FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

GASTeF
Generative Adversarial Stress Testing for Forecasting - an Application to Finance

Vânia Patrícia da Silva Ribeiro

Master in Data Science and Engineering

Supervisor: Carlos Soares, PhD

Co-supervisor: Vítor Cerqueira, PhD

November 10, 2025

GASTeF
Generative Adversarial Stress Testing for Forecasting - an Application to Finance

Vânia Patrícia da Silva Ribeiro

Master in Data Science and Engineering

November 10, 2025

Resumo

Com a crescente utilização de modelos de aprendizagem automática e aprendizagem profunda em domínios críticos, como a previsão financeira, tornou-se imperativo o estabelecimento de quadros de avaliação rigorosos e transparentes. Apesar do aumento da precisão de previsão, estes modelos apresentam frequentemente uma interpretabilidade limitada devido às suas arquiteturas complexas e não lineares, tornando-os frequentemente "caixas negras" em aplicações práticas. Esta opacidade coloca desafios significativos em discernir as condições em que as previsões dos modelos se tornam pouco fiáveis ou se degradam sistematicamente. Os testes de stress, definidos como a avaliação sistemática do desempenho dos modelos em condições raras, adversas ou de elevada incerteza, constituem um mecanismo vital para avaliar a robustez dos modelos e garantir a sua responsabilização. No entanto, a enumeração exaustiva de todos os cenários de stress plausíveis é computacionalmente inviável, e as metodologias de avaliação prevalecentes captam frequentemente de forma inadequada estes casos extremos críticos, ocultando assim potenciais vulnerabilidades.

No contexto da previsão de séries temporais financeