

Figure 28: Baseline Price NSDE — walk-forward versus long-horizon forecasts, confidence-interval width evolution, and sampled Monte Carlo paths……………………………………………………47

Figure 29: Baseline Price NSDE — RMSE, MAE, R^2, coverage, and interval width as a function of forecast horizon…………………………………………………..……………………………48

Figure 30: Variation 3 (sentiment volatility → diffusion) — walk-forward versus long-horizon forecasts, confidence-interval width evolution, and sampled Monte Carlo paths…………………………49

Figure 31: Variation 3 — RMSE, MAE, R^2, coverage, and interval width across forecast horizons…………………………………………………………………………………………50

Figure 32. Baseline comparison on the identical NVDA Open dataset……………………………….53

Table of Contents

List of Tables ....

List of Figures ....

Abstract ....

1. Introduction **1**

1.2 Personal Motivation 4

1.3 Goal of this thesis 5

1.4 Thesis Structure 6

2. Review of Related Work **7**

2.1 Review of Related Work 7

2.2 Theoretical Background 9

3. Methodology **15**

3.1 Thesis Assumptions 15

3.2 Dataset Description 16

3.3 Sentiment extraction via large language models 18

3.4 Proposed forecasting architecture and experimental protocol 21

3.5 Evaluation metrics 25

3.6 Long horizon and multi-step horizon forecasting 26

3.7 Implementation notes 27

4. Results 28

4.1 Node Implementation 28

4.2 Baseline price NSDE assessment 33

4.3 Loss (baseline, no sentiment) 34

4.4 Sentiment-augmented price NSDE 37

4.5 Lag sensitivity 45

4.6 Long-horizon and multi-step evaluation without sentiment 47

4.7 Long-horizon and multi-step evaluation with sentiment 49

4.8 Comparison with classical and deep learning baselines 51

4.9 Summary of results 54

5. Conclusions 55

5.1 Interpretation of findings 55

5.2 Limitations 56

5.3 Future work 56

References 58
Appendix 66

1. INTRODUCTION

1.1 Introduction

Financial markets operate in an information-rich environment in which prices continuously adjust in response to new signals. Beyond numerical indicators such as earnings, interest rates, or macroeconomic statistics, qualitative information conveyed through financial news plays a crucial role in shaping investor expectations and market behaviour. News related to corporate performance, strategic decisions, technological innovation, or broader economic conditions can influence market sentiment and trigger price movements, often within short time horizons. The modern financial ecosystem is characterized by unprecedented information velocity, with news disseminated globally within seconds through digital channels, social media platforms, and specialized financial information services. This rapid flow of information creates a dynamic environment in which market participants must continuously process and react to new data, making the study of information-driven price dynamics increasingly relevant.

The relationship between information and asset prices has been a central topic in financial economics for decades. The Efficient Market Hypothesis (EMH), introduced by Fama (1970), posits that asset prices fully reflect all available information, implying that consistent outperformance through information-based trading should be impossible in an efficient market. The semi-strong form of the EMH specifically suggests that prices adjust rapidly to publicly available information, including news announcements. However, empirical evidence has revealed systematic deviations from market efficiency, particularly in how prices respond to qualitative information. These deviations have motivated extensive research into the mechanisms through which news sentiment affects market dynamics and whether sentiment-based signals can provide predictive value beyond what is captured by price and volume data alone.

Behavioral finance offers a complementary perspective by emphasizing the role of investor psychology in price formation. Unlike the rational agent assumption underlying traditional finance theory, behavioral models recognize that market participants are subject to cognitive biases, emotional responses, and herd behavior. Investor sentiment, in this context, represents the collective psychological state of market participants and can drive prices away from fundamental values, at least temporarily. Periods of excessive optimism may lead to asset bubbles, while waves of pessimism can trigger sharp corrections that appear disproportionate to changes in underlying fundamentals. Understanding and quantifying this collective sentiment has therefore become an important objective for both academic researchers and market practitioners.

The concept of investor sentiment encompasses both rational expectations based on new information and irrational components driven by psychological factors. News media plays a dual role in this process: it serves as a channel for transmitting fundamental information about companies and economic conditions, and it simultaneously shapes and amplifies investor perceptions through framing, emphasis, and narrative construction. The tone and language used in financial reporting can influence how investors interpret objective facts, potentially creating feedback loops between media coverage and market behavior. This interplay between information content and presentation underscores the importance of sophisticated sentiment extraction methods capable of capturing nuanced textual signals.

A substantial body of empirical research has demonstrated that textual information contained in financial news conveys predictive signals about asset prices, volatility, and trading activity. The notion of investor sentiment—a measure of collective optimism or pessimism expressed in market discourse—has therefore become a central concept in financial economics and quantitative finance. Sentiment extracted from news articles, reports, and announcements has been shown to complement traditional price-based indicators by capturing aspects of market psychology that are not directly observable in numerical data. Studies have documented that news sentiment affects not only short-term price movements but also longer-horizon returns, trading volume, and measures of market uncertainty such as implied volatility.

The technology sector, and semiconductor companies in particular, presents a compelling case study for sentiment-based analysis. Companies such as NVIDIA (NVDA) operate at the intersection of multiple high-profile trends including artificial intelligence, data center computing, gaming, and autonomous vehicles. These firms attract substantial media attention and are subject to rapidly evolving narratives about technological progress, competitive dynamics, and future growth prospects. The stock prices of such companies can be highly sensitive to news about product launches, partnerships, regulatory developments, and broader industry trends. This heightened sensitivity to information flow makes technology stocks particularly suitable for investigating the relationship between news sentiment and price dynamics.

Despite its importance, modelling sentiment in a way that faithfully reflects its temporal evolution remains a challenging task. Financial sentiment does not change abruptly at fixed time intervals but instead evolves gradually as information accumulates and market participants update their beliefs. Moreover, sentiment data derived from news sources are often irregularly sampled and noisy, reflecting the asynchronous nature of news publication and the inherent uncertainty of textual interpretation. These characteristics pose significant challenges for conventional time-series modelling approaches. News arrives at unpredictable times throughout the day and week, with clustering around market events, earnings announcements, and macroeconomic releases. The resulting time series exhibits irregular spacing, missing observations on certain days, and heterogeneous information content across observations.

The challenge of irregular sampling is compounded by the inherent noise in sentiment measurement. Even with sophisticated extraction methods, converting unstructured text into numerical sentiment scores involves uncertainty arising from linguistic ambiguity, context dependence, and the subjective nature of sentiment itself. Different readers may interpret the same news article differently, and the relevance of a given piece of news to a specific stock may be unclear. These sources of measurement error must be accounted for in any rigorous modelling framework. Furthermore, the relationship between sentiment and prices may itself be time-varying, with stronger effects during periods of market stress or heightened uncertainty compared to calmer market conditions.

Advances in natural language processing have enabled increasingly sophisticated methods for extracting sentiment from financial text. Early approaches relied on manually curated sentiment lexicons and rule-based techniques, while more recent methods employ machine learning and deep learning models trained on domain-specific corpora. Transformer-based architectures and, more recently, large language models (LLMs) have demonstrated strong capabilities in capturing semantic nuance and contextual meaning in financial language. The evolution of sentiment analysis techniques reflects broader trends in artificial intelligence, progressing from simple keyword matching to statistical classifiers to deep neural networks capable of understanding complex linguistic patterns.

The development of finance-specific sentiment tools has been driven by the recognition that general-purpose methods often fail in financial contexts. Financial language contains specialized terminology, and words that carry negative connotations in everyday usage may have neutral or even positive meanings in financial discourse. For example, terms such as "liability," "risk," or "debt" are routine in financial reporting and do not necessarily indicate negative sentiment. Conversely, seemingly neutral phrases may carry implicit sentiment when understood in the context of market expectations and analyst forecasts. This domain specificity has motivated the creation of financial sentiment lexicons and the fine-tuning of language models on financial corpora.

An important advantage of LLM-based sentiment extraction is the ability to represent sentiment as a continuous-valued signal, rather than as a discrete category such as positive, neutral, or negative. Continuous sentiment scores provide a more expressive representation of sentiment intensity and allow for finer-grained analysis of how market mood evolves over time. This is particularly relevant in financial applications, where small changes in sentiment may precede observable movements in asset prices. A continuous representation preserves information about sentiment magnitude that would be lost in categorical classification, enabling downstream models to distinguish between mildly positive and strongly positive news, for instance.

The use of large language models for sentiment extraction also addresses several practical challenges that have limited earlier approaches. LLMs can perform sentiment analysis with minimal task-specific training, reducing the need for large labeled datasets that are expensive to create in specialized domains. They can handle the short, information-dense text typical of financial news headlines and summaries, extracting meaningful sentiment signals even from limited context. Furthermore, LLMs exhibit some ability to generalize across different types of financial content and adapt to evolving language patterns without explicit retraining, making them more robust to distributional shifts in news coverage over time.

Traditional financial time-series models, including autoregressive and volatility-based approaches, are formulated in discrete time and typically assume observations at fixed, evenly spaced intervals. While such models have been widely applied and remain useful in many contexts, they may struggle to represent processes that evolve continuously and irregularly, such as sentiment derived from news flows. Classical approaches such as ARIMA (Autoregressive Integrated Moving Average) and GARCH (Generalized Autoregressive Conditional Heteroskedasticity) have been workhorses of financial econometrics, providing interpretable models for return dynamics and volatility clustering. However, these models require regular time spacing and may not naturally accommodate the irregular arrival of news information.

The assumption of discrete, evenly spaced observations is particularly problematic when modeling sentiment dynamics. News does not arrive at regular intervals corresponding to market trading sessions. Important announcements may occur outside trading hours, on weekends, or in rapid succession during periods of heightened activity. Forcing such data into a regular time grid through aggregation or interpolation inevitably loses information and may introduce artifacts. A more principled approach would model the underlying sentiment process as continuous, with discrete observations representing noisy measurements of this latent process at irregular time points.

Deep learning approaches, including recurrent neural networks and long short-term memory (LSTM) models, have extended the modelling capacity of discrete-time frameworks by capturing nonlinear dependencies and long-range temporal patterns. Nevertheless, these models still operate on discretized time steps and may implicitly impose assumptions that are misaligned with the continuous nature of information arrival in financial markets. While LSTMs and related architectures have achieved impressive results in sequence modeling tasks, including financial forecasting, they treat time as an index over a sequence rather than as a continuous variable. This limitation becomes apparent when dealing with data sampled at varying frequencies or when attempting to make predictions at arbitrary time points not aligned with the training grid.

This mismatch between the continuous evolution of underlying market processes and the discrete structure of many predictive models motivates the exploration of alternative modelling paradigms that operate directly in continuous time. From a theoretical perspective, many processes in finance are naturally described by continuous-time stochastic models. Asset prices in mathematical finance are typically modeled using stochastic differential equations, with the Black-Scholes model and its extensions serving as foundational examples. Extending this continuous-time perspective to sentiment dynamics and sentiment-price interactions offers the potential for more theoretically coherent models that better reflect the underlying data-generating processes.

Neural differential equations provide a flexible framework for modelling dynamical systems whose evolution unfolds continuously over time. In particular, Neural Ordinary Differential Equations (Neural ODEs) represent the evolution of hidden states through a learned differential equation, enabling smooth trajectories that can be evaluated at arbitrary time points. This property makes Neural ODEs well suited for modelling latent processes inferred from irregularly sampled observations. By parameterizing the dynamics with neural networks, Neural ODEs combine the expressiveness of deep learning with the principled continuous-time formulation of differential equations.

The advantages of Neural ODEs extend beyond their ability to handle irregular sampling. These models offer memory efficiency during training through the use of adjoint sensitivity methods, which compute gradients without storing intermediate activations. They also provide a natural framework for incorporating prior knowledge about system dynamics through architectural choices and regularization. In the context of time series modeling, Neural ODEs can learn smooth latent representations that filter noise and capture underlying trends, potentially improving both interpretability and predictive performance.

In the context of sentiment analysis, Neural ODEs offer a principled way to model the latent continuous-time dynamics underlying observed sentiment scores. Rather than treating sentiment as a sequence of independent daily observations, this approach allows sentiment to be interpreted as the realization of an underlying dynamical system that evolves smoothly over time. This perspective is consistent with the intuition that market sentiment does not jump discontinuously but rather shifts gradually as new information is absorbed and interpreted by market participants. The learned dynamics can capture patterns such as mean reversion, momentum, and responses to external shocks in a unified framework.

To further account for uncertainty and randomness inherent in textual sentiment data, Neural Stochastic Differential Equations (Neural SDEs) extend this framework by explicitly modelling stochasticity in the system dynamics. This enables the representation of both the expected evolution of sentiment and the variability around it, providing a richer characterization of sentiment dynamics in noisy financial environments. The stochastic component can capture both measurement noise in sentiment extraction and genuine randomness in how collective market mood evolves. By learning both drift and diffusion components, Neural SDEs provide probabilistic forecasts with uncertainty quantification, which is essential for risk-aware decision making in financial applications.

The integration of sentiment-derived features with price prediction models represents the final component of the proposed framework. By incorporating learned sentiment representations—including trend and volatility proxies derived from the Neural SDE—into the price forecasting model, we can systematically evaluate whether continuous-time sentiment modeling provides incremental predictive value. This integration must be performed carefully to avoid data leakage and ensure that the evaluation reflects realistic forecasting scenarios where future information is not available at prediction time.

1.2 Personal Motivation

The choice of this thesis topic reflects a convergence of personal interests that have developed throughout my academic journey: a deep fascination with artificial intelligence and machine learning, an intellectual curiosity about cutting-edge research at the intersection of multiple disciplines, and career aspirations in fields where these technologies are reshaping traditional practices.

My interest in artificial intelligence and machine learning began during my undergraduate studies and has grown substantially as the field has undergone remarkable advances in recent years. The emergence of deep learning as a dominant paradigm, followed by the transformative impact of large language models, has been genuinely exciting to witness and study. Neural networks, in particular, have captured my imagination—not merely as powerful function approximators, but as systems that can learn complex representations and dynamics directly from data. The ability of these models to discover patterns that elude explicit programming represents, in my view, a fundamental shift in how we approach computational problem-solving. Neural Ordinary Differential Equations and their stochastic extensions represent a particularly elegant development in this space, bridging the gap between the discrete computational graphs of traditional neural networks and the continuous mathematical structures that describe many real-world phenomena.

Beyond the technical appeal of machine learning methods, I have been drawn to the challenge of applying these tools to domains where their impact can be both intellectually interesting and practically meaningful. Financial markets present such a domain. The combination of vast data availability, clear success metrics, and genuine economic stakes makes finance an ideal testing ground for advanced analytical methods. Moreover, the efficient market hypothesis and the inherent difficulty of prediction create a challenging benchmark: any method that demonstrates even modest predictive improvement is contributing something non-trivial. The interplay between quantitative analysis and human behavior in financial markets adds another layer of complexity that I find intellectually stimulating.

The specific focus on natural language processing and sentiment analysis reflects my belief that some of the most valuable applications of AI lie at the boundary between structured numerical data and unstructured human communication. Financial news represents a rich source of information that has traditionally been difficult to incorporate into quantitative models. The recent capabilities of large language models to extract nuanced meaning from text open new possibilities for bridging this gap. Working with these models—understanding their capabilities and limitations, designing appropriate prompts, and integrating their outputs into downstream systems—has been a valuable learning experience that I expect to draw upon throughout my career.

From a career perspective, the skills developed through this thesis align closely with roles in quantitative finance, fintech, and data science more broadly. The financial industry is increasingly seeking professionals who can combine domain knowledge with advanced technical skills in machine learning and data analysis. Experience with time-series modeling, natural language processing, and probabilistic forecasting—all central to this thesis—is directly relevant to roles ranging from quantitative research to algorithmic trading to risk management. Beyond finance specifically, the methodological skills acquired through this work—including working with neural differential equations, implementing end-to-end machine learning pipelines, and conducting rigorous experimental evaluations—are transferable to many other domains where data-driven decision making is becoming essential.

Finally, this thesis provided an opportunity to engage with genuinely novel research directions. While sentiment analysis and stock price prediction are well-established topics individually, their combination with continuous-time neural models remains relatively unexplored. Contributing to this emerging area, even in a modest way, has been intellectually rewarding. The process of reading recent literature, identifying gaps, formulating research questions, and designing experiments to address them has deepened my appreciation for the research process and reinforced my interest in pursuing technically challenging work throughout my career.

1.3 Goal of this thesis

The goal of this thesis is to investigate the extent to which financial news sentiment, extracted from textual data and modelled as a continuous-time process, can enhance stock price prediction. To this end, continuous sentiment scores are derived from financial news article descriptions using a large language model and aggregated into a time series corresponding to a specific publicly traded company. The temporal evolution of this sentiment series is modelled using Neural Ordinary Differential Equations in order to uncover latent sentiment dynamics. These learned representations are subsequently incorporated into a forecasting framework for predicting future stock price movements, with the aim of evaluating the effectiveness of continuous-time sentiment modelling in capturing market behavior beyond traditional price-based approaches.

1.4 Thesis Structure

This thesis is structured as follows. In Section **1**, Introduction, the general context of financial markets and the role of information in asset price formation are introduced. The chapter discusses the importance of financial news sentiment as a market signal and motivates the use of textual data for stock price prediction. The research problem addressed in this thesis is formulated, and the objectives and scope of the study are defined.

In Section 2, Review of related work, a review of related work is presented. This chapter summarizes existing research on financial sentiment analysis, the use of natural language processing and large language models in finance, and time-series forecasting methods for stock price prediction. Particular emphasis is placed on studies that combine sentiment information with market data, as well as on recent advances in continuous-time modeling approaches such as Neural Ordinary Differential Equations. Finally, the theoretical background of Neural Ordinary Differential Equations and the evaluation metrics used for model assessment are introduced.

In Section 3, Methodology, the methodology followed in this thesis is described in detail. First, the data sources are presented, including the financial news data and stock price data used in the analysis. The process of sentiment extraction using a large language model is then explained, along with the construction of the sentiment time series. Subsequently, the data preprocessing steps, exploratory analysis, and feature engineering procedures are discussed as well as the proposed methodology as it was applied in this thesis.

In Section 4, Results, the experimental setup and results are presented and analyzed. This chapter focuses on the training and testing of the proposed models, including the Neural ODE-based sentiment modeling and the stock price prediction framework. The predictive performance of the models is evaluated and compared, and the results are discussed in relation to the objectives of the thesis.

Finally, Section 5, Conclusions, concludes the thesis by summarizing the main findings and discussing the limitations of the proposed approach. Possible directions for future research and potential extensions of the methodology are also outlined.

2. REVIEW OF RELATED WORK

2.1 REVIEW OF RELATED WORK

This chapter reviews existing research related to the use of financial news sentiment for stock market prediction, the evolution of sentiment analysis techniques in finance, and the application of advanced time-series modelling methods. Particular emphasis is placed on recent developments in large language models for sentiment extraction and continuous-time modelling approaches, which form the methodological foundation of this thesis.

The relationship between financial news and asset price dynamics has been extensively studied in the literature. Early empirical evidence established that textual information contained in news articles conveys valuable signals about investor sentiment and future market behavior. Tetlock demonstrated that the tone of news coverage is correlated with stock returns and trading volume, highlighting that market prices respond not only to numerical data but also to qualitative information embedded in text (Tetlock, 2007). This line of research motivated the systematic extraction of sentiment indicators from financial news as inputs for predictive models.

Many studies report **weak unconditional correlations** between text sentiment and returns. Ranco et al. (2015) study Twitter sentiment and stock returns for DJIA constituents and report that unconditional Pearson correlation and Granger causality between sentiment and returns are generally weak. However, by reframing the problem as an event-study—where events are defined as peaks in Twitter message volume—they find statistically significant cumulative abnormal returns following high-volume sentiment events. Their results suggest that sentiment effects may be episodic and concentrated around periods of elevated information flow rather than persistent in the unconditional time series.

Subsequent studies have confirmed that news sentiment affects both short-term price movements and longer-term market dynamics. Davidovic and McCleary show that aggregated news sentiment exhibits measurable influence on stock volatility and return patterns, supporting the view that sentiment evolves as a latent market factor rather than as isolated discrete events (Davidovic, M. & McCleary, 2025)

Initial approaches to financial sentiment analysis relied primarily on dictionary-based methods and predefined sentiment lexicons. However, such approaches often struggled in financial contexts due to the domain-specific nature of financial language. Loughran and McDonald demonstrated that general-purpose sentiment dictionaries misclassify many financial terms, leading to biased or misleading sentiment estimates (Loughran & McDonald, 2011). Their work motivated the construction of finance-specific lexicons and the use of supervised learning techniques.

Despite these improvements, many traditional sentiment analysis methods represent sentiment using discrete labels (e.g., positive, neutral, negative). While such representations are intuitive, they fail to capture the intensity of sentiment and may discard valuable information, particularly when sentiment changes gradually over time. This limitation is especially relevant in time-series forecasting applications, where smooth variations in sentiment may precede observable price movements.

The introduction of machine learning and deep learning techniques significantly advanced sentiment analysis in finance. Transformer-based models typically require sufficiently long and information-rich text to perform reliably. When applied to short financial news summaries or article descriptions, their predictive accuracy may degrade due to limited contextual information. This limitation is particularly relevant in practical settings where access to full article text may be restricted due to cost, copyright, or data availability constraints.

Recent advances in large language models (LLMs) have further expanded the capabilities of sentiment extraction from financial text. Brown et al. introduced GPT-style architectures capable of few-shot and zero-shot learning, allowing sentiment analysis to be performed without extensive task-specific fine-tuning (Brown et al., 2020) . These models have been shown to capture semantic nuance and implicit sentiment even in short texts.

Recent studies suggest that LLM-based sentiment extraction can outperform traditional classifiers in financial applications, particularly in low-resource settings. Kirtac and Germano demonstrate that sentiment signals generated by large language models can be effectively used in trading strategies, outperforming benchmark sentiment indicators (Kirtac & Germano, 2024). Importantly, LLMs enable sentiment to be represented as a continuous score rather than a discrete category, allowing for a more expressive characterization of sentiment intensity.

Continuous sentiment representations have been shown to provide stronger explanatory power in financial forecasting tasks than categorical indicators. By preserving sentiment magnitude, continuous signals allow models to capture subtle shifts in market perception that may not be detectable through discrete labels alone.

Financial time series are characterized by nonlinear dynamics, noise, and non-stationarity, which pose challenges for classical statistical models such as ARIMA and GARCH. While deep learning models such as recurrent neural networks and LSTMs have been widely applied to stock price forecasting, these models operate in discrete time and may struggle to represent underlying continuous processes.

To address this limitation, Chen et al. introduced Neural Ordinary Differential Equations (Neural ODEs), a framework in which the evolution of hidden states is governed by a learned differential equation (Chen et al., 2018). Neural ODEs provide a principled approach to modelling continuous-time dynamics and have demonstrated promising results in time-series reconstruction and forecasting tasks. Their ability to model latent processes evolving continuously over time makes them particularly suitable for representing sentiment dynamics derived from irregularly spaced or smoothly varying textual data.

While prior work has explored the use of sentiment for stock price prediction and the application of deep learning models to financial forecasting, limited research has investigated the integration of continuous sentiment signals extracted via large language models with continuous-time neural modelling frameworks. Most existing approaches either rely on discrete sentiment representations or employ discrete-time forecasting architectures.

Neural SDE performance can be highly sensitive to the chosen drift/diffusion parameterization: naive designs may cause unstable training or poor robustness under irregular sampling and missingness. Oh et al. (ICLR 2024) address this by proposing *stable* Neural SDE classes (e.g., Langevin-type, linear-noise, and geometric forms) with theoretical support and empirical evidence of improved robustness under missingness-induced distribution shift, motivating diffusion constraints and stability-aware SDE parameterizations as first-class modelling choices.

This thesis addresses this gap by combining large language models for continuous sentiment extraction with Neural Ordinary Differential Equations to model latent sentiment dynamics and their relationship with stock price movements. By integrating advances in natural language processing and continuous-time machine learning, this work contributes to the growing literature on sentiment-enhanced financial forecasting.

2.2 Theoretical Background

This section provides the theoretical foundations underlying the models employed in this thesis. We begin with an overview of neural network fundamentals, then introduce residual networks as a bridge to continuous-time formulations. Subsequently, we present the mathematical framework of Neural Ordinary Differential Equations (Neural ODEs) and their stochastic extensions (Neural SDEs). Finally, we briefly discuss the evaluation metrics used to assess model performance.

2.2.1 Fundamentals of Neural Networks

Artificial neural networks are computational models inspired by the structure and function of biological neural systems. A neural network consists of interconnected processing units called neurons, organized in layers that transform input data through a sequence of nonlinear operations. The fundamental building block is the feedforward layer, which computes an affine transformation followed by a pointwise nonlinearity.

Formally, consider an input vector x ∈ R^n. A single fully-connected layer computes:

$$h = \sigma(Wx + b) \qquad (1)$$

where W ∈ R^(m×n) is a weight matrix, b ∈ R^m is a bias vector, and σ(·) is a nonlinear activation function applied element-wise. Common activation functions include the rectified linear unit (ReLU), defined as σ(z) = max(0, z), and the hyperbolic tangent, σ(z) = tanh(z). The choice of activation function affects both the representational capacity of the network and the dynamics of gradient-based optimization.

Deep neural networks stack multiple such layers to form hierarchical representations. For a network with L layers, the forward computation proceeds as:

$$h^{\{(0)\}} = x \qquad (2)$$

$$h^{\{(l)\}} = \sigma\left(W^{\{(l)\}h^{\{(l-1)\}}} + b^{\{(l)\}}\right), \; l = 1, \ldots, L \qquad (3)$$

$$y = h^{\{(L)\}} \qquad (4)$$

where $h^{\{(L)\}}$ denotes the hidden representation at layer l. The parameters θ = $\{W^{(1)}, b^{(1)}, \ldots, W^{(L)}, b^{(L)}\}$are learned by minimizing a loss function L(y, y*) that measures the discrepancy between the network output y and the target y*.

Training is performed via backpropagation, an efficient algorithm for computing gradients of the loss with respect to all parameters by applying the chain rule of calculus layer by layer. The gradients are then used to update parameters via gradient descent or its variants such as stochastic gradient descent (SGD), Adam, or RMSprop. The backpropagation algorithm, formalized by Rumelhart, Hinton, and Williams (1986), enabled the practical training of multi-layer networks and remains the foundation of modern deep learning.

The universal approximation theorem establishes that feedforward networks with a single hidden layer containing a sufficient number of neurons can approximate any continuous function on a compact domain to arbitrary precision (Hornik et al., 1989; Cybenko, 1989). However, theoretical guarantees on approximation do not address the practical challenges of training such networks or the sample efficiency of learning. In practice, deeper networks with appropriate architectural innovations have proven more effective than shallow networks with equivalent parameter counts.

2.2.2 Residual Networks and the Connection to Continuous Dynamics

A significant advancement in deep learning came with the introduction of Residual Networks (ResNets) by He et al. (2015). Training very deep networks was historically difficult due to the vanishing gradient problem, where gradients become exponentially small as they propagate through many layers, making weight updates ineffective. ResNets address this by introducing skip connections that allow gradients to flow directly through the network.

In a standard residual block, the output is computed as:

$$h_{\{t+1\}} = h_t + f(h_t, \theta_t) \qquad (5)$$

where f represents the residual function learned by the network layers. Rather than learning the direct mapping from $h_t \ to \ h_{t+1}$, the network learns the residual $f(h_t, \theta_t) = h_{t+1} - h_t$. This formulation allows the identity mapping to be learned easily (by driving f toward zero) and facilitates gradient flow through the skip connection.

The success of ResNets in training networks with over 100 layers demonstrated that extremely deep networks could be effectively optimized when equipped with appropriate architectural structures. He et al. (2015) achieved 3.57% top-5 error on ImageNet with a 152-layer network, winning the ILSVRC 2015 classification competition. Subsequent work extended these ideas to various domains and architectures.

The residual formulation $h_{t+1} = h_t + f(h_t, \theta_t)$ bears a striking resemblance to the Euler discretization of an ordinary differential equation. If we consider t as a continuous variable and take the limit as the step size approaches zero, the discrete residual update converges to:

$$\frac{dh}{dt} = f(h,(t), t\,\theta) \qquad (6)$$

This observation, formalized by Chen et al. (2018), provides the conceptual foundation for Neural Ordinary Differential Equations. The discrete "depth" of a ResNet becomes a continuous integration time, and the layer-wise transformations become the instantaneous rate of change of the hidden state. This continuous perspective offers both theoretical insights into the nature of deep learning and practical advantages in terms of memory efficiency and adaptive computation.

2.2.3 Neural Ordinary Differential Equations

Neural Ordinary Differential Equations (Neural ODEs), introduced by Chen et al. (2018) in their seminal NeurIPS paper "Neural Ordinary Differential Equations," represent a paradigm shift in how neural networks can be conceptualized and implemented. Rather than specifying a fixed sequence of discrete transformations, Neural ODEs define the dynamics of hidden states through a continuous-time differential equation parameterized by a neural network.

The core formulation is:

$$\frac{dh(t)}{dt} = f((h(t), \theta, t) \qquad (7)$$

where h(t) ∈ R^d is the hidden state at time t, and f_θ is a neural network with parameters θ that specifies the instantaneous rate of change. Given an initial condition $h(t_0) = h_0$, the hidden state at any subsequent time t_1 is obtained by solving the initial value problem:

$$h(t_1) = h(t_0) + \int_{t_0}^{t_1} f((h(t), \theta, t) \qquad (8)$$

In practice, this integral is computed numerically using standard ODE solvers such as Euler's method, Runge-Kutta methods, or adaptive solvers like Dormand-Prince (DOPRI5). The choice of solver involves a trade-off between computational cost and numerical accuracy.

A key innovation of Chen et al. (2018) is the adjoint sensitivity method for computing gradients. Rather than backpropagating through the operations of the ODE solver (which would require storing all intermediate states), the adjoint method solves a separate ODE backward in time to compute the gradients. Specifically, defining the adjoint state $a(t) = \frac{\partial L}{\partial h(t)}$, where L is the loss function, the gradients can be computed by solving

$$\frac{da(t)}{dt} = -\frac{a(t)^T \partial f_\theta}{\partial h} \qquad (9)$$

backward from t_1 to t_0, along with auxiliary equations for the parameter gradients. This approach has O(1) memory cost regardless of the number of solver steps, compared to O(N) for naive backpropagation through N solver steps.

Neural ODEs offer several advantages for time series modeling:

Continuous-time representation: The model defines the hidden state h(t) for any t in the integration interval, not just at discrete observation times. This is particularly valuable for irregularly sampled data where observations do not occur at fixed intervals.

Adaptive computation: Modern ODE solvers automatically adjust their step sizes based on the local complexity of the dynamics. Regions where the dynamics are smooth require fewer function evaluations, while regions with rapid changes receive more attention. This adaptive behavior can improve both accuracy and efficiency.

Memory efficiency: The adjoint method enables training with constant memory cost, making it feasible to train models with very long effective depths that would be prohibitive with standard backpropagation.

Interpretable dynamics: The learned vector field f_θ directly represents the rate of change of the system state, which can provide insights into the modeled process. For example, regions where f_θ points toward a fixed point indicate stable equilibria.

However, Neural ODEs also present challenges. The computational cost depends on the complexity of the learned dynamics—if the network learns a stiff or highly nonlinear vector field, the solver may require many small steps. Finlay et al. (2020) proposed regularization techniques including Jacobian penalty and kinetic energy regularization that encourage smoother dynamics and reduce the number of function evaluations. Kelly et al. (2020) introduced methods to directly optimize for solver efficiency by including a differentiable surrogate for computation time in the training objective.

2.2.4 Neural Stochastic Differential Equations

While Neural ODEs provide a powerful framework for modeling deterministic continuous-time dynamics, many real-world processes—including financial markets and sentiment evolution—exhibit inherent randomness that cannot be captured by deterministic models. Neural Stochastic Differential Equations (Neural SDEs) extend the Neural ODE framework to incorporate stochasticity directly into the dynamics.

A stochastic differential equation (SDE) in the Itô sense takes the form:

$$dx_t = f(x_t, t)dt + g(x_t, t)dW_t \qquad (10)$$

where:

- $x_t \in R^d$ is the state process at time t
- $f: R^d \times R \rightarrow R^d$ is the drift function, representing the deterministic trend
- $g: R^d \times R \rightarrow R^{d\times m}$ is the diffusion function, representing the noise intensity
- W_t is an m-dimensional standard Brownian motion (Wiener process)

The drift term $f(x_t, t)dt$ captures the expected instantaneous change in the state, while the diffusion term $g(x_t, t)dW_t$ introduces random fluctuations scaled by the diffusion coefficient. The Brownian motion W_t has independent Gaussian increments: for any $s < t$, $W_t - W_s \sim N(0, (t-s)I)$.

In a Neural SDE, both the drift f_θ and diffusion g_φ are parameterized by neural networks with learnable parameters θ and φ respectively. This allows the model to learn both the deterministic trends and the state-dependent noise structure from data. To ensure well-defined dynamics, the diffusion output is typically constrained to be non-negative through transformations such as the softplus function:

$$softplus(z) = log\big(1 + exp(z)\big). \qquad (11)$$

The theoretical foundations of SDEs were established by Kiyosi Itô in the 1940s through his development of stochastic calculus. A key result is Itô's lemma, which provides a rule for computing the differential of a function of a stochastic process. For a twice-differentiable function $F(x, t)$, Itô's lemma states:

$$dF = \left(\frac{\partial F}{\partial t} + f\frac{\partial F}{\partial x} + \left(\frac{1}{2}\right)g^2\frac{\partial^2 F}{\partial x^2}\right)dt + g\frac{\partial F}{\partial x}dW_t \qquad (12)$$

The additional second-order term $\left(\frac{1}{2}\right)g^2\frac{\partial^2 F}{\partial x^2}$ arises from the quadratic variation of Brownian motion and has no analog in ordinary calculus.

Training Neural SDEs requires computing gradients through stochastic dynamics, which presents additional challenges compared to the deterministic case. Li et al. (2020) extended the adjoint sensitivity method to SDEs, deriving a backward SDE whose solution yields the required gradients. A key technical contribution is a memory-efficient algorithm for reconstructing the Brownian path during the backward pass using a "Virtual Brownian Tree," which stores only the random seed rather than the full trajectory.

Kidger et al. (2021) established a connection between Neural SDEs and Generative Adversarial Networks (GANs), showing that fitting an SDE to data can be viewed as a Wasserstein GAN where the generator produces sample paths and the discriminator is parameterized as a Neural Controlled Differential Equation. This perspective provides theoretical guarantees: since the Wasserstein distance has a unique global minimum, any target SDE can be learned in the infinite data limit. The authors released their implementation as part of the torchsde library, which provides practical tools for training Neural SDEs.

For financial applications, SDEs have a long history dating back to the geometric Brownian motion model of Black and Scholes (1973). The ability to learn flexible drift and diffusion functions from data, rather than specifying them a priori, makes Neural SDEs particularly attractive for capturing complex market dynamics that may not conform to classical parametric models.

2.2.5 Numerical Methods for Stochastic Differential Equations

Solving SDEs numerically requires specialized methods that account for the stochastic nature of the dynamics. The simplest and most widely used method is the Euler-Maruyama scheme, which discretizes the SDE as:

$$x_{t+\Delta t} = x_t + f(x_t, t)\Delta t + g(x_t, t)\Delta W_t \qquad (13)$$

where $\Delta W_t \sim N(0, \Delta t)$ is a Gaussian increment. This scheme has strong order 0.5 and weak order 1.0, meaning that the pathwise error decreases as $O(\sqrt{\Delta t})$ while the error in expected values decreases as $O(\Delta t)$.

Higher-order methods such as the Milstein scheme achieve strong order 1.0 by including an additional correction term involving the derivative of the diffusion coefficient. However, for multi-dimensional SDEs with non-commutative noise, implementing Milstein requires simulating iterated stochastic integrals (Lévy areas), which adds complexity.

For Neural SDEs in machine learning applications, the Euler-Maruyama scheme is often preferred due to its simplicity and the fact that the learned dynamics can adapt to compensate for discretization error. Adaptive step-size methods developed for ODEs can be extended to SDEs, though care must be taken to handle the stochastic component consistently.

An important consideration for prediction and uncertainty quantification is that each simulation of an SDE produces a different sample path due to the random increments. To characterize the distribution of outcomes, Monte Carlo sampling is employed: the SDE is simulated N times with independent noise realizations, and statistics such as the mean and variance are estimated from the ensemble of paths:

$$\hat{\mu}_t = \left(\frac{1}{N}\right) \Sigma_{j=1}^{N} x_t^j \qquad (14)$$

$$\hat{\sigma}_t^2 = \left(\frac{1}{N-1}\right) \Sigma_{j=1}^{N} \left(x_t^j - \hat{\mu}_t\right)^2 \qquad (15)$$

Prediction intervals can be constructed from empirical quantiles of the simulated distribution. For instance, the 95% prediction interval is bounded by the 2.5th and 97.5th percentiles of the Monte Carlo samples.

2.2.6 Evaluation Metrics

This section briefly summarizes the metrics used to evaluate model performance. Since Neural SDEs produce probabilistic forecasts, we employ both point forecast metrics and probabilistic metrics.

Point Forecast Metrics Point forecasts are typically taken as the mean of the predictive distribution. Common metrics include:

Mean Absolute Error (MAE): Measures the average magnitude of prediction errors without considering direction.

$$MAE = \left(\frac{1}{T}\right) \Sigma_{t=1}^{T} |y_t - \hat{y}_t| \qquad (16)$$

Root Mean Squared Error (RMSE): Similar to MAE but penalizes larger errors more heavily due to the squaring operation.

$$RMSE = \sqrt{\left(\frac{1}{T}\right) \Sigma_{t=1}^{T} (y_t - \hat{y}_t)^2} \qquad (17)$$

Coefficient of Determination (R^2): Measures the proportion of variance in the target variable explained by the model, relative to a baseline that predicts the mean.

$$R^2 = 1 - \frac{(\Sigma_{t=1}^{T} (y_t - \hat{y}_t)^2)}{(\Sigma_{t=1}^{T} (y_t - \bar{y})^2)} \qquad (18)$$

An R^2 of 1 indicates perfect prediction, while R^2 of 0 indicates performance equivalent to predicting the mean. Negative R^2 indicates performance worse than the mean baseline.

Directional Accuracy: For financial applications, correctly predicting the direction of price movement can be more important than minimizing absolute error. Directional accuracy is the fraction of time steps where the predicted direction matches the realized direction.

Probabilistic Metrics Because Neural SDEs provide full predictive distributions, we also evaluate:

Coverage: The empirical frequency with which the true value falls within the predicted confidence interval. For well-calibrated 95% intervals, coverage should be approximately 0.95.

Interval Width (Sharpness): The average width of prediction intervals. For a given coverage level, narrower intervals indicate more precise predictions.

Negative Log-Likelihood (NLL): When the predictive distribution is approximated as Gaussian with mean μ̂_t and variance σ̂²_t, the NLL provides a proper scoring rule that rewards both accuracy and appropriate uncertainty:

$$NLL = \left(\frac{1}{T}\right) \Sigma_{t=1}^{T} \left[log(\hat{\sigma}_t) + \frac{(y_t - \hat{\mu}_t)^2}{2\hat{\sigma}_t^2} \right] \qquad (19)$$

Lower NLL indicates better probabilistic forecasts. Unlike point metrics, NLL penalizes both overconfident predictions (low variance when errors are large) and underconfident predictions (high variance when predictions are accurate).

3. Methodology

In this chapter we lay the theoretical ground for the main analysis of the thesis. In the first segment we state the assumptions we make across the thesis. In the second segment we give a detailed description of the provided dataset and the included features.
Lastly, we provide a short review of the theoretical background of the algorithms used when creating the model as well as the metrics used to evaluate it.
The overall process of the project is illustrated in the flowchart in figure 1.

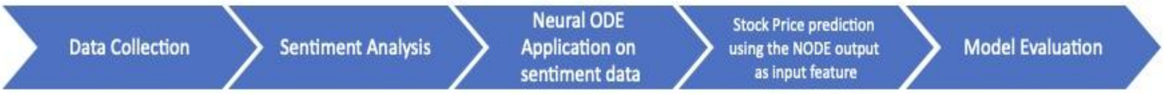

Figure 1: Project Flowchart

3.1 Thesis Assumptions

In this segment we state the various assumptions we make in this thesis. First, sentiment extraction is performed on financial news article descriptions rather than on the full article text. This choice is motivated by both practical and regulatory considerations. Full-text article scraping is often subject to legal and copyright restrictions, while the use of large language models on lengthy textual inputs can be computationally and financially costly. It is therefore assumed that article descriptions, which are designed to summarize the core content of news items, contain sufficient semantic information to capture the dominant sentiment relevant to market participants—particularly when analyzed using large language models capable of inferring sentiment from short and context-limited text.

Second, only article descriptions that explicitly mention the name of the target publicly traded company are retained in the dataset. Given the short length of article descriptions, typically consisting of one to two sentences, it is assumed that sentiment signals extracted from texts lacking an explicit company reference may be ambiguous or unrelated to the stock under consideration. By restricting the dataset to descriptions that directly reference the company, the analysis assumes that the extracted sentiment more accurately reflects firm-specific information and reduces the risk of false positive or false negative sentiment attribution.

Firm-specific filtering improves attribution. Only descriptions that explicitly mention the target publicly traded company are retained. Because descriptions are short (often one to two sentences), texts without an explicit mention may be ambiguous or unrelated to the firm. By restricting the dataset to descriptions containing the firm's name/ticker, we assume the extracted sentiment is more likely to reflect *firm-specific* information and reduces misattribution.

Sentiment is modeled as a continuous-time latent process. News arrives asynchronously and investor mood is not expected to change only at fixed discrete intervals. We adopt the view that the observed daily sentiment score is a noisy measurement of an underlying latent process evolving in continuous time, with irregular sampling and stochastic variability.

Predictive contribution, not strict causality. The experiments are designed to evaluate whether sentiment-derived signals improve forecasting of NVDA prices/returns under leakage-safe protocols. The results are interpreted as predictive associations consistent with sentiment usefulness, rather than definitive causal claims.

No-leakage rule. At forecast time t, the model for t+1 must only use information that would have been available up to (and including) time t. This is enforced both in feature construction and in evaluation:

- The sentiment model produces out-of-sample one-step-ahead forecasts on the sentiment test period.
- The price model uses only sentiment features that are in-sample for price-train rows and out-of-sample for price-test rows.
- Where lead–lag effects are plausible, sentiment features are lagged so that same-day sentiment is used to predict next-session prices.

3.2 Dataset Description

This section describes the datasets used for the sentiment and stock price time series.

3.2.1 *Financial news dataset*

News items related to the target company (NVDA) are collected from a financial news provider API and stored in JSON format. Each item includes at least:

- a publication timestamp (published_utc),
- a short text description (description),
- a firm-specific sentiment label (e.g., positive/neutral/negative) when available (NVDA_sentiment), and
- a continuous sentiment score (sentiment_score) produced by an LLM-based procedure described in Section 3.3.

In the current dataset used in this thesis:

- Total collected news articles: **2041**
- Articles with a valid continuous sentiment score: **1950**
- News timestamp range: **2025-01-02 → 2026-01-04**

Not every article yields a valid continuous score (e.g., due to missing fields or filtering decisions). Such items are excluded from the sentiment time-series construction.

3.2.2 *Sentiment time-series construction (daily aggregation)*

News arrive irregularly, with multiple articles on some days and no articles on others. To obtain a daily sentiment series that can be aligned with daily market prices, we aggregate the scored articles by calendar date.
Let $\mathcal{A}d$ be the set of scored articles published on day d, and let $\overline{s}_i \in [0,1]$ denote the continuous sentiment score for article i. We define the daily sentiment observation as the mean:

$$s_d = \frac{1}{A_d} \sum_{i \in A_d} \overline{s}_i \qquad (20)$$

This produces a daily series with the following characteristics:

- Unique calendar dates with sentiment observations: **352**
- Daily sentiment date range: **2025-01-11 → 2026-01-03**

Days without news are not imputed at this stage; instead, the continuous-time models are trained to handle irregular sampling, and the downstream price alignment is performed on trading dates (Section 3.5.3).

3.2.3 *Stock price dataset*

Historical daily price data for NVDA are obtained from Yahoo Finance via the yfinance Python package. The dataset includes standard fields displayed in table 1 below:

Feature name	***Description***
Open	The initial price at which a financial instrument begins trading at the start of a specified time interval.
High	The maximum price reached by the asset during the specified time interval.
Low	The minimum price reached by the asset during the specified time interval.
Close	The final price at which the asset trades at the conclusion of the specified time interval.
Volume	The total number of shares or contracts exchanged between buyers and sellers throughout the specified time interval.

Table 1: Definitions of Stock Price Dataset Features (OHLCV)

Open, HLCV fields and corporate action columns. For the main experiments we focus on the **Open** price and use the remaining fields for diagnostics/plots.

In the current dataset:

- Trading-day observations: **245**
- Price date range: **2025-01-13 → 2026-01-02**
- Fields: Date, Open, High, Low, Close, Adj Close, Volume (plus Dividends and Stock Splits)

The date range is chosen to match, as closely as possible, the sentiment observation window to enable aligned modelling.
Figure 2 shows the evolution of NVDA stock price for the sampled date range.

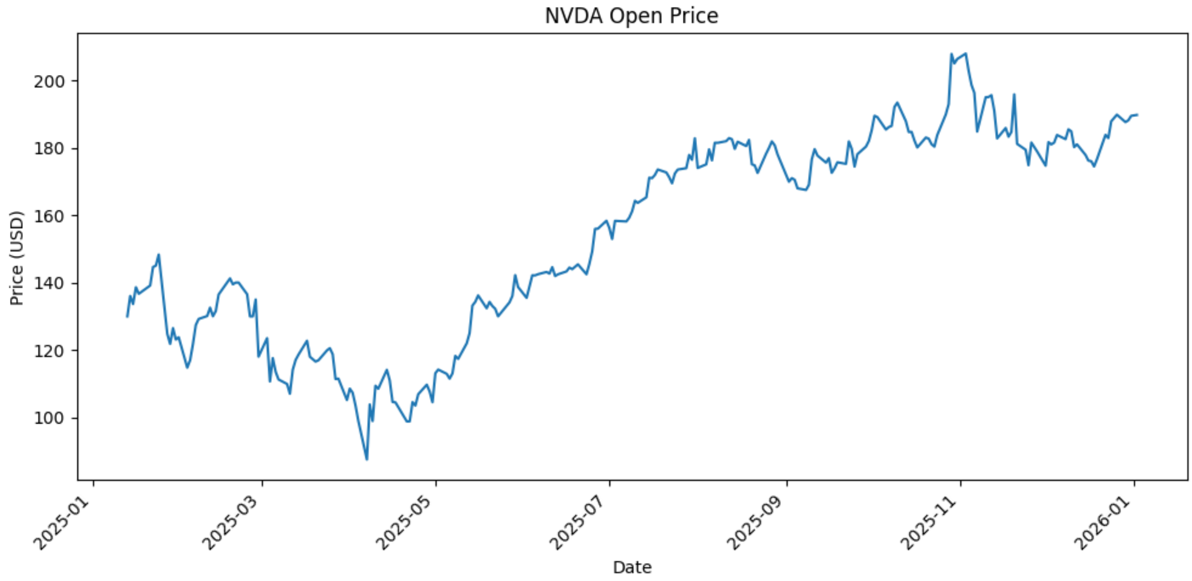

Figure 2: Evolution of stock price of NVDA

3.3 Sentiment Extraction via Large Language Models

Sentiment analysis, frequently referred to as opinion mining or opinion analysis, is a prominent subfield of Natural Language Processing (NLP) dedicated to the automatic extraction and evaluation of sentiments and viewpoints from textual data. The exponential growth of the Internet has led to a massive influx of unstructured online commentary, making the manual analysis of public opinion practically impossible. Consequently, SA provides an efficient, rapid, and cost-effective mechanism to capture the psychological states of commentators, thereby aiding in decision-making and public opinion supervision across various industries. Furthermore, SA is recognized as a foundational component in the advancement of artificial intelligence, playing a critical role in human-machine interaction systems and modern Large Language Models (LLMs) (Zhang et al., 2024).

At its core, the evaluation of sentiment involves categorizing text into distinct emotional polarities. Most commonly, sentiments are classified into three primary broad categories: positive, neutral, and negative. However, deeper linguistic evaluations can further subdivide these polarities into more fine-grained emotional states, such as joy, trust, anticipation, surprise, anger, fear, sadness, and disgust.

Consider the hypothetical financial headline shown in figure 3:

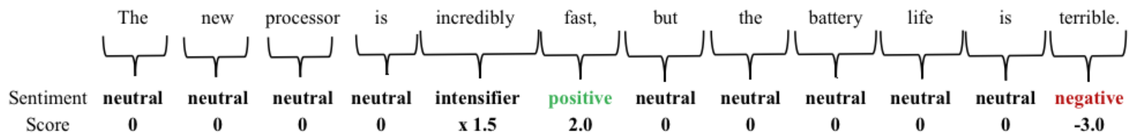

Figure number 3: Example of a token based sentiment evaluation of a financial headline.

A lexicon algorithm would map base valences to the relevant tokens (e.g., *fast* = +2.0, *terrible* = -3.0). The intensifier “*incredibly”* scales the subsequent token's value (+2.0 × 1.5 = +3.0). The mathematical aggregation yields a sum of 0.0, resulting in a strictly neutral compound score despite the presence of strong emotional extremes. While this lexicon-based pipeline is computationally efficient and highly interpretable, its reliance on static dictionary lookups limits its ability to understand deep context, complex financial jargon, or domain-specific semantic shifts. Specifically, Nandwani and Verma (2021) explain that lexicon-based methods rely on "keyword-based search" and have "drawbacks" such as failing to capture sarcasm, handling multiple emotions, and struggling with "lexical and syntactical ambiguity". This fundamental limitation is precisely why Large Language Models (LLMs) have become indispensable for modern sentiment extraction; by leveraging deep contextual embeddings, LLMs evaluate sentiment based on the entire sequence of the text rather than isolated keywords, allowing them to resolve the very ambiguities that lead lexicon-based tools to misinterpret sentiment.

As stated earlier, classical sentiment analysis methods typically output discrete labels (positive/neutral/negative). While useful, discrete labels can be too coarse for financial forecasting, where small shifts in market tone can matter. To capture *intensity* and produce a smoother time series, this thesis adopts a continuous sentiment score.

Large language models can be used as general-purpose semantic evaluators capable of inferring implicit sentiment from short, context-limited descriptions. In this work, the LLM is treated as a scoring function:

$$\text{LLMScore} : \text{text} \mapsto \bar{s} \in [0,1] \qquad (21)$$

where $\bar{s} = 0$ indicates strongly negative tone, $\bar{s} = 1$ indicates strongly positive tone, and intermediate values capture varying intensity.

Figures 4 and 5 show the evolution of the extracted sentiment scores of the NVDA stock for the sampled date range and the per-article distribution score respectively.

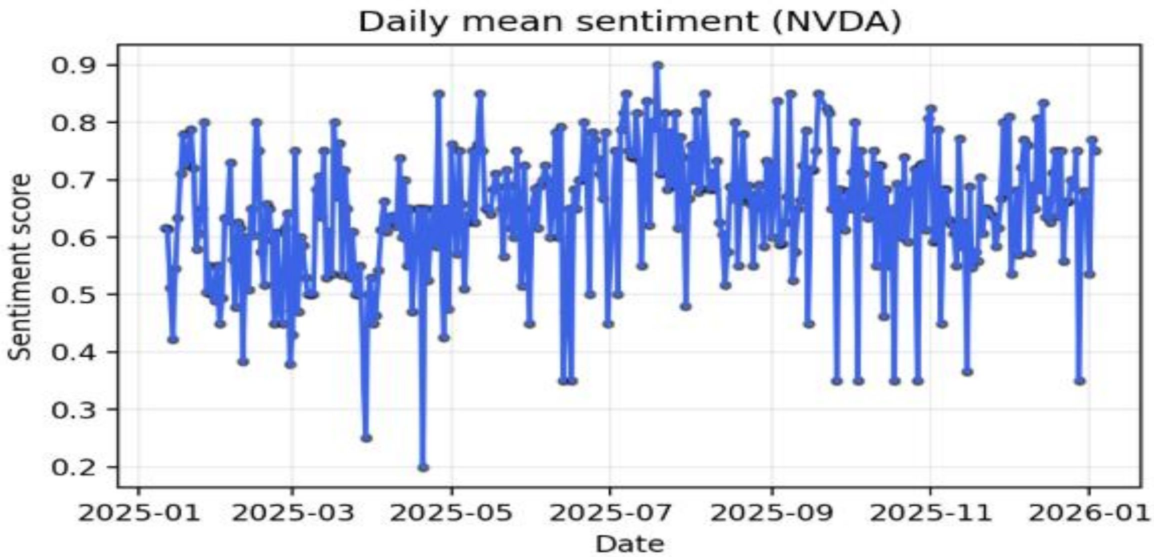

Figure 4: Evolution of sentiment scores

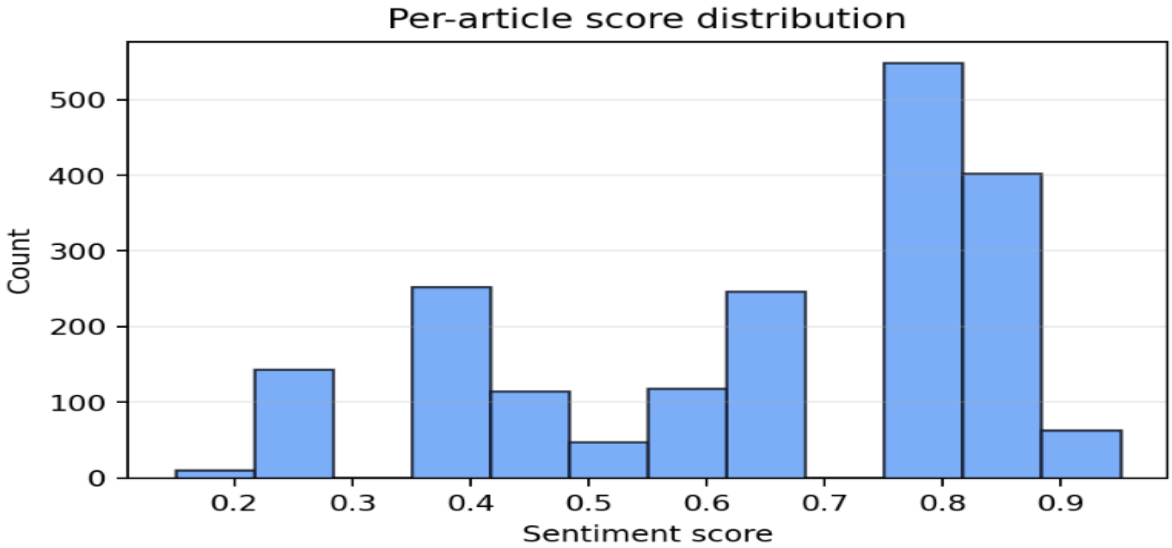

Figure 5: Per-article score distribution

3.3.1 *Scoring protocol and quality controls*

For each news description, the LLM is prompted to produce a single numeric sentiment value on a fixed scale. To improve consistency, the prompt specifies:

- the target company context (NVDA),
- the sentiment scale and its interpretation,
- instructions to output only a numeric value, and
- handling of ambiguous/neutral language.

Basic quality controls are applied during preprocessing:

- enforce numeric parsing and valid range constraints,
- drop articles with missing or malformed outputs,
- store both the raw label (when provided by the news API) and the continuous score.

The continuous score distribution in the dataset is summarized below (computed over scored articles):

- mean: 0.637
- standard deviation: 0.203
- min / max: 0.15 / 0.95

These summary statistics are useful for understanding the baseline tone and variability in the collected news.

3.3.2 *Agentic implementation (LangGraph orchestration)*

In addition to the conceptual scoring procedure described above, the sentiment extraction was implemented as an agentic workflow to support (i) robust schema-constrained outputs, and (ii) scalable scoring over large batches of article descriptions.
Concretely, a LangGraph state machine is used to orchestrate the scoring pipeline:

- **Batch ingestion:** the dataset is processed in batches (e.g., 30 descriptions per batch).
- **Fan-out execution:** each article description is routed to an LLM-scoring node (parallelized with a bounded concurrency limit).
- **Structured output enforcement:** the LLM is queried with a prompt that defines a *[0,1]*[0,1] sentiment scale and returns a structured response containing a single float sentiment score.
- **Persistence:** scores are written back to the original JSON records as a sentiment_score field and saved to disk.

Each score is a single float in *[0,1].* Malformed outputs are prevented by structured parsing.
Figure 6 provides a schematic view of the processing flow used to score large collections of articles.

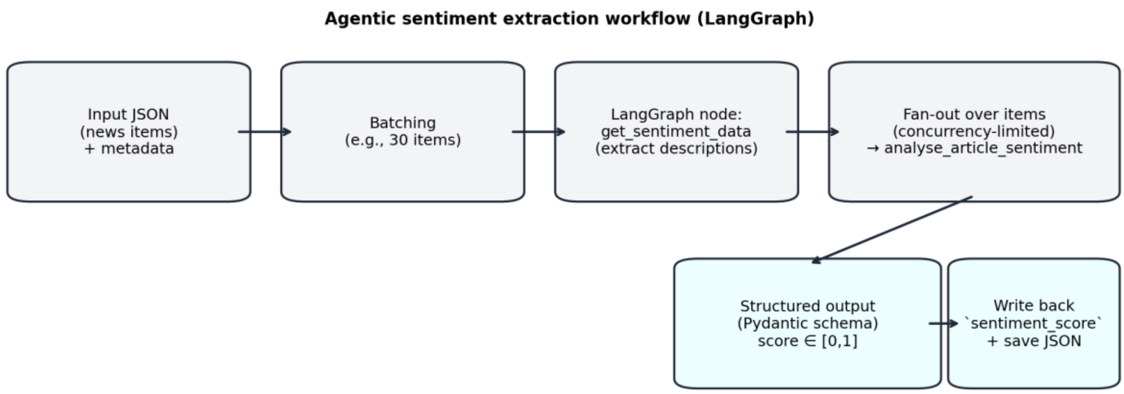

Figure 6: Agentic sentiment extraction workflow

This design reduces failures caused by malformed free-form responses and improves reproducibility of downstream experiments. In the current implementation, the chat model is run with temperature set to 0 to minimize sampling variability during scoring.

3.4 Proposed Forecasting Architecture and Experimental Protocol

This section describes the modelling pipeline that is implemented in the project notebooks.

3.4.1 *Sentiment modelling with a Neural SDE*

Let s_t denote the latent sentiment state. We model its continuous-time evolution as:

$$ds_t = f_s(s_t, t)dt + g_s(s_t, t)dW_t^s \quad (22)$$

where f_s and g_s are neural networks representing sentiment drift and diffusion.

The sentiment model is trained on the first 80% of the sentiment time series (chronological split). Evaluation is performed using a **one-step-ahead walk-forward** protocol on the remaining 20%:

- at each test step, the model forecasts the next day,
- uncertainty is obtained by Monte Carlo path sampling,
- the latent state is updated using the observed value (teacher forcing) to prevent compounding error during evaluation.

In addition to producing forecasts, the sentiment model exports leakage-safe derived features that summarize the inferred sentiment dynamics, such as:

In addition to producing forecasts, the sentiment model exports leakage-safe derived features that summarize the inferred sentiment dynamics, such as:

- predicted mean μ_s^t (trend proxy),
- predicted standard deviation σ_s^t (volatility proxy),
- optional derived quantities (e.g., interval width, momentum-like changes).

To maximize usable aligned rows for the price model without violating the no-leakage rule, a **hybrid feature table** is constructed:

- on sentiment-train dates: features are generated from in-sample fitted values,
- on sentiment-test dates: features are generated from out-of-sample walk-forward forecasts,
- the two segments are concatenated into a full-range feature table that preserves the train/test boundary.

After training the sentiment NSDE on the sentiment-train split, we use the trained model to generate a distribution of latent sentiment trajectories via Monte Carlo path sampling. On the training segment, we obtain in-sample fitted features by simulating the trained NSDE from the first training observation s_{t_0} across the training time grid and computing, at each time point, the empirical predictive mean and dispersion across sampled paths.

Concretely, the exported proxies are computed as:
(i) a trend proxy μ_t^s given by the Monte Carlo means,
(ii) a volatility proxy σ_t^s given by the Monte Carlo standard deviation, and
(iii) a 95% interval-width proxy w_t^s given by the difference between the 97.5% and 2.5% empirical quantiles.

On the test segment, the same quantities are produced in a one-step-ahead walk-forward manner: at step t we forecast s_t from information up to $t-1$, then update the state with the observed *st* (teacher forcing) before forecasting s_{t+1} .

This procedure is considered data-leakage-safe for the downstream price model as long as (a) the sentiment NSDE itself is trained using sentiment-train data only, and (b) sentiment features used on price-test rows are derived from the out-of-sample walk-forward forecasts (not from a full-sample fit that implicitly uses future sentiment observations). Using the trained sentiment model to generate in-sample fitted features on the sentiment-train segment does not leak information from the future test period; it is analogous to using in-sample fitted values from a model estimated on training data.

3.4.2 Price-level modelling with a Neural SDE (baseline)

Let y_t denote the observed NVDA Open price. Since price levels are non-stationary and on a larger scale than sentiment, we standardize using training statistics only:

$$z_t = \frac{y_t - \mu_{\text{train}}}{\sigma_{\text{train}}} \qquad (23)$$

The baseline price model is an Itô Neural SDE on z_t :

$$dz_t = f_p(t, z_t)dt + g_p(t, z_t)dW_t^p \qquad (24)$$

The model is trained on the first 80% of trading days and evaluated with one-step-ahead walk-forward prediction on the remaining 20%.

Training losses. Besides a probabilistic loss based on Gaussian negative log-likelihood (NLL), the project also uses a hybrid objective that encourages accurate mean tracking while retaining uncertainty learning:

$$L = 0.7\text{MSE}(\widehat{\mu_t}, z_t) + 0.3\text{NLL}\left(\widehat{\mu_t}, z_t, \widehat{\sigma_t^2}\right) \qquad (25)$$

3.4.3 Classical and deep-learning baselines (discrete baselines)

To contextualize the continuous-time price NSDE against established forecasting methods, we additionally implement two standard baselines and evaluate them under an identical protocol: the same NVDA Open series, the same chronological 196/49 train/test split, and the same one-step-ahead walk-forward evaluation used for the price NSDE. This ensures the comparison is apples-to-apples rather than dependent on differing data or evaluation choices.
Both baselines are also evaluated with a lagged sentiment feature ($L = 3$), the lag that was most favourable in the NSDE lag-sensitivity analysis (Section 4.5), so that any advantage from sentiment is given a fair chance in the classical/deep setting as well.

ARIMA / ARIMAX (classical statistical baseline). We fit an ARIMA model to the training prices, selecting the $(p|d|q)$ order by minimizing the Akaike Information Criterion (AIC). Forecasts are produced by a rolling one-step-ahead procedure: at each test step the model issues a one-day forecast, from which we take the predicted mean and the forecast standard error to form a Gaussian 95% interval; the model state is then advanced with the realized observation (teacher forcing), mirroring the walk-forward protocol used elsewhere. The sentiment-augmented variant (ARIMAX) adds the lagged daily sentiment score ($L = 3$) as an exogenous regressor, standardized using training statistics only.

LSTM (deep sequence baseline). We train a heteroscedastic LSTM that outputs, at each step, both a predictive mean and a positive predictive variance (via a Softplus variance head), trained with a Gaussian negative log-likelihood so that its intervals are learned rather than assumed homoskedastic. Inputs are standardized with training statistics only, and the same one-step-ahead walk-forward evaluation is applied. The sentiment-augmented variant appends the lag-3 standardized sentiment feature as an additional input channel.

Because the standard error of a directional hit-rate over the short 49-day test window is large, giving a 95% no-skill band of roughly $[36\%|64\%]$), directional-accuracy differences of a few percentage points between these baselines and the NSDE are interpreted with corresponding caution. The comparison results are reported in Section 4.8.

3.4.4 Sentiment-augmented price models (exogenous inputs and lags)

The central hypothesis is that sentiment dynamics can improve price forecasting when injected as exogenous inputs to the drift and/or diffusion components.

Let x_t denote a vector of sentiment-derived features. In the simplest case:

$$x_t = [\mu_t^s, \sigma_t^s] \quad (26)$$

Because news sentiment may influence *future* price changes more strongly than same-session prices, we consider lagged features $x_{t-L}\ for\ L \in \{1,2,3\}$. After aligning sentiment feature dates to trading dates, the price model is extended to:

$$dz_t = f_p(t, z_t, x_{t-L})dt + g_p(t, z_t, x_{t-L})dW_t^p \quad (27)$$

The thesis evaluates a set of controlled variations that isolate which sentiment component influences which SDE component:

- **Variation 0 (baseline):** no sentiment input.

$$dz_t = f_p(t, z_t)dt + g_p(t, z_t)dW_t^p \quad (28)$$

- **Variation 1:** sentiment trend μ^s enters drift only.

$$dz_t = f_p(t, z_t, \mu_{t-L}^s)dt + g_p(t, z_t)dW_t^p \quad (29)$$

- **Variation 2:** sentiment trend μ^s enters both drift and diffusion.

$$dz_t = f_p(t, z_t, \mu_{t-L}^s)dt + g_p(t, z_t, \mu_{t-L}^s)dW_t^p \quad (30)$$

- **Variation 3:** sentiment volatility σ^s enters diffusion only.

$$dz_t = f_p(t, z_t)dt + g_p(t, z_t, \sigma_{t-L}^s)dW_t^p \quad (31)$$

- **Variation 4:** μ^s enters drift and σ^s enters diffusion.

$$dz_t = f_p(t, z_t, \mu_{t-L}^s)dt + g_p(t, z_t, \sigma_{t-L}^s)dW_t^p \quad (32)$$

- **Variation 5:** full sentiment vector enters both drift and diffusion.

$$dz_t = f_p\left(t, z_t, x_{t-L}^{\text{full}}\right)dt + g_p\left(t, z_t, x_{t-L}^{\text{full}}\right)dW_t^p \quad (33)$$

All variations are evaluated under the same walk-forward protocol and the same train/test split. Alignment is performed via an inner join on trading dates, and missing-feature rows created by lagging are dropped consistently across baseline and variants to ensure fair comparisons.

Empirically, sentiment signals in financial text are often **episodic** rather than persistent: unconditional correlation between sentiment and returns can be weak, while stronger dependence may emerge around high-attention periods (e.g., bursts in message volume) when information is rapidly incorporated into prices. This motivates both (i) testing multiple lags (to allow for delayed reactions), and (ii) evaluating sentiment-augmented models under a leakage-safe forecasting protocol rather than relying on contemporaneous correlation alone (cf. the event-based findings of Ranco et al., 2015).

3.4.5 *Simulation and uncertainty estimation*

To generate predictions from a Neural SDE, the SDE is simulated forward in time using a numerical method such as Euler–Maruyama. Because each simulation corresponds to one stochastic path, uncertainty can be estimated via Monte Carlo sampling.

Given *N* simulated paths producing samples $\left\{\hat{x}_t^{(j)}\right\}_{j=1}^{N}$ at a forecast time, the predictive mean and variance are estimated as:

$$\widehat{\mu_t} = \frac{1}{N}\sum_{j=1}^{N} \widehat{x_t^{(j)}} , \quad (34)$$

$$\widehat{\sigma_t^2} = \frac{1}{N-1}\sum_{j=1}^{N}\left(\widehat{x_t^{(j)}} - \widehat{\mu_t}\right)^2 \quad (35)$$

Prediction intervals (e.g., 95%) can be computed empirically from Monte Carlo quantiles.

3.5 Evaluation Metrics

The performance of each model is assessed using both point forecast and probabilistic metrics.

3.5.1 Point forecast metrics

Let y_t be the true target and $\widehat{y_t}$ the point prediction (e.g., predictive mean). For a test set of size *T*:

- **Mean Absolute Error (MAE)**

$$\text{MAE} = \frac{1}{T}\sum_{t=1}^{T}|y_t - \widehat{y_t}| \quad (36)$$

- ***Root Mean Squared Error (RMSE)***

$$\text{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t - \widehat{y_t})^2} \qquad (37)$$

- ***Coefficient of Determination (R^2)***

$$R^2 = 1 - \frac{\sum_{t=1}^{T}(y_t - \widehat{y_t})^2}{\sum_{t=1}^{T}(y_t - \overline{y})^2} \qquad (38)$$

where $\overline{y}$ is the mean of y_t over the test set.

- **Directional accuracy** (for price changes): the fraction of steps where the predicted direction matches the realized direction.

3.5.2 Probabilistic metrics

Because Neural SDEs provide predictive distributions (via Monte Carlo sampling), we also evaluate:

- **Coverage of 95% prediction intervals:** the fraction of test points for which y_t falls within the model's 95% interval.
- **Average interval width:** a sharpness measure; lower is better for a fixed coverage.

Optionally, when the predictive distribution is approximated as Gaussian with mean $\widehat{\mu_t}$ and variance $\hat{\sigma}_t^2$, the **negative log-likelihood (NLL)** can be used as a proper scoring rule.

3.6 Long horizon and multi-step horizon forecasting

In addition to the standard one-step-ahead walk-forward evaluation, the thesis also reports two horizon-based diagnostics to understand how forecast quality changes as the prediction window grows.

The horizon-based evaluations are included for two reasons. First, they reveal whether a model that performs well at one day ahead remains stable when used operationally over longer windows. Second, they provide a direct check on uncertainty calibration: if the interval width grows too slowly, the model is overconfident; if it grows too quickly, the model becomes too diffused and loses practical sharpness.

In the Neural SDE setting, these diagnostics are especially informative because the diffusion term controls how uncertainty propagates over time. A miscalibrated diffusion can look acceptable under walk-forward evaluation but fail dramatically in long-horizon or multi-step forecasting.

3.6.1 Long-horizon free-running forecast

The long-horizon setting integrates the Neural SDE once over the entire test period without resetting to the observed state at each day. Starting from the last training observation, the model produces a single Monte Carlo forecast trajectory across the full test window:

$$\hat{z}_{t_1:t_T} = \text{SDEint}(z_{t_0}, t_0, \ldots, t_T),$$

where t_0 is the final training time and $t_1, \dots, t_T$ are the test times. This evaluation is more demanding than walk-forward forecasting because any drift bias or diffusion miscalibration compounds over time. It is therefore useful for diagnosing whether the model has learned stable long-run dynamics or only short-step local behavior.

The long-horizon evaluation is applied both to the price-only baseline and to the sentiment-augmented variants. For the sentiment case, the exogenous inputs are held fixed using the leakage-safe aligned sentiment features already constructed in Section 3.5.3.

3.6.2 Multi-step horizon evaluation

The multi-step setting measures forecast quality at fixed horizons $h \in \{1,3,5,10,15,20\}$ days ahead. For each valid starting point in the test window, the model is integrated from the current observed state to the target horizon and compared against the realized price at that future date:

$$\hat{z}_{t+h} = \mathrm{SDEint}(z_t, t, t+h).$$

This protocol sits between one-step walk-forward forecasting and full free-running simulation. It isolates how errors accumulate with horizon while still evaluating a direct h-day-ahead forecast rather than repeatedly resetting to the ground truth.

For sentiment-augmented variants, the same lagged sentiment features are supplied at the target horizon using the leakage-safe aligned exogenous table. The reported metrics include RMSE, MAE, R^2, empirical 95% coverage, and average interval width as functions of forecast horizon.

3.7 Implementation Notes (reproducibility)

All experiments are implemented in Python. Neural models are implemented in PyTorch, with continuous-time integration provided by standard differential equation solvers and Monte Carlo simulation. Key dependencies include `torch`, `torchdiffeq` (for Neural ODE integration), `pandas`/`numpy` for preprocessing, and `yfinance` for price data acquisition. The full source code and notebooks are available in the project repository at https://github.com/johnpararas/sentiment---stock--analysis .
The core experimental notebooks are:

- Sentiment NSDE: `src/NSDE_sentiment_analysis.ipynb`
- Price-level NSDE baseline + sentiment-augmented variants: `src/NSDE_price_levels_analysis.ipynb`
- Returns NSDE: `src/NSDE_returns_analysis.ipynb`

For leakage-safe comparisons, all standardization parameters are computed on training data only and walk-forward evaluation is performed chronologically.

4. Results

This chapter reports the empirical results for the **price-level Neural SDE (Price NSDE)** experiments. We evaluate one-step-ahead, walk-forward forecasting performance on NVDA daily **Open** prices under a leakage-safe protocol. The goal is to quantify (i) how a baseline continuous-time price model performs relative to a naive benchmark, and (ii) whether injecting sentiment-derived signals improves point accuracy and/or probabilistic calibration.

Experimental setup :

- **Target:** NVDA daily Open price y_t.
- **Standardization:** price is standardized using training statistics only, $z_t = \frac{y_t - \mu_{\text{train}}}{\sigma_{\text{train}}}$,and metrics are reported back in **price space**.
- **Protocol:** chronological 80/20 split, with **one-step-ahead walk-forward** evaluation on the test segment.
- **Uncertainty:** predictive intervals are computed from Monte Carlo path sampling; we report 95% coverage and average interval width.
- **Sentiment inputs:** sentiment-derived features (e.g., predicted trend μ^s and volatility σ^s) enter drift and/or diffusion, with lag $L \in \{1,2,3\}$ to ensure leakage-safe alignment.

4.1 Sentiment NODE implementation

Before introducing the sentiment NSDE, we evaluate a Neural ODE (NODE) baseline to understand how a deterministic continuous-time model behaves on the sentiment series. The NODE uses the same train/test split and one-step-ahead walk-forward protocol, but the dynamics are purely deterministic (no diffusion term). This makes the NODE a useful diagnostic: it should capture smooth, low-frequency trends, yet it cannot represent stochastic fluctuations that are prominent in news-driven sentiment.

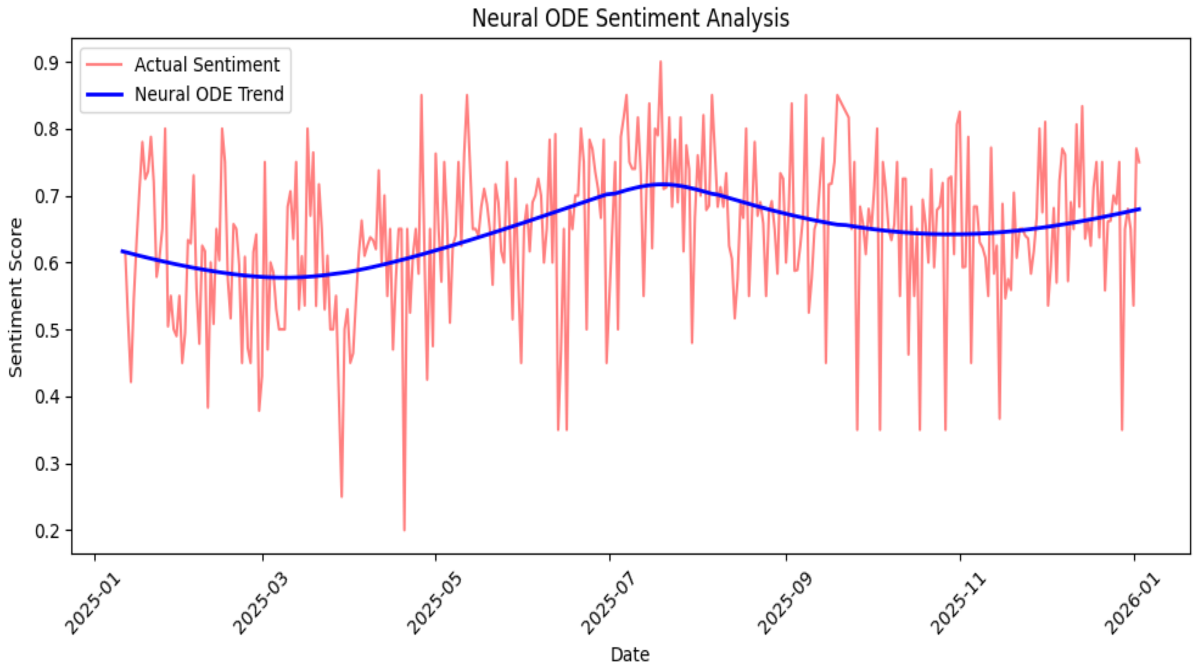

Figure 7. Neural ODE fit for the sentiment series: in-sample training fit and out-of-sample one-step-ahead test forecasts.

The NODE fit typically tracks broad sentiment drift but struggles to keep pace with the short, irregular jumps caused by episodic news bursts. Figure 7 shows that the deterministic trajectory lags abrupt reversals and systematically underestimates variability, visible in the flattening of predicted paths during high-volatility intervals. This behavior is expected: a deterministic flow cannot generate the random shocks observed in real sentiment dynamics.

To visualize the learned dynamics directly, we plot the NODE vector field in time and the implied trajectory. Figure 8 shows a smooth, low-curvature flow, indicating that the model prefers gradual changes in sentiment rather than sudden jumps. The blue curve represents the trajectory generated by integrating the learned differential equation, while the red points correspond to the observed sentiment data. The trajectory captures the underlying trend while filtering short-term fluctuations and measurement noise. Integrating this learned vector field produces the smooth trajectory shown by the blue curve, which closely follows the observed sentiment data while filtering short-term noise. This helps explain why the deterministic NODE smooths out bursty news reactions. Such continuous sentiment representations can subsequently serve as informative features for downstream stock price forecasting models.

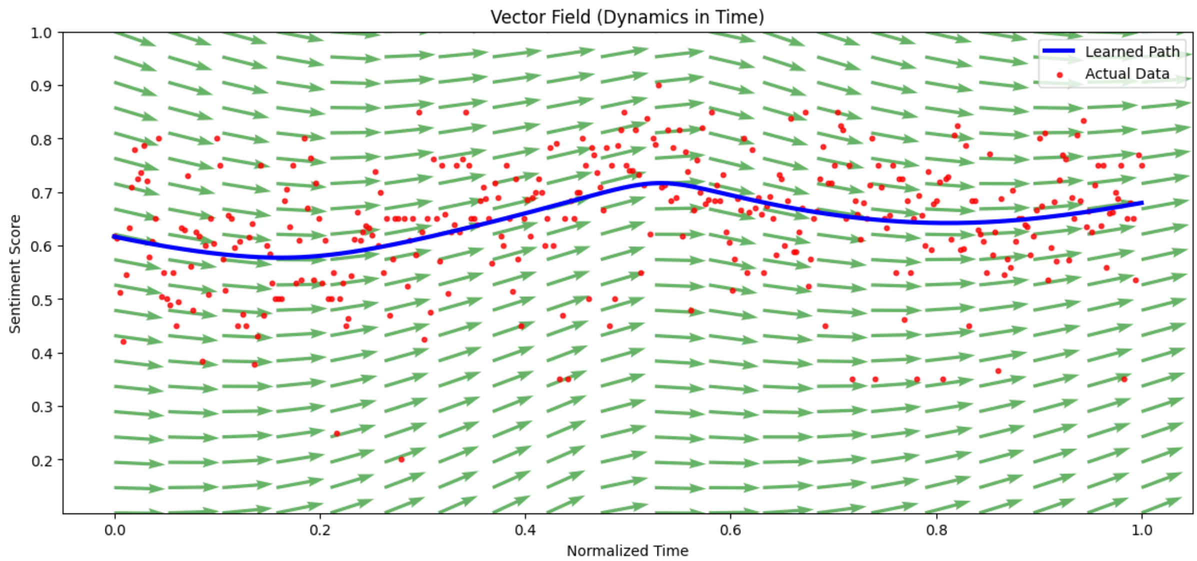

Figure 8. NODE vector field over time with the learned trajectory and observed sentiment points.

We also examine the 3D extended phase portrait (time, sentiment level, derivative). Figure 9 shows a smooth arc with slow drift in the derivative dimension and no stochastic dispersion, highlighting the deterministic nature of the NODE. In other words, the model learns a single, smooth path through phase space rather than a distribution of possible paths, which is insufficient for capturing the observed variability. Around sentiment values near $s \approx 0.66$, the derivative approaches zero, indicating the existence of a locally stable operating region where sentiment changes slowly. This may represent a market consensus state in which news arrivals are insufficient to induce large shifts in investor expectations. When sentiment rises from lower values, $\dot{s}$ increases rapidly and remains positive for an extended period before gradually decreasing. Similarly, declining sentiment exhibits a period of negative momentum before recovering. This suggests that the Neural ODE has learned persistence effects commonly observed in financial markets.

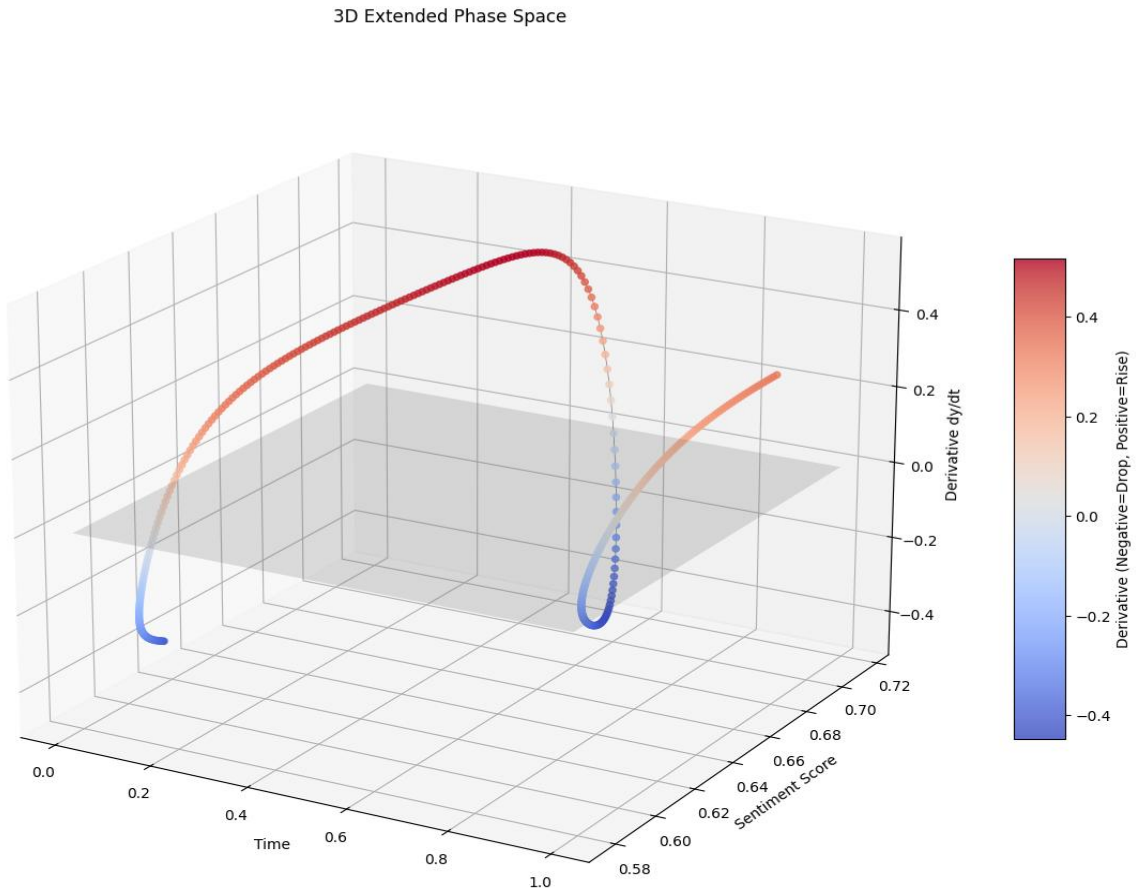

Figure 9. 3D extended phase space of the NODE trajectory, showing smooth deterministic dynamics.

To make this failure mode explicit, we also visualize the residual behavior and error dynamics. Figures 10 and 11 show clustered deviations rather than white-noise errors, consistent with unmodeled stochasticity. This motivates the move to an SDE-based model that can represent both the mean trend and the time-varying uncertainty.

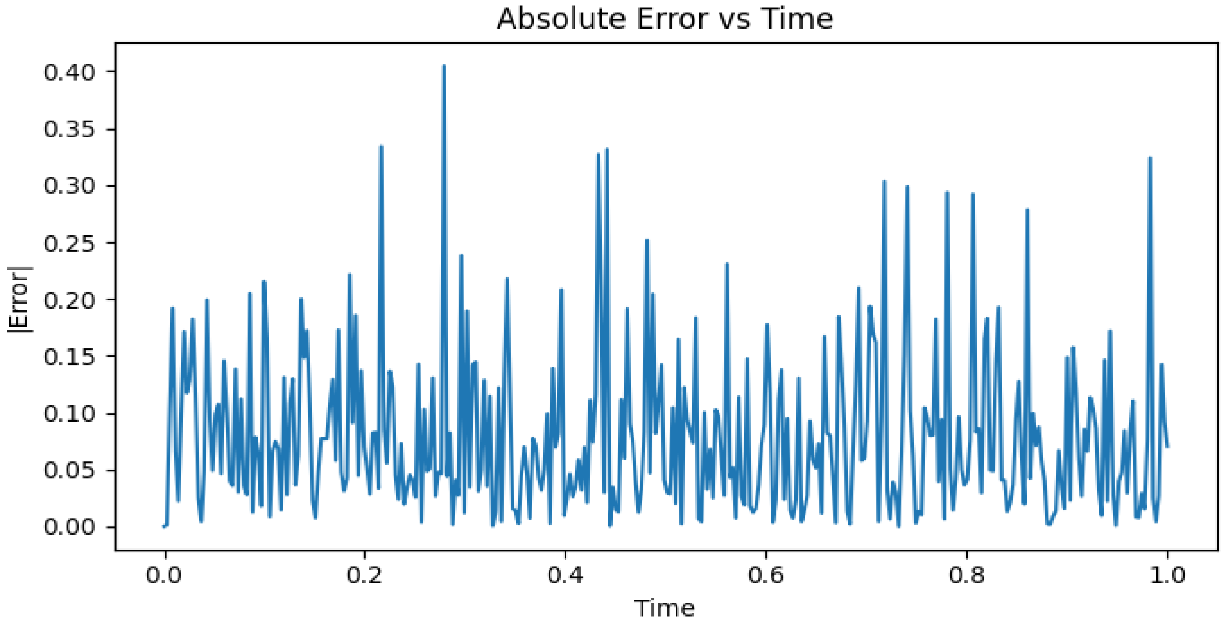

Figure 10. Neural ODE error behavior for sentiment: clustered deviations highlight unmodeled stochastic noise.

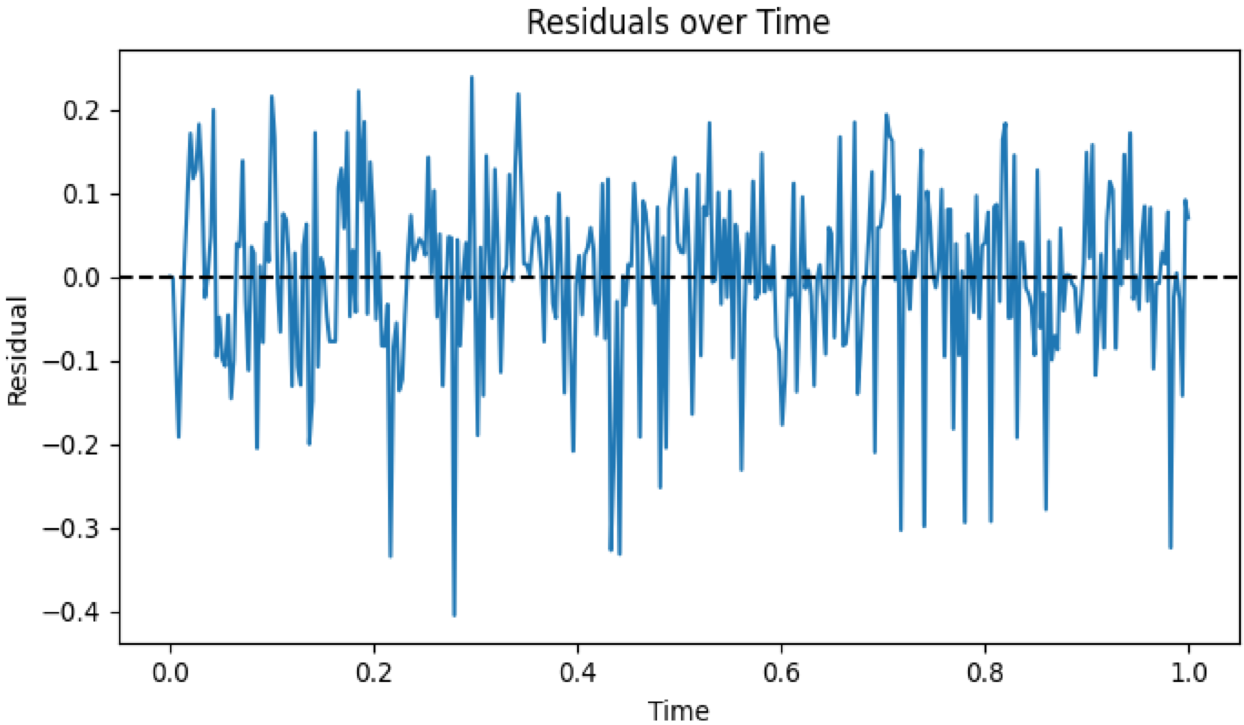

Figure 11. Neural ODE residual behavior for sentiment: clustered deviations highlight unmodeled stochastic noise.

The deterministic NODE captures long-run sentiment direction but fails to capture the irregular, bursty volatility in the series. This leads to systematic underfitting during high-attention periods and motivates the use of a Neural SDE with an explicit diffusion component.

4.1.2 Sentiment NSDE

Before evaluating price models, we summarize the sentiment NSDE fit that generates the sentiment trend/volatility proxies used downstream. The model is trained on the first 80% of the sentiment series and evaluated with one-step-ahead walk-forward forecasts on the remaining 20%.

Figure 12 shows the training and test fit for the NSDE applied on the sentiment-score time series.

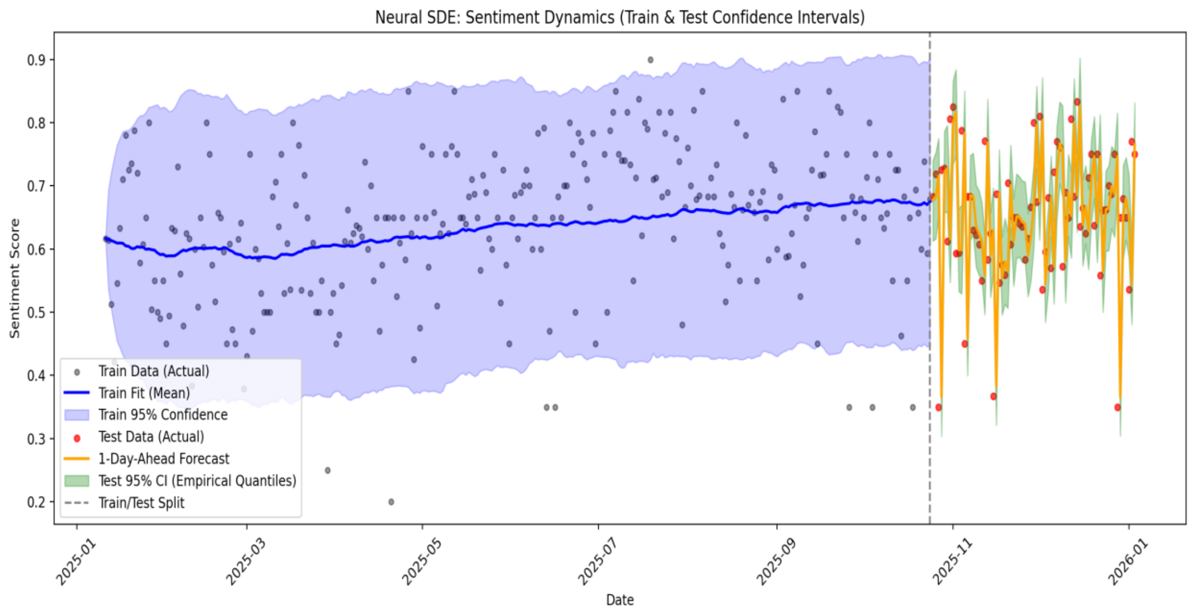

Figure 12. Neural SDE fit for the sentiment series: in-sample training fit and out-of-sample one-step-ahead test forecasts with 95% intervals.

The sentiment NSDE provides modest improvements over the naive baseline in point accuracy and achieves relatively high directional accuracy, indicating it captures short-term sentiment direction changes. Interval coverage is below 50%, suggesting under-dispersed uncertainty estimates; nevertheless, the learned trend and volatility proxies are sufficiently stable to serve as exogenous inputs in the price experiments that follow.

Table 2 shows the resulting test-performance metrics of the sentiment NSDE

Baseline (Naive 'Tomorrow = Today')	***Neural SDE***
Naive RMSE: 0.1524	RMSE: 0.1486
Naive MAE: 0.1121	MAE: 0.1095
	Directional Accuracy: 70.42%
	95% CI Coverage: 46.48%

Table 2: Sentiment NSDE test metrics vs Baseline

4.2 Baseline price NSDE assessment

We first evaluate the baseline Price NSDE (Variation 0) against a naive benchmark:

- **Naive benchmark:** “tomorrow equals today $\widehat{y_{t+1}} = y_t$

Table 3 summarizes the results.

Model	RMSE	MAE	R^2	Directional Acc.	95% CI Coverage	Avg CI Width
Price NSDE (Variation 0, hybrid loss)	**5.1431**	**3.7110**	**0.6620**	**55.10%**	**18.37%**	**1.6922**
Naive (tomorrow=today)	**5.1913**	**3.7302**	—	—	—	—

Table 3: Baseline price NSDE performance metrics vs Naive (tomorrow = today)

Figure 13 shows the corresponding one-step-ahead forecasts over the test window.

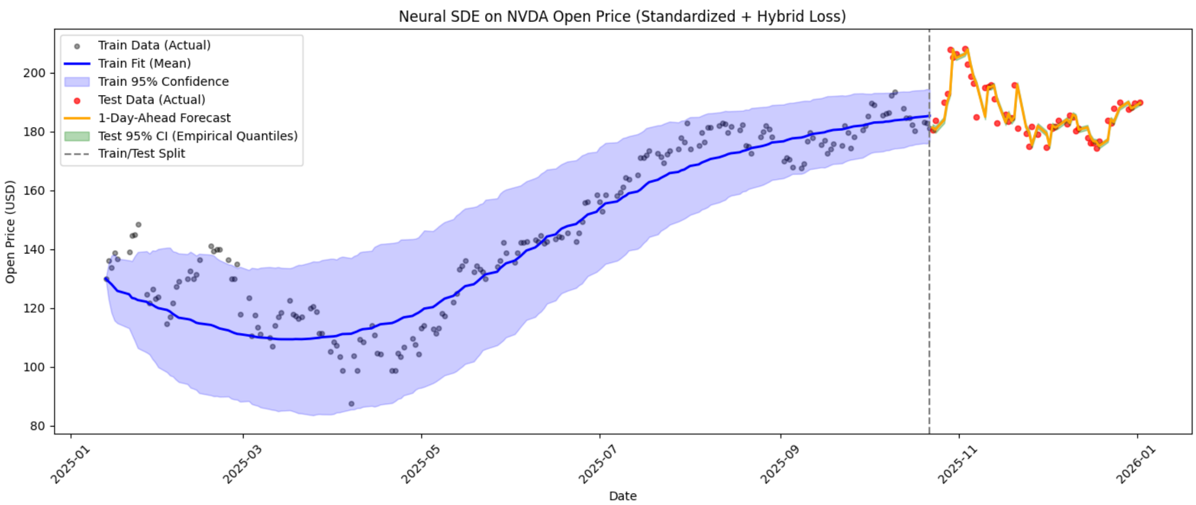

Figure 13: Neural SDE on stock price (train + test)

Observation (baseline): the baseline Price NSDE achieves slightly better point metrics than the naive benchmark. However, the 95% interval coverage is low, indicating under-dispersed uncertainty estimates in this configuration.

4.3 Loss ablation (baseline, no sentiment)

To disentangle point accuracy and calibration effects, we compare three loss configurations for the baseline model:

- **Hybrid loss:** *0.7 MSE+0.3 NLL*
- **MSE-only**
- **NLL-only**

The results of the 3 loss configurations are displayed in table 4 below.

Loss	RMSE	MAE	*R2*	Directional Acc.	95% CI Coverage	Avg CI Width
Hybrid	5.1431	3.7110	0.6620	55.10%	18.37%	1.6922
MSE-only	5.1353	3.7082	0.6630	53.06%	8.16%	0.4268
NLL-only	5.0741	3.6845	0.6710	55.10%	30.61%	3.4175

Table 4: Loss ablation (baseline, price space).

MSE-only tends to produce very narrow intervals (low coverage), while NLL-only increases dispersion and improves coverage at the cost of wider intervals. The hybrid loss provides a middle ground.

The NLL-only objective can look slightly better on point metrics. We still keep the hybrid loss as the default because it jointly constrains the mean fit (MSE) and the predictive uncertainty (NLL), which tends to stabilize training and prevents variance miscalibration. NLL-only can overfit the likelihood by trading off mean accuracy against variance (Lakshminarayanan et al. (2017)).

Figures 14-19 show the training loss curves for the baseline and Variations 1–5. These curves are useful as an optimization diagnostic (e.g., convergence and stability), but should not be interpreted as a direct measure of out-of-sample forecasting performance.

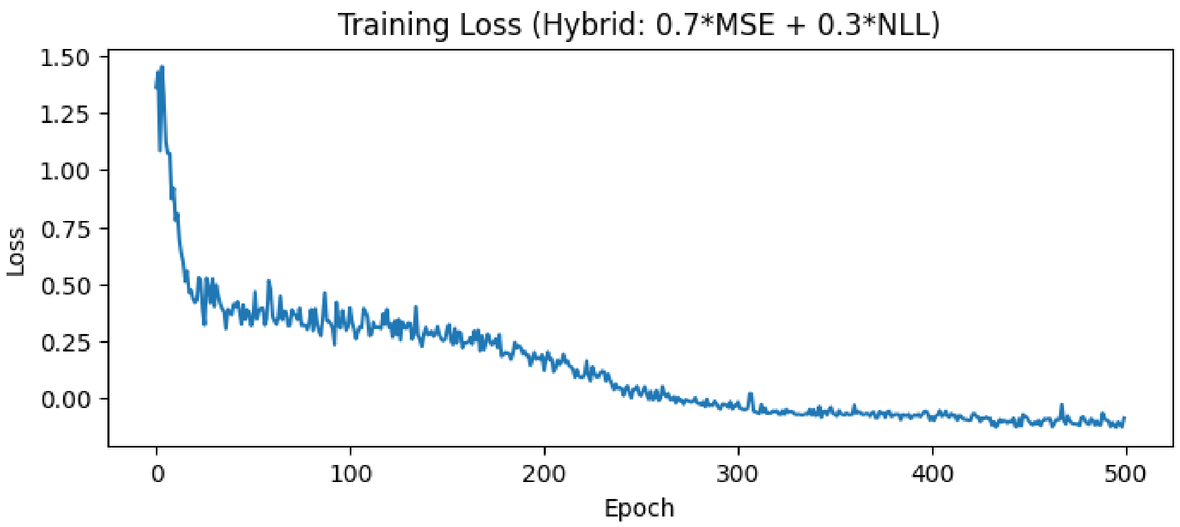

Figure 14: Training loss curve — baseline (Variation 0).

Figure 15: Training loss curve — Variation 1.

Figure 16: Training loss curve — Variation 2.

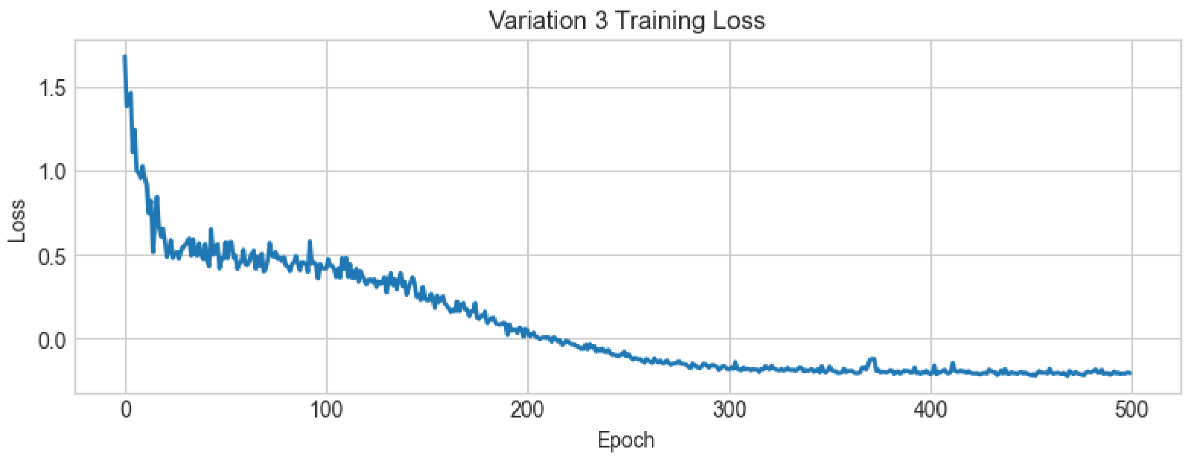

Figure 17: Training loss curve — Variation 3.

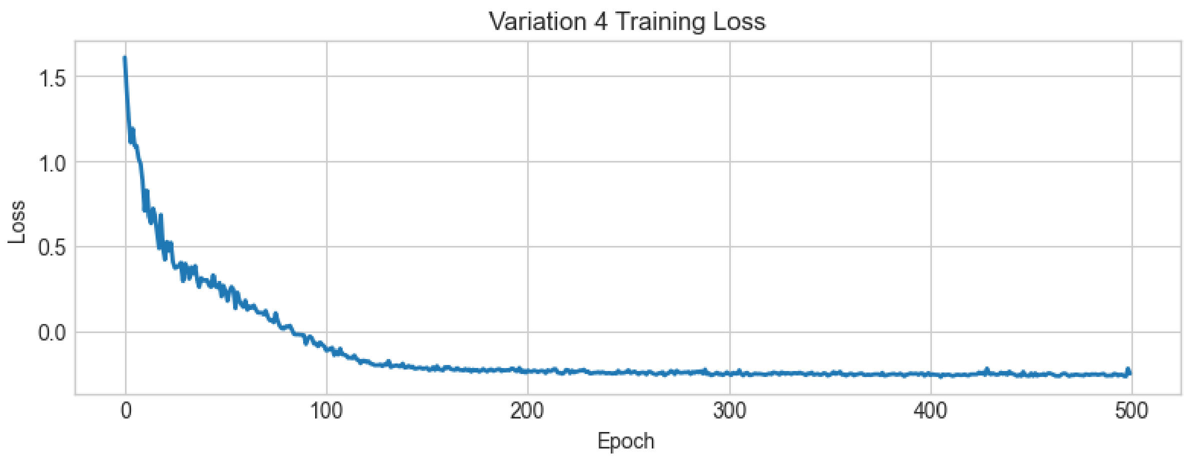

Figure 18: Training loss curve — Variation 4.

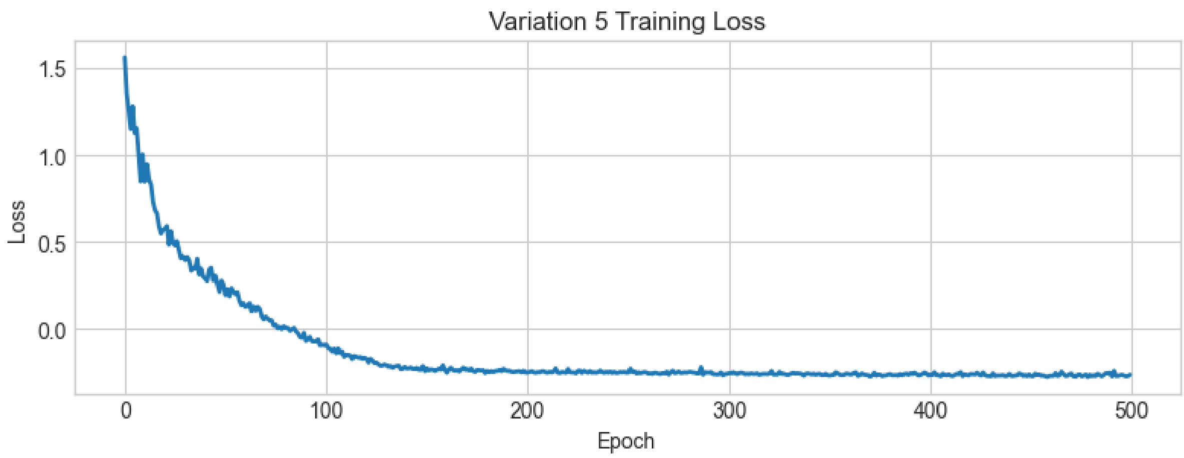

Figure 19: Training loss curve — Variation 5.

4.4 Sentiment-augmented price NSDE (Variations 1–5, lag *L=1*)

We now test whether sentiment-derived features improve price forecasting when injected into the SDE drift and/or diffusion. We focus on the controlled variations defined in Chapter 3:

In all cases, the standardized price process zt is modeled as an Itô SDE. Let W_t^p denote the Brownian motion driving the price model. For *L=1*, the sentiment features are aligned and lagged as $(\mu_{t-1}^s, \sigma_{t-1}^s)$

Exploratory correlation check (returns vs lagged sentiment proxies)

Before fitting the sentiment-augmented NSDE variants, we run a simple *descriptive* check of linear/monotone association: the Pearson and Spearman correlations between daily **Open-to-Open returns** and the lagged sentiment NSDE proxies (predicted trend, predicted volatility, and the 95% interval-width proxy), for lags $L \in \{1,2,3\}$. We report correlations in **return space** (rather than price levels) to reduce spurious effects driven by non-stationarity in price levels.

This check is included only to build intuition and is not used for model selection. In particular, the correlations should not be interpreted causally (and, strictly speaking, any exploratory summary that uses the full sample can be seen as "peeking" at the test window). The main, leakage-safe evaluation remains the walk-forward forecasting protocol reported in Table 5 of section 4.4.6.

Figures 20 and 21 below depict the correlations in the returns space.

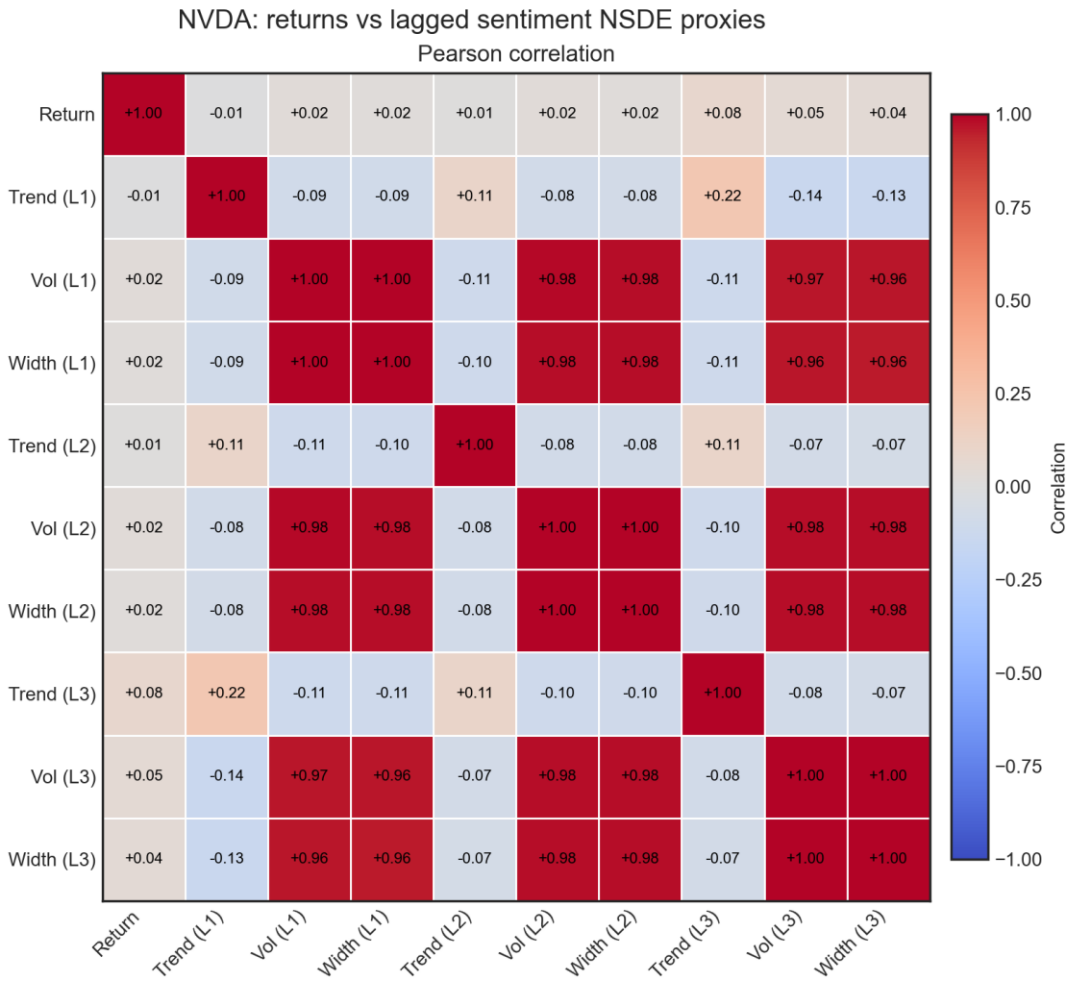

Figure 20. Pearson correlations between NVDA daily Open-to-Open returns and lagged sentiment NSDE proxies.

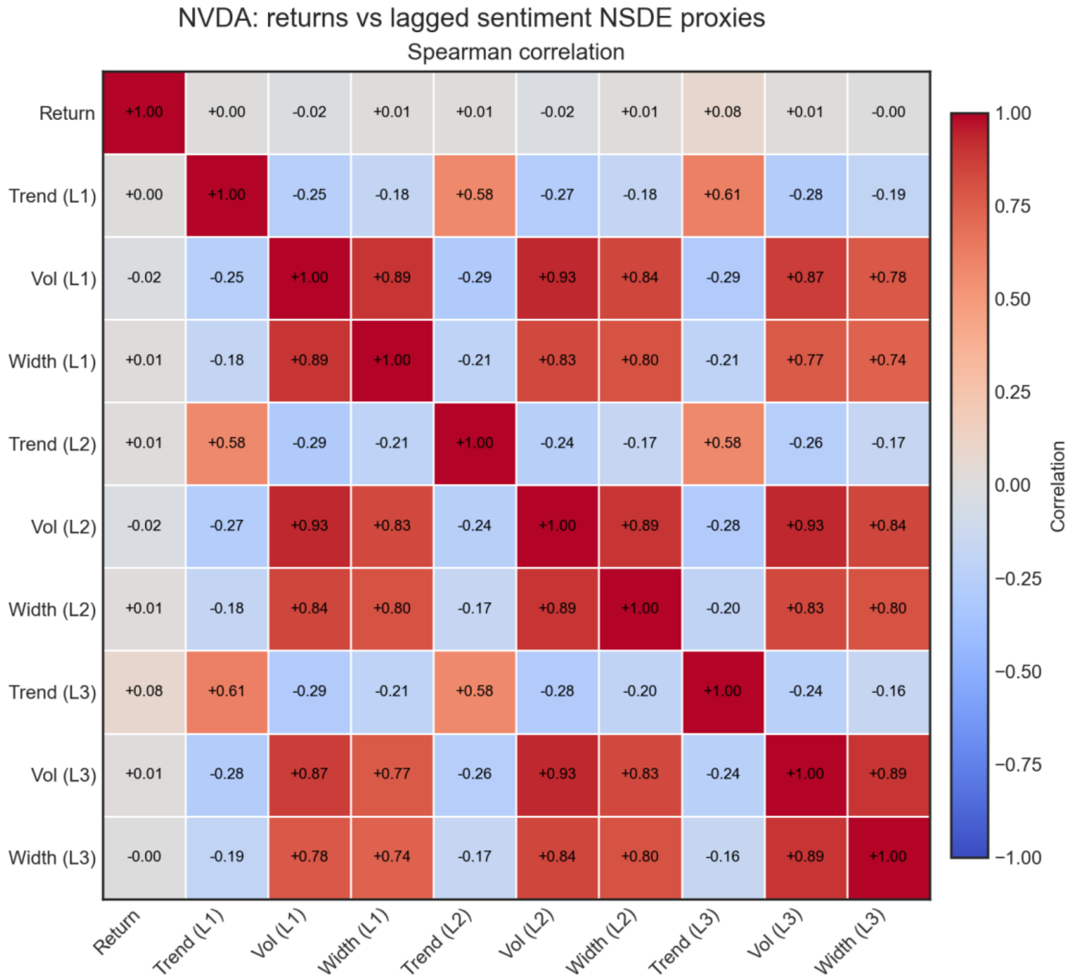

Figure 21. Spearman correlations between NVDA daily Open-to-Open returns and lagged sentiment NSDE proxies.

Correlations are generally modest, which is consistent with the view that any sentiment effect on prices may be weak, nonlinear, and/or delayed. This motivates evaluating sentiment as an exogenous input inside a flexible continuous-time model rather than relying on contemporaneous linear correlation alone. Related evidence in the literature suggests that sentiment–return dependence may be easier to detect **conditionally** during high-attention "event" periods (e.g., volume peaks), even when unconditional correlation is low (Ranco et al., 2015).

4.4.1 *Variation 0 (baseline: no sentiment)*

$$dz_t = f_p(t, z_t)dt + g_p(t, z_t)dW_t^p \qquad (39)$$

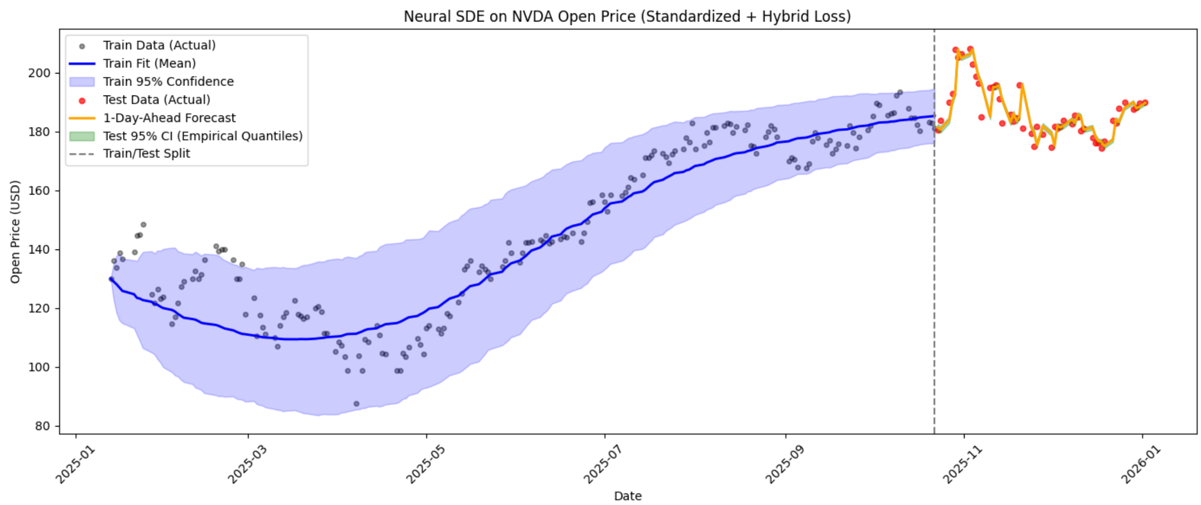

Figure 22: Variation 0 (baseline: no sentiment) — in-sample training fit and out-of-sample one-step-ahead forecasts with 95% predictive interval.

Figure 22 shows the full sample split into training and test periods. To the left of the dashed line, the figure shows the in-sample training fit of the baseline NSDE; to the right, it shows the out-of-sample one-step-ahead walk-forward forecasts and their 95% predictive interval.

The Train Fit (Mean) trajectory smoothly captures the broader market structure—properly identifying the Q1 2025 drawdown and the subsequent prolonged rally—without overfitting to high-frequency daily noise. This indicates that the neural network parameterizing the drift function is successfully generalizing the overarching momentum.

The broad 95% Confidence Interval in the training set demonstrates the model's grasp of NVDA's substantial inherent volatility. By encompassing nearly all historical data points, the learned diffusion term realistically maps the boundaries of expected price distributions rather than collapsing to a deterministic mean.

The baseline model tracks the general price level reasonably well, and its test forecasts remain close to the observed path in several parts of the test window. However, the predictive bands are relatively narrow, which is consistent with the low interval coverage reported in Table 4.3. The test set's 95% Confidence Interval (Empirical Quantiles) is remarkably narrow around the 1-day-ahead prediction, heavily contrasting with the wide unconditional bounds of the training phase. This suggests that given the exact state at time t the model holds high confidence in its state prediction for $t+1$.

4.4.2 *Variation 1 (trend μ_{t-1}^{s} enters drift only)*

$$dz_t = f_p(t, z_t, \mu_{t-1}^{s})dt + g_p(t, z_t)dW_t^p \qquad (40)$$

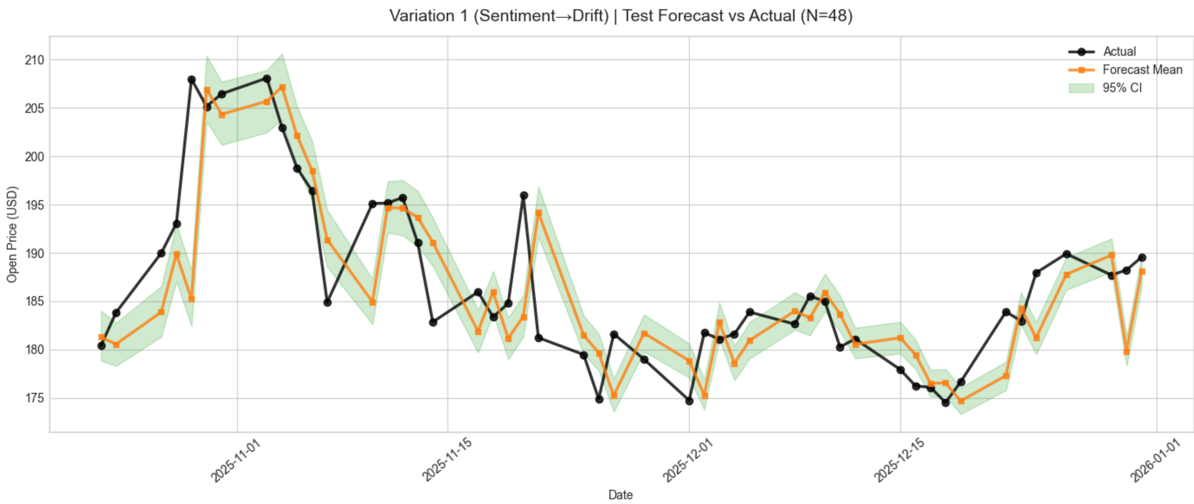

Figure 23: Variation 1 (trend→drift) — in-sample training fit and out-of-sample one-step-ahead forecasts with 95% predictive interval.

Figure 23 again shows both the training fit and the test forecasts, but now the lagged sentiment trend enters only the drift term. In this setup, sentiment is intended to influence the direction of the mean price dynamics, while uncertainty remains governed by the original diffusion specification.

Relative to the baseline, the test forecasts appear less well aligned with the observed series, and the model does not visibly improve directional behavior. The intervals are somewhat wider than in Variation 0, but the gain in coverage is not accompanied by better point accuracy. While the general trend is accurate, a close inspection of the Forecast Mean reveals a slight phase shift or 1-day lag during acute market pivots (e.g., the sharp upward spikes around November 1st and November 25th). The forecast often peaks one day *after* the actual price. This indicates that while the sentiment data is contributing to the drift, the autoregressive nature of the recent price history might still be dominating the network's short-term predictions. The sentiment features may act more as a coincident indicator rather than a strong leading indicator for these sudden shocks.

4.4.3 *Variation 2 (trend μ_{t-1}^{s} enters drift and diffusion)*

$$dz_t = f_p(t, z_t, \mu_{t-1}^{s})dt + g_p(t, z_t, \mu_{t-1}^{s})dW_t^p \qquad (41)$$

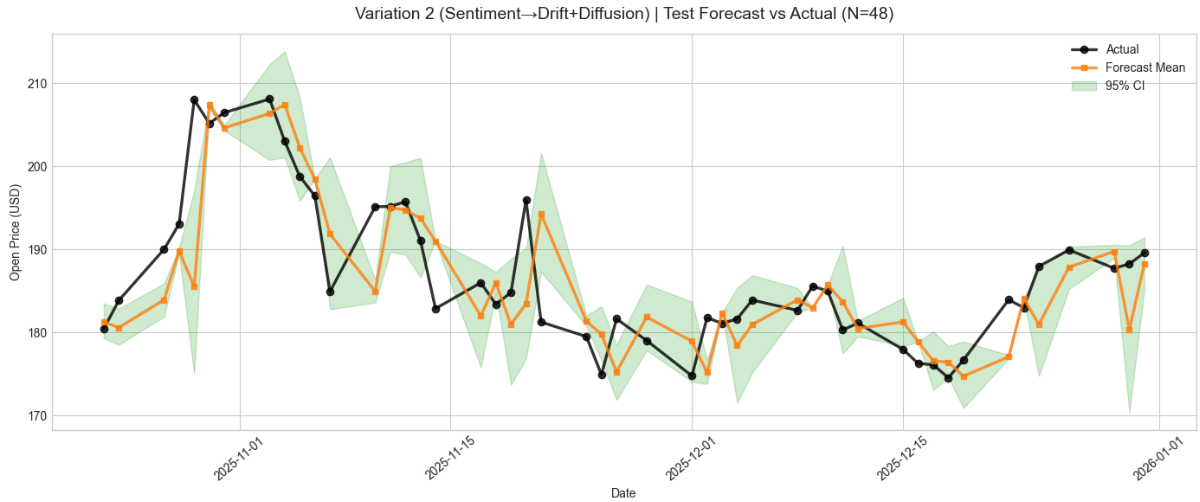

Figure 24: Variation 2 (trend→drift+diffusion) — in-sample training fit and out-of-sample one-step-ahead forecasts with 95% predictive interval.

In this variation, as shown in figure 24, the lagged sentiment trend affects both the drift and the diffusion terms. Therefore, sentiment is allowed to influence not only the central forecast path, but also the spread of the predictive distribution. The predictive mean closely follows the observed opening prices over the testing horizon, indicating that the learned drift successfully captures the dominant market dynamics while responding to changes in investor sentiment.

Compared with Variation 1, the uncertainty bands are visibly broader, indicating that the model is using sentiment to produce more dispersed forecasts. Even so, the test mean forecasts still do not outperform the baseline, so the main effect here is increased uncertainty rather than improved tracking of the observed prices.

4.4.4 *Variation 3 (volatility σ_{t-1}^{s} enters diffusion only)*

$$dz_t = f_p(t, z_t)dt + g_p(t, z_t, \sigma_{t-1}^{s})dW_t^{p} \qquad (43)$$

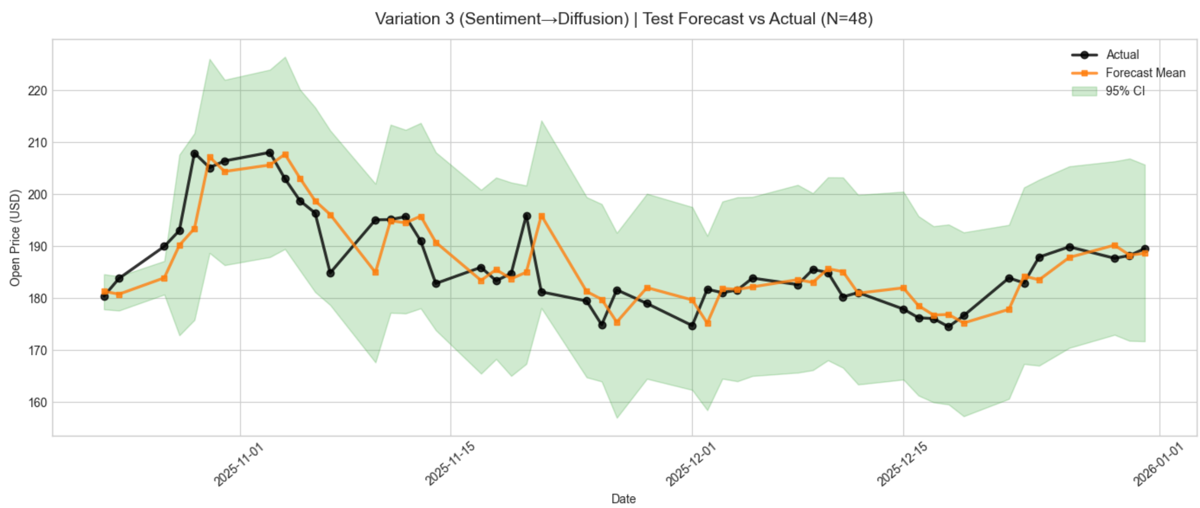

Figure 25. Variation 3 (vol→diffusion) — in-sample training fit and out-of-sample one-step-ahead forecasts with 95% predictive interval.

Figure 25 shows the case where lagged sentiment volatility enters only the diffusion term. As a result, sentiment mainly affects the width of the forecast distribution, while the mean price dynamics remain close to the baseline specification. Incorporating sentiment into the diffusion network enables the model to treat investor sentiment as a driver of market volatility rather than only expected returns. This is consistent with financial theory, where news sentiment often influences the magnitude of price fluctuations even when the expected trend remains unchanged.

Visually, the central forecast path remains broadly similar to the baseline, but the predictive interval becomes much wider across the test window. This matches the quantitative results: point metrics stay close to Variation 0, while coverage increases sharply at the cost of much less sharp intervals.

4.4.5 *Variation 4 (μ^s_{t-1} enters drift , σ^s_{t-1} enters diffusion)*

$$dz_t = f_p(t, z_t, \mu^s_{t-1})dt + g_p(t, z_t, \sigma^s_{t-1})dW^p_t \quad (44)$$

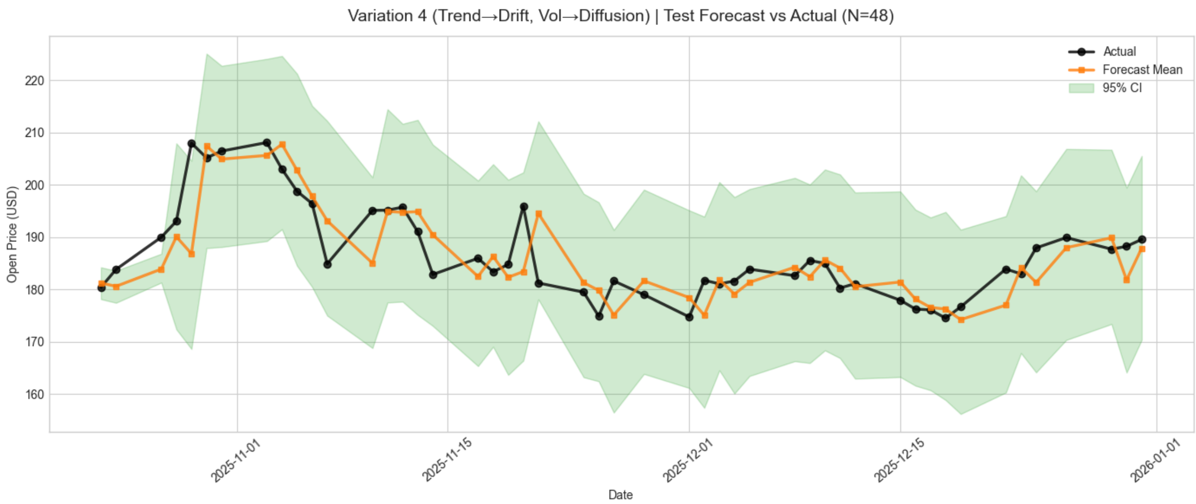

Figure 26. Variation 4 (trend→drift, vol→diffusion) — in-sample training fit and out-of-sample one-step-ahead forecasts with 95% predictive interval.

This specification combines the previous two ideas: lagged sentiment trend affects the drift, while lagged sentiment volatility affects the diffusion. In this variation, trend-related information is incorporated into the drift network, while volatility-related features influence the diffusion network. This separation allows the model to independently learn the expected market direction and the uncertainty surrounding future price movements. The model therefore uses sentiment to shape both the forecast direction and the forecast uncertainty, but through separate channels.

Figure 26 suggests that this combined specification produces very wide test intervals without a corresponding improvement in how closely the mean forecast follows the observed price path. In other words, it improves coverage mainly by increasing dispersion, not by strengthening point prediction.

4.4.6 Variation 5 (full sentiment vector enters drift and diffusion)

Let x_{t-1}^{full} denote the full sentiment feature vector used in the experiment (e.g., including trend and volatility proxies). The model is:

$$dz_t = f_p\left(t, z_t, x_{t-1}^{\text{full}}\right)dt + g_p\left(t, z_t, x_{t-1}^{\text{full}}\right)dW_t^p \quad (45)$$

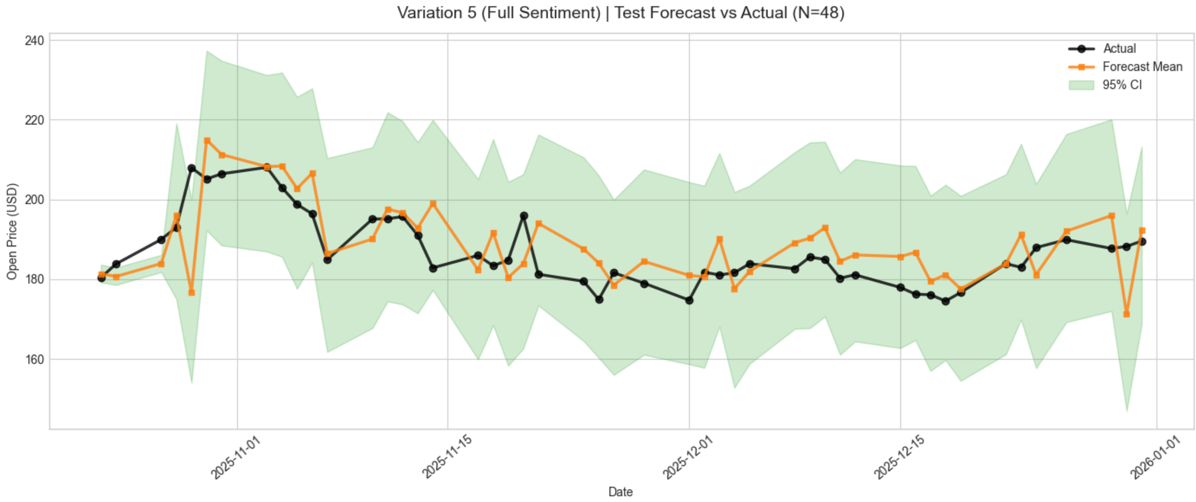

Figure 27: Variation 5 (full→both) — in-sample training fit and out-of-sample one-step-ahead forecasts with 95% predictive interval.

Figure 27 corresponds to the richest specification, where the full lagged sentiment feature vector enters both drift and diffusion. In this variation, the complete sentiment feature vector is supplied to both the drift and diffusion networks. Consequently, investor sentiment influences both the expected direction of price movement and the stochastic variability around that expectation. The intention is to let the model exploit as much sentiment information as possible in both the mean and uncertainty dynamics.

Among all the variations, this plot shows the weakest overall forecasting behavior: the test mean forecast appears less stable and the intervals are extremely wide. This is consistent with the poor point metrics in Table 4.3 and suggests that the full exogenous sentiment input may be too noisy or too complex for the available sample.

Overall Observations for the 5 variations:

- Variations 1–2 (trend injected into drift, and optionally diffusion) do not improve point accuracy in this run, and directional accuracy decreases.
- Variation 3 (volatility injected into diffusion) yields point metrics comparable to the baseline while dramatically increasing interval coverage—at the expense of very wide intervals.
- Variation 5 (full sentiment vector into both drift and diffusion) performs substantially worse in point metrics and produces the widest intervals, suggesting over-parameterization and/or noisy exogenous inputs.

Table 5 summarizes the performance metrics of each variation.

Variation	RMSE	MAE	*R^2*	Directional Acc.	95% CI Coverage	Avg CI Width
Var 0 (baseline)	5.1431	3.7110	0.6620	55.10%	18.37%	1.6922
Var 1 (trend→drift)	5.7372	4.1506	0.5871	39.58%	31.25%	4.3539
Var 2 (trend→drift+diffusion)	5.7224	4.1210	0.5892	45.83%	45.83%	7.0277
Var 3 (vol→diffusion)	5.1363	3.7386	0.6691	54.17%	97.92%	33.3472
Var 4 (trend→drift, vol→diffusion)	5.6035	4.0670	0.6061	41.67%	93.75%	33.3256
Var 5 (full→both)	8.1938	6.2256	0.1578	45.83%	93.75%	42.9061

Table 5: Variation comparison (lag *L=1*, price space).

4.5 Lag sensitivity ($L \in \{2,3\}$)

To test whether sentiment effects appear with delayed impact, we evaluate additional lag settings for selected variations.

Table 6 depicts the performance metrics of the various variations of the price-NSDE models for the various configurations of lagged sentiment.

Model	RMSE	MAE	*R^2*	Directional Acc.	95% CI Coverage	Avg CI Width
Var 1 (Lag 2, trend→drift)	5.5763	4.0897	0.6099	47.92%	29.17%	3.3125
Var 1 (Lag 3, trend→drift)	5.2098	3.8785	0.6595	58.33%	27.08%	3.3220
Var 2 (Lag 2, trend→drift+diffusion)	5.8107	4.2172	0.5764	47.92%	45.83%	7.8181
Var 2 (Lag 3, trend→drift+diffusion)	5.3934	3.9893	0.6351	58.33%	39.58%	4.8499
Var 3 (Lag 2, vol→diffusion)	5.1950	3.8028	0.6615	52.08%	89.58%	19.5634
Var 3 (Lag 3, vol→diffusion)	5.1528	3.7340	0.6669	54.17%	87.50%	17.8080
Var 4 (Lag 2, trend→drift, vol→diffusion)	5.6846	4.1141	0.5946	47.92%	97.92%	33.4116
Var 4 (Lag 3, trend→drift, vol→diffusion)	5.3778	4.0239	0.6372	47.92%	87.50%	22.2269

Table 6: Lag sensitivity (price space).

Observation (lags): some lag configurations improve directional accuracy (notably for lag 3 in Variations 1–2), suggesting any sentiment effect may be delayed. However, improvements are not uniform across metrics.

4.6 Long-horizon and multi-step evaluation without sentiment

The previous sections focused on the standard leakage-safe one-step walk-forward protocol. To test whether the baseline Price NSDE remains stable beyond a single day, we also evaluate two longer-range diagnostics on the **baseline model without sentiment**:

- **Long-horizon free-running forecast:** a single integration over the full test window without resetting to the observed state.
- **Multi-step horizon forecast:** direct h-day-ahead predictions for $h \in \{1,3,5,10,15,20\}$.

These diagnostics are stricter than walk-forward forecasting because any drift bias or diffusion miscalibration compounds as the horizon grows. They therefore reveal whether the baseline model has learned meaningful long-range dynamics or only short-range local transitions as shown in figures 27 and 28.

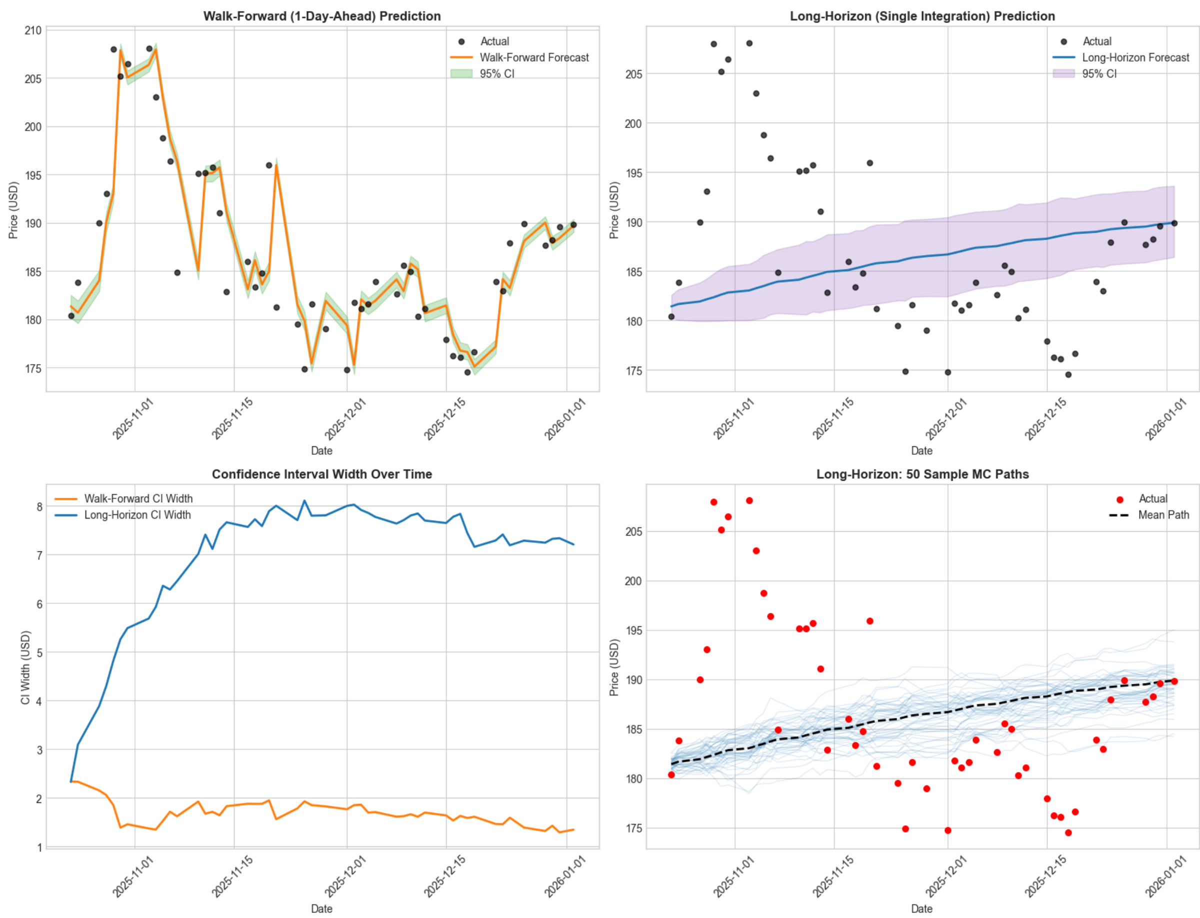

Figure 28. Baseline Price NSDE — walk-forward versus long-horizon forecasts, confidence-interval width evolution, and sampled Monte Carlo paths.

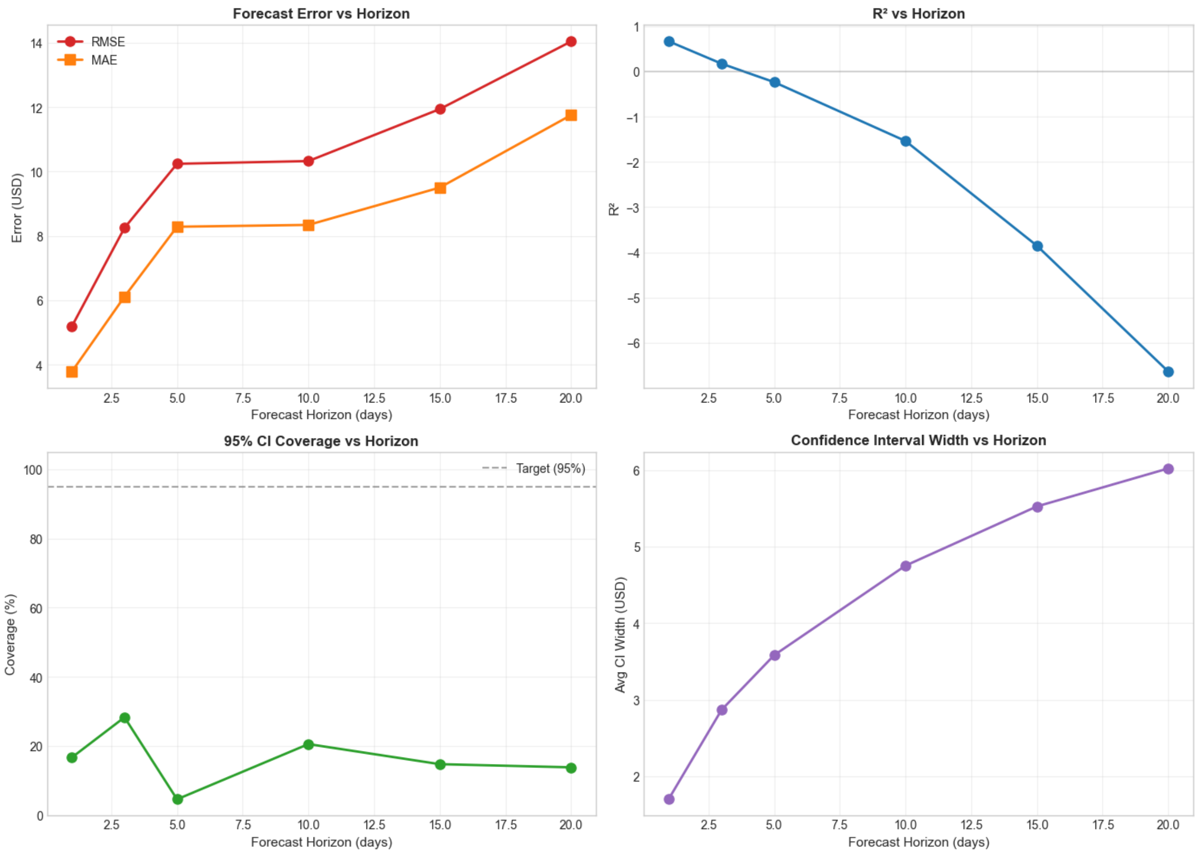

Figure 29. Baseline Price NSDE — RMSE, MAE, R^2, coverage, and interval width as a function of forecast horizon.

The baseline remains the strongest model at the one-day horizon, but forecast error compounds quickly as the horizon increases. This is the expected behavior when a model is tuned primarily for short-run teacher-forced forecasting rather than free-running simulation. The widening confidence bands also indicate that uncertainty accumulates over time; however, the growing interval width does not fully compensate for the degradation in mean forecast quality.

Taken together, the baseline horizon results show that the price NSDE is most reliable as a short-horizon forecaster. It is useful for one-step-ahead updates, but its free-running long-horizon dynamics are not yet robust enough for accurate multi-day extrapolation.

4.7 Long-horizon and multi-step evaluation with sentiment

We repeat the horizon analysis for **Variation 3**, the sentiment-augmented model in which lagged sentiment volatility enters the diffusion term.

Unlike the baseline, this variant is expected to shape the predictive distribution more directly, so it provides a useful test of whether sentiment-derived uncertainty proxies improve long-range calibration.

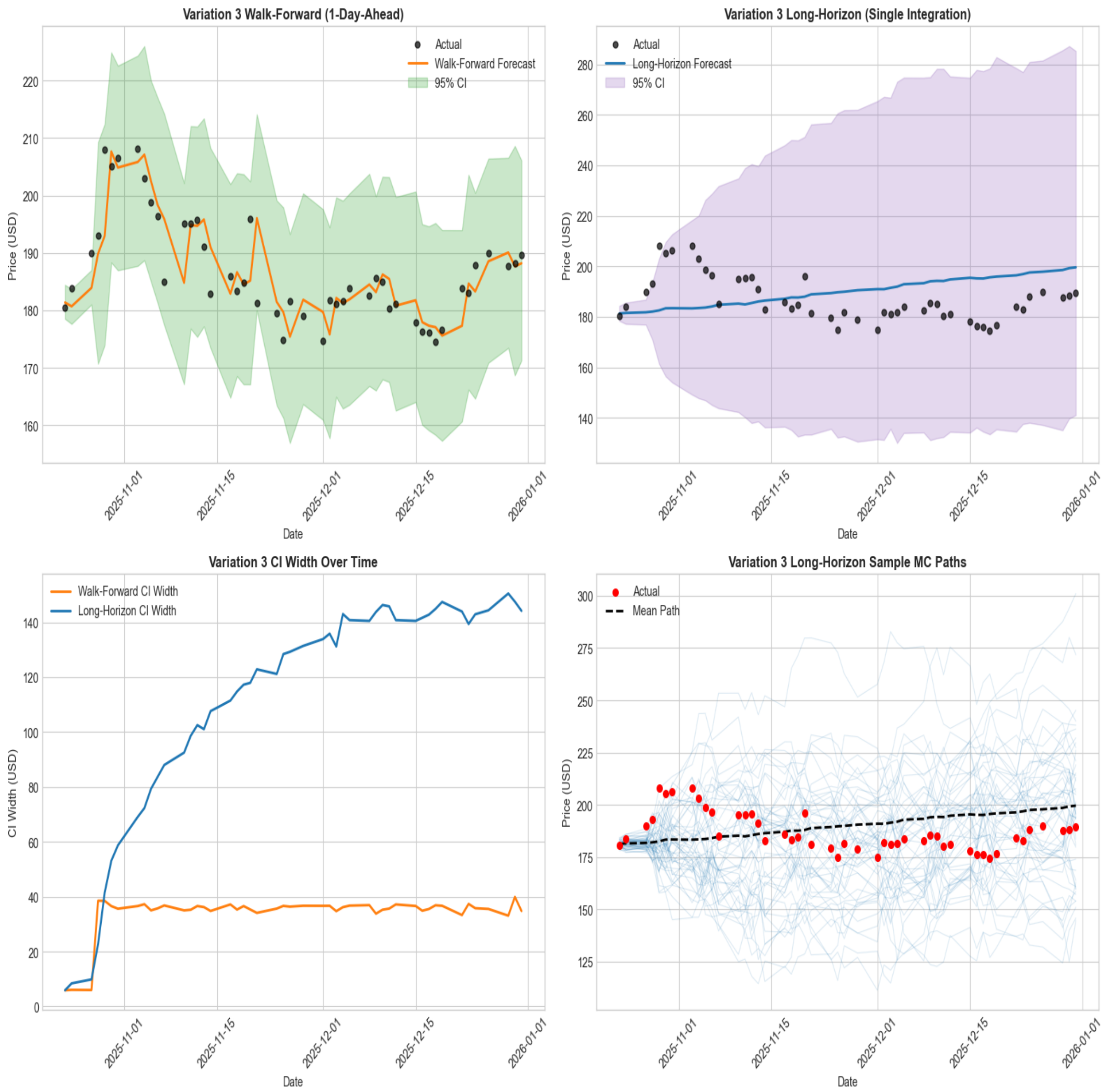

Figure 30. Variation 3 (sentiment volatility → diffusion) — walk-forward versus long-horizon forecasts, confidence-interval width evolution, and sampled Monte Carlo paths.

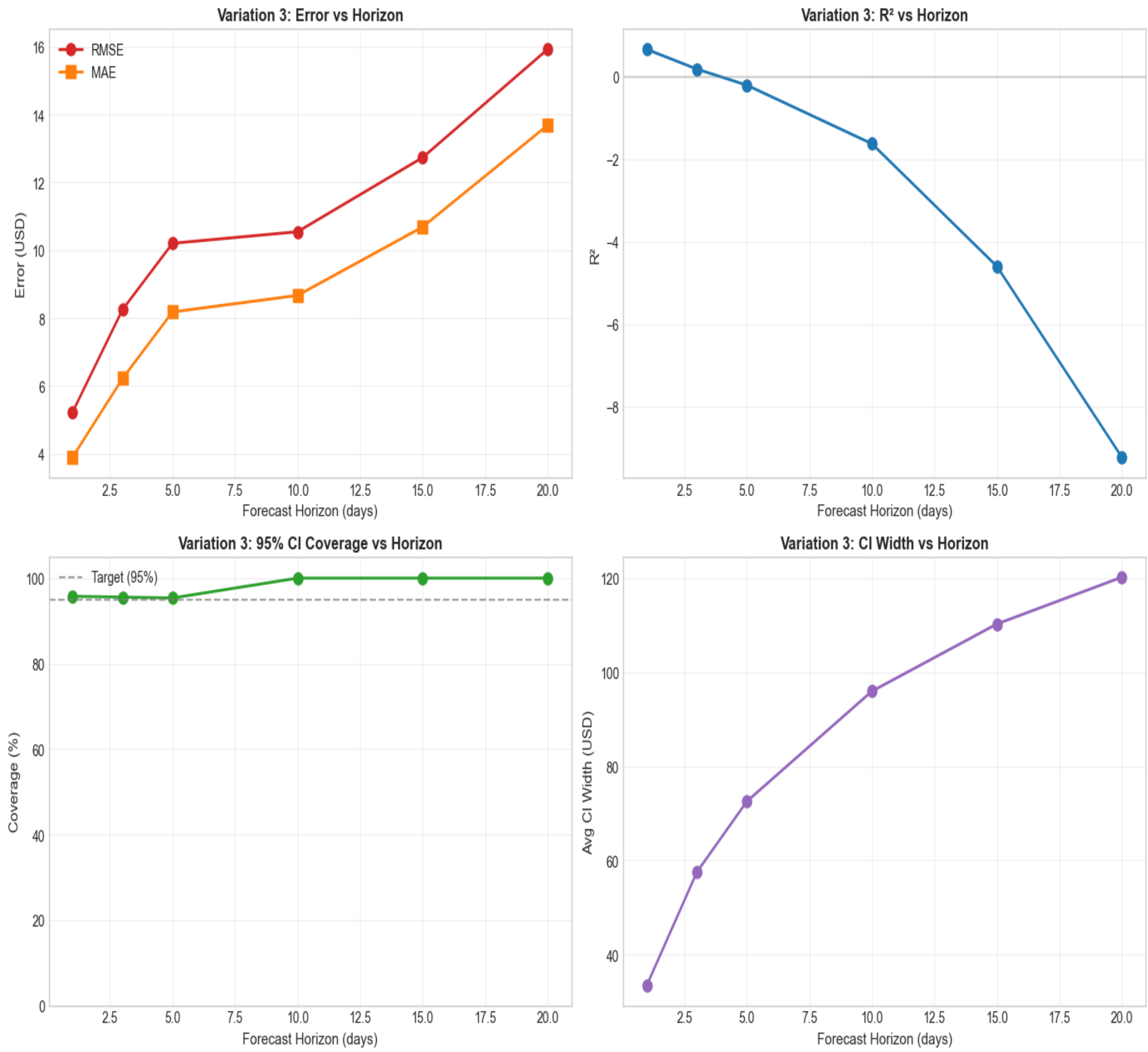

Figure 31. Variation 3 — RMSE, MAE, R^2, coverage, and interval width across forecast horizons.

From the results depicted in figures 30 and 31 we can observe that the long-horizon forecast shows the same overall pattern as the baseline: the one-step fit is reasonable, but free-running integration becomes increasingly difficult as errors accumulate. The key difference is that the sentiment-driven diffusion term increases the spread of the predictive paths, which improves interval coverage but does not materially solve the mean forecast drift.

Variation 3 multi-step summary The results are shown in the table below.

Horizon	RMSE	MAE	R^2	Coverage	CI Width
1	5.2221	3.8939	0.6644	95.74%	33.5834
3	8.2467	6.2166	0.1874	95.56%	57.5823
5	10.2016	8.1776	-0.2028	95.35%	72.5946
10	10.5424	8.6646	-1.6218	100.00%	95.9774
15	12.7279	10.6806	-4.5953	100.00%	110.2586
20	15.9177	13.6788	-9.2240	100.00%	120.2022

Table 7 — Variation 3 multi-horizon evaluation (price space).

The model is credible at the 1–3 day range, where R^2 remains positive and coverage is close to the nominal 95%. Beyond roughly 5 days, the mean forecast deteriorates sharply and R^2 becomes negative, which means the model performs worse than a simple mean-based baseline at those horizons. Coverage eventually reaches 100%, but only because the predictive bands become excessively wide, making them informative for calibration but less useful for practical decision-making.

The Variation 3 results suggest that sentiment-driven diffusion can improve short-horizon calibration, but it does not fix the underlying long-horizon mean dynamics. In other words, the model becomes more conservative as the horizon grows, yet the free-running trajectory still drifts away from the realized price path.

4.8 Comparison with classical and deep-learning baselines

The preceding sections benchmarked the price NSDE against a naive random walk. To place the continuous-time model in a broader methodological context, we additionally train two standard forecasting families—a classical statistical model (**ARIMA**) and a deep sequence model (**LSTM**)—on the **exact same NVDA Open price series, the same chronological 196/49 split, and the same one-step-ahead walk-forward protocol** used for the price NSDE. Each family is evaluated both without sentiment and with a lagged sentiment feature ($L = 3$), and the results are compared against the price NSDE baseline (Variation 0) and the best-calibrated sentiment variant (Variation 3, sentiment volatility into diffusion). Full specifications are given in Section 3.4.4.

Table 8 reports the results. Predictive intervals are constructed in a model-appropriate way: ARIMA/ARIMAX use the Gaussian forecast standard error . The LSTM uses its heteroscedastic variance head; and the NSDE uses Monte Carlo path quantiles. The NSDE V0 and V3 rows are the validated metrics from the price NSDE experiments (Sections 4.2 and 4.4).

Model	**RMSE**	**MAE**	R^2	**Directional Acc.**	**95% CI Coverage**	**Avg CI Width**
Naive (tomorrow=today)	5.1913	3.7302	0.6558	51.02%	89.80%	17.41
ARIMA (2,1,2)	5.02	3.56	0.678	67.35%	89.80%	17.41
ARIMAX + sentiment (lag 3)	5.02	3.55	0.678	69.39%	89.80%	17.41
LSTM (price only)	5.24	3.90	0.649	51.02%	79.59%	11.40
LSTM + sentiment (lag 3)	6.11	4.71	0.524	48.98%	63.27%	9.20
Price NSDE V0 (baseline)	5.1431	3.7110	0.6620	55.10%	18.37%	1.6922
Price NSDE V3 (sentiment→diffusion)	5.1363	3.7386	0.6691	54.17%	97.92%	33.3472

Table 8 — Classical and deep-learning baselines vs. price NSDE (identical data, price space).

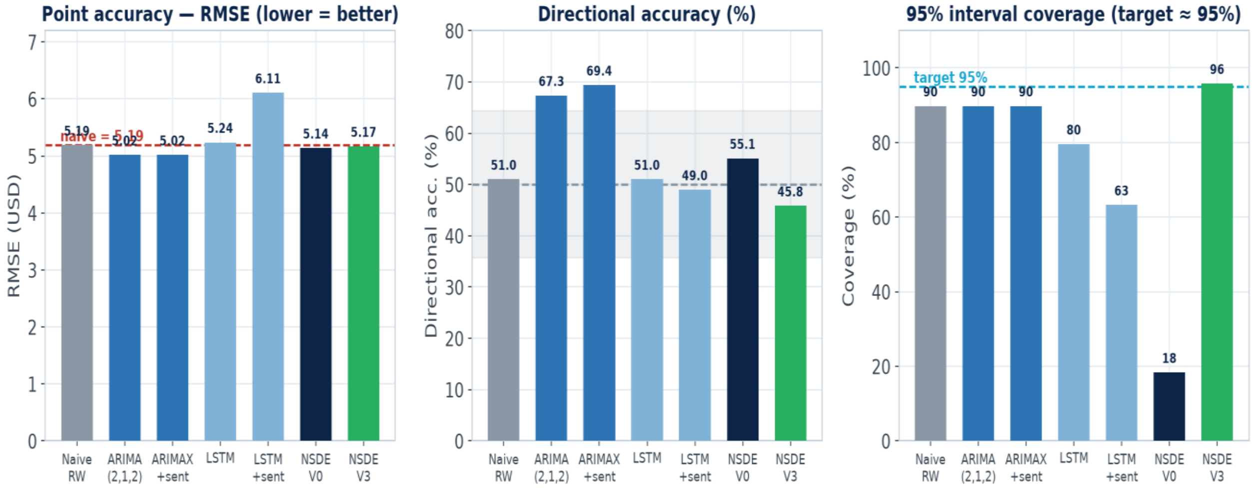

Figure 32. *Baseline comparison on the identical NVDA Open dataset: point accuracy (RMSE), directional accuracy, and 95% interval coverage across the naive benchmark, ARIMA/ARIMAX, LSTM/LSTM + sentiment, and the price NSDE (V0, V3). The shaded band in the middle panel is the approximate 95% no-skill interval for directional accuracy on a 49-day test window.*

Observations:

- **Point accuracy is essentially flat near the naive floor.** All models cluster around RMSE ≈ 5.0–5.2 and $R^2 \approx 0.65$–0.68; the only clear outlier is LSTM+sentiment, whose accuracy degrades (RMSE 6.11, R^2 0.52). No model meaningfully beats "tomorrow = today" in mean forecasting, which is the expected behaviour for a near-random-walk daily price series over a short window.

- **ARIMA shows a marginal directional edge.** ARIMA (order (2|1|2), AIC-selected) and ARIMAX reach 67–69% directional accuracy, versus ≈ 51% for the naive and LSTM baselines. On a 49-day test window the standard error of a hit-rate is , so the 95% no-skill band is roughly [36%|64%]. The ARIMA/ARIMAX values sit just above this band: the edge is suggestive and only marginally significant, and should not be over-interpreted given the thin window.

- **Lagged sentiment does not help the mean.** Adding lag-3 sentiment leaves ARIMA point/interval metrics essentially unchanged and slightly harms the LSTM (both accuracy and calibration). As an additive exogenous regressor in these models, sentiment is neutral-to-harmful.

- **Where sentiment does act is on uncertainty, inside the NSDE.** The classical models obtain reasonable coverage "for free" from their residual-variance intervals (≈ 90% for ARIMA, at moderate width). The price NSDE V0 is the opposite: very sharp but badly under-dispersed (18% coverage, width 1.69). Injecting sentiment volatility into the diffusion term (V0→V3) raises coverage from 18% to 96%—at the cost of much wider intervals—without changing point accuracy. This reinforces the central finding: **sentiment reshapes the predictive uncertainty, not the mean.**

4.9 Summary of results

Overall, the results provide a consistent narrative across the sentiment and price experiments. The sentiment NSDE produces a stable trend proxy and achieves higher directional accuracy than the naive baseline, but its predictive intervals are under-dispersed, indicating that uncertainty calibration remains imperfect at the sentiment stage.

On the price side, the baseline Price NSDE slightly improves upon the naive "tomorrow = today" benchmark in point accuracy, yet its 95% coverage is low, reinforcing the same calibration issue observed in sentiment. Loss ablations confirm that MSE-only yields overly narrow intervals, whereas NLL-only broadens intervals and improves coverage at the cost of width; the hybrid objective provides a practical compromise.

When sentiment features are injected into the price NSDE, the effects are nuanced. Variations that inject sentiment trend into drift (with or without diffusion) do not improve point accuracy in this run and tend to reduce directional accuracy.

In contrast, injecting sentiment volatility into diffusion (Variation 3) preserves point accuracy while substantially increasing coverage, but the resulting intervals become very wide. The full feature vector in both drift and diffusion (Variation 5) performs worst on point metrics and produces the widest intervals, suggesting that noisy or high-dimensional exogenous inputs can destabilize forecasts.

Lag sensitivity experiments indicate that sentiment effects, where present, are more likely to appear with a delay (lag 3 improves directional accuracy for some variants), but gains are not uniform across metrics.

The exploratory correlation heatmaps further support this picture, showing modest unconditional associations that motivate nonlinear, regime-aware models rather than simple correlation-based inference.

The horizon-based experiments add an important caveat: the price NSDE is substantially more reliable at short horizons than at medium or long horizons. The baseline model remains acceptable under one-step walk-forward evaluation, but its free-running long-horizon behavior degrades rapidly. Variation 3 improves short-horizon calibration by expanding uncertainty, yet this does not translate into robust long-range point accuracy. The multi-step results therefore reinforce the interpretation that the current diffusion specification is too conservative or too weakly structured for multi-day extrapolation, even when sentiment is included.

Finally, the head-to-head comparison against classical (ARIMA/ARIMAX) and deep (LSTM) baselines on the identical dataset and protocol (Section 4.8) confirms and sharpens this narrative. In terms of point accuracy, every model—naive, ARIMA, LSTM, and NSDE—clusters near the same random-walk floor (RMSE $\approx$ 5.0–5.2), and lagged sentiment as an additive regressor does not improve the mean in either the classical or the deep model. ARIMA does show a marginal directional edge (67–69% vs. $\approx$ 51%), but it sits only just above the 95% no-skill band for a 49-day window and is therefore suggestive rather than conclusive. The genuinely distinctive behaviour of sentiment appears in the uncertainty channel of the NSDE: moving from V0 to V3 lifts interval coverage from 18% to 96% while leaving point accuracy unchanged. Taken together, the baselines make the thesis's central message concrete—sentiment reshapes the predictive distribution rather than the conditional mean.

5. Conclusions and future work

5.1 Interpretation of Findings

This thesis evaluated continuous-time Neural SDE models for both sentiment dynamics and price forecasting on NVDA daily data using a leakage-safe, one-step-ahead walk-forward protocol. The sentiment NSDE provided modest improvements over a naive "tomorrow = today" baseline (RMSE 0.1486 vs 0.1524; MAE 0.1095 vs 0.1121) and achieved relatively high directional accuracy (70.42%), indicating that the learned sentiment trend captures short-term directional changes. However, predictive intervals were under-dispersed (95% coverage ≈ 46%), suggesting that uncertainty calibration in the sentiment model remains imperfect.

For price forecasting, the baseline price NSDE (no sentiment) slightly outperformed the naive benchmark in point accuracy but exhibited low interval coverage. Loss ablation showed that MSE-only yields overly narrow intervals, NLL-only improves coverage at the expense of width, and the hybrid loss offers a practical middle ground. Injecting sentiment into diffusion (Variation 3) substantially increased coverage but produced very wide intervals, while injecting the full sentiment vector into both drift and diffusion (Variation 5) degraded point accuracy and led to the widest intervals. Lag sensitivity experiments suggested that sentiment effects, where present, may be delayed rather than contemporaneous. Exploratory correlation analysis showed only modest linear/monotone associations between returns and lagged sentiment proxies, consistent with the need for flexible nonlinear models rather than simple correlation-based inference.

An additional practical consideration is that the sentiment-augmented price models operate on a multi-stage pipeline: sentiment scores are first extracted from text, then modeled by the sentiment NSDE to produce trend/volatility proxies, and finally consumed by the price NSDE. Each stage can introduce noise; thus, some degradation in downstream performance likely reflects **cumulative noise propagation** rather than a lack of economic signal per se.

The horizon-based analysis adds a more sobering picture. Under long-horizon free-running simulation, the baseline price NSDE remains useful only at short ranges and degrades as errors accumulate. The multi-step horizon study confirms this pattern: the 1-day and 3-day forecasts remain reasonable, but performance deteriorates quickly beyond roughly 5 days. Variation 3 (sentiment volatility into diffusion) improves short-horizon calibration and coverage, yet its longer-horizon forecasts become overly diffused, reaching very high coverage only because the intervals widen substantially. This means sentiment helped calibration at short horizons, but it did not solve the core long-range mean-dynamics problem.

To situate the continuous-time model against established methods, we also compared the price NSDE with classical (ARIMA/ARIMAX) and deep (LSTM) baselines on the **identical NVDA Open dataset, split, and one-step walk-forward protocol**. Point accuracy was essentially flat near the random-walk floor for every model (RMSE ≈ 5.0–5.2), and adding lagged sentiment as an additive regressor did not improve the conditional mean of either baseline. ARIMA displayed a marginal directional edge (67–69% vs. ≈ 51%), but only just above the 95% no-skill band for a 49-day window. The distinctive role of sentiment again appeared not in the mean but in the uncertainty channel of the NSDE: injecting sentiment volatility into the diffusion (V0→V3) lifted 95% interval coverage from 18% to 96% with unchanged point accuracy. This baseline comparison makes the thesis's central message concrete—sentiment reshapes the predictive distribution rather than the conditional mean.

The experiments support four core conclusions. First, continuous-time Neural SDEs provide a viable framework for modeling sentiment and price dynamics under irregular information flow, with walk-forward evaluation enabling leakage-safe assessment. Second, sentiment-derived signals can influence predictive distributions, but their contribution is subtle and can be overwhelmed by uncertainty inflation unless diffusion is carefully constrained. Third, uncertainty calibration is tightly coupled to the diffusion specification, making diffusion design and loss balancing central to both forecast sharpness and coverage.

Fourth, benchmarking against classical (ARIMA/ARIMAX) and deep (LSTM) models on identical data shows that the NSDE's value does not lie in point accuracy—there all methods sit near the random-walk floor—but in providing a principled, tunable uncertainty channel through its diffusion term. It is precisely this channel that sentiment acts upon, which is why sentiment improves calibration (coverage) rather than the mean forecast.

The long-horizon and multi-step diagnostics sharpen this conclusion: good one-step walk-forward results do not automatically imply stable multi-day forecasting. In this thesis, the models are most reliable at the shortest horizons, while free-running extrapolation exposes drift bias and diffusion miscalibration. This is especially important for the model interpretation because it shows that the model is useful as a short-horizon forecasting tool, but not yet as a robust long-horizon simulator.

5.2 Limitations

This study faced several practical limitations that likely affected performance. First, sentiment was extracted from article titles rather than full text due to size and access constraints, which limits contextual richness and can increase noise in the sentiment signal. Second, to avoid ambiguous attribution, we retained only articles that explicitly mention the target company (NVDA); while this reduces misattribution risk, it also discards potentially relevant market context and reduces data coverage. Third, the prompting strategy used for LLM sentiment extraction is relatively plain; more sophisticated prompt designs (e.g., domain-knowledge prompting or structured chain-of-thought) could yield higher-quality sentiment scores. Finally, the pipeline is inherently multi-stage—LLM sentiment extraction → sentiment NSDE → price NSDE—so errors accumulate across stages, making downstream performance sensitive to upstream noise.

An additional limitation is that the diffusion term appears to be the main driver of calibration behavior across horizons. The long-horizon and multi-step results show that the model can become either under-dispersed at short horizons or overly diffuse at longer horizons, depending on the specification. This indicates that the current diffusion parameterization is too flexible for some settings and too weakly constrained for others.

5.3 Future work

Several extensions can strengthen the findings and improve robustness. First, results should be replicated across multiple random seeds and additional equities or indices to quantify variability and generality. Second, stability-aware diffusion design (e.g., structured or constrained parameterizations such as geometric or multiplicative noise) may improve calibration without excessive interval widening. Third, event-driven evaluation can test whether sentiment effects concentrate around high-attention periods (e.g., news-volume spikes or sentiment-uncertainty peaks). Finally, upstream sentiment extraction can be improved by incorporating full-article text where feasible, using more advanced prompting strategies (domain-knowledge or chain-of-thought), and exploring consensus voting to reduce LLM variance; these steps would also help mitigate cumulative noise in the sentiment-to-price pipeline.

Sentiment extraction can be improved with domain-knowledge prompting and/or consensus voting, and the overall sentiment-to-price pipeline can be made cleaner by denoising sentiment features or by jointly training sentiment and price models with shared uncertainty calibration. Finally, adding classical statistical baselines (ARIMA/VAR) and systematic ablations of sentiment features (trend vs volatility vs interval width) would further contextualize the marginal contribution of each exogenous input.

The new horizon diagnostics suggest two additional future directions. First, calibration should be optimized explicitly for multi-horizon performance, not just for one-step coverage. Second, the model should be evaluated with horizon-aware objectives or constrained diffusion schedules so that the predictive variance grows at an appropriate rate over time. That would allow the framework to move from a short-term forecasting model toward a genuinely stable multi-day simulator.

Overall, the thesis establishes a rigorous, leakage-safe continuous-time framework for integrating sentiment and price dynamics, while highlighting the importance of stability and calibration in Neural SDE-based forecasting.

References

Araci, D. (2019). FinBERT: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*. https://arxiv.org/abs/1908.10063

Black, F., & Scholes, M. (1973). The pricing of options and corporate liabilities. Journal of Political Economy, 81(3), 637-654. https://doi.org/10.1086/260062

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*. https://arxiv.org/abs/2005.14165

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., & Duvenaud, D. (2018). Neural ordinary differential equations. *arXiv preprint arXiv:1806.07366*. https://arxiv.org/abs/1806.07366

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems, 2(4), 303-314. https://doi.org/10.1007/BF02551274

Davidovic, M., & McCleary, J. (2025). News sentiment and stock market dynamics: A machine learning investigation. *Journal of Risk and Financial Management, 18*(8), 412. https://doi.org/10.3390/jrfm18080412

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778. https://doi.org/10.1109/CVPR.2016.90

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. Neural Networks, 2(5), 359-366. https://doi.org/10.1016/0893-6080(89)90020-8

Kelly, J., Bettencourt, J., Johnson, M. J., & Duvenaud, D. K. (2020). Learning differential equations that are easy to solve. Advances in Neural Information Processing Systems, 33, 4370-4380. https://arxiv.org/abs/2007.04504

Kidger, P., Foster, J., Li, X., Oberhauser, H., & Lyons, T. (2021). Neural SDEs as infinite-dimensional GANs. International Conference on Machine Learning (ICML), 5453-5463. https://arxiv.org/abs/2102.03657

Kirtac, K., & Germano, G. (2024). Sentiment trading with large language models. *arXiv preprint arXiv:2412.19245*. https://arxiv.org/abs/2412.19245

Li, X., Wong, T. K. L., Chen, R. T. Q., & Duvenaud, D. (2020). Scalable gradients for stochastic differential equations. International Conference on Artificial Intelligence and Statistics (AISTATS), 3870-3882. https://arxiv.org/abs/2001.01328

Li, X., Xie, H., Chen, L., Wang, J., & Deng, X. (2021). Exploiting textual and relationship information for fine-grained financial sentiment analysis. *Knowledge-Based Systems, 229*, 107334. https://doi.org/10.1016/j.knosys.2021.107334

Loughran, T., & McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *The Journal of Finance, 66*(1), 35–65. https://doi.org/10.1111/j.1540-6261.2010.01625.x

Oh, Y., Lim, D.-Y., & Kim, S. (2024). Stable neural stochastic differential equations in analyzing irregular time series data. *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/2402.14989

Ranco, G., Aleksovski, D., Caldarelli, G., Grčar, M., & Mozetič, I. (2015). The effects of Twitter sentiment on stock price returns. *PLOS ONE, 10*(9), e0138441. https://doi.org/10.1371/journal.pone.0138441

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533-536. https://doi.org/10.1038/323533a0

Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance, 62*(3), 1139–1168. https://doi.org/10.1111/j.1540-6261.2007.01232.x

Yang, X., Mei, Y., & Ni, H. (2021). Neural network stochastic differential equation models with applications to financial data forecasting. *arXiv preprint arXiv:2111.13164*. https://arxiv.org/abs/2111.13164

Zhang, Y., et al. (2024). *A Comprehensive Survey of Sentiment Analysis: Approaches, Challenges and Trends*. ScienceDirect. https://doi.org/10.1016/j.jksuci.2024.102048

Nandwani, P., & Verma, R. (2021). A review on sentiment analysis and emotion detection from text. *Social Network Analysis and Mining, 11*(1), https://pmc.ncbi.nlm.nih.gov/articles/PMC8402961/

Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles (Lakshminarayanan et al., 2017) https://arxiv.org/pdf/1612.01474

Finlay, C., Jacobsen, J.-H., Nurbekyan, L., & Oberman, A. M. (2020). How to train your neural ODE: the world of Jacobian and kinetic regularization. arXiv. https://doi.org/10.48550/arxiv.2002.02798

Ghosh, A., Behl, H. S., Dupont, E., Torr, P. H. S., & Namboodiri, V. (2020). STEER: Simple temporal regularization for neural ODEs. arXiv. https://doi.org/10.48550/arxiv.2006.10711

APPENDIX A

Algorithm 1 Sentiment NSDE Modeling and Feature Extraction

Input: Irregular sentiment observations $(t_i, s_i)i = 1^N$, split time t , $MonteCarlopaths\ \ M$

Output: Leakage-safe sentiment features $\mu_t^s, \sigma_t^s, w_t^s$

$\mathcal{D}_{train}, \mathcal{D}_{test} \leftarrow \text{Split}\left((t_i, s_i), t_{\text{split}}\right)$ ▷ ***Chronological split***

$f_s, g_s \leftarrow \text{TrainNSDE}(\mathcal{D}_{\text{train}})$ ▷ ***Learn drift and diffusion networks***

$\boldsymbol{for}\ t \in \mathcal{D}_{\text{train}}\ \ \boldsymbol{do}:$ ▷ ***In − sample fitted features***

$\mathbf{P} \leftarrow \text{EulerMaruyama}\left(s_{t_0}, f_s, g_s, [t_0, t], M\right)$

$\mu_t^s \leftarrow \frac{1}{M}\sum_{j=1}^{M} P^{(j)}$

$$\sigma_t^s \leftarrow \sqrt{\frac{1}{M-1}\sum_{j=1}^{M}(P^{(j)} - \mu_t^s)^2}$$

$w_t^s \leftarrow Q_{0.975}(\mathbf{P}) - Q_{0.025}(\mathbf{P})$

`end for`

$\boldsymbol{for}\ t \in \mathcal{D}_{\text{test}}$ (in chronological order) do: ▷ Out − of − sample walk − forward

$\mathbf{P} \leftarrow \text{EulerMaruyama}(s_{t-1}, f_s, g_s, [t-1, t], M)$

$$\mu_t^s \leftarrow \frac{1}{M}\sum_{j=1}^{M} P^{(j)}$$

$$\sigma_t^s \leftarrow \sqrt{\frac{1}{M-1}\sum_{j=1}^{M}(P^{(j)} - \mu_t^s)^2}$$

$w_t^s \leftarrow Q_{0.975}(\mathbf{P}) - Q_{0.025}(\mathbf{P})$

$s_t \leftarrow s_t^{\text{obs}}$ ▷ Teacher forcing: update state with observed value

`end for`

$\boldsymbol{return}\ \mu_t^s, \sigma_t^s, w_t^s$

Algorithm 2 Sentiment-Augmented Price NSDE Forecasting

Input: Price observations $(t_j, y_j)_{j=1}^{T}$, sentiment features $\mathbf{x}_t = [\mu_t^s, \sigma_t^s, w_t^s]$, lag L, $MonteCarlopaths\ \ M$

Output: Predictive distributions $\hat{\mu}_t, \hat{\sigma}_t^2$, evaluation metrics

$(t_j, y_j, \mathbf{x}_j) \leftarrow \text{AlignByDate}(y_j, \mathbf{x}_t)$ $\triangleright$ *Inner join on trading dates*

$\mathbf{x}_{t-L} \leftarrow \text{Lag}(\mathbf{x}_t, L)$ $\triangleright$ *Construct lagged sentiment features*

$\mathcal{D} \leftarrow \text{DropIncomplete}(\mathcal{D})$
$\triangleright$ *Remove rows with missing lags*

$\mu_{\text{train}}, \sigma_{\text{train}} \leftarrow \text{Mean}(y_{\text{train}}), \text{Std}(y_{\text{train}})$ $\triangleright$ *Standardize using training statistics only*

$f_p, g_p \leftarrow \text{TrainNSDE}(D_{train})$ $\triangleright$ *Learn price dynamics with exogenous inputs*

$\boldsymbol{for}\ t \in \mathcal{D}_{\text{test}}\ \ (in\ chronological\ order)\ \boldsymbol{do}$: $\triangleright$ *One step − ahead walk forward evaluation*

$\mathbf{P} \leftarrow \text{EulerMaruyama}(z_{t-1}, f_p(\cdot, \mathbf{x}t - L), g_p(\cdot, \mathbf{x}t - L), [t-1, t], M)$

$\hat{\mu}t \leftarrow \frac{1}{M} \sum j = 1^M\ P^{(j)}$

$\hat{\sigma}t^2 \leftarrow \frac{1}{M-1} \sum j = 1^M \left(P^{(j)} - \hat{\mu}_t\right)^2$

$z_t \leftarrow z_t^{\text{obs}}$ $\triangleright$ *Teacher forcing*

$\boldsymbol{end\ for}$

$$\text{MAE} \leftarrow \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_t \left| z_t - \hat{\mu}_t \right|$$

$$\text{RMSE} \leftarrow \sqrt{\frac{1}{|\mathcal{D}_{\text{test}}|} \sum_t \left(z_t - \hat{\mu}_t\right)^2}$$

$$\text{Coverage} \leftarrow \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_t 1\left[z_t \in \left[\hat{\mu}_t - 1.96\hat{\sigma}_t, \hat{\mu}_t + 1.96\hat{\sigma}_t\right]\right]$$

$\boldsymbol{return}\ \hat{\mu}_t, \hat{\sigma}_t^2$, MAE, RMSE, R^2, Coverage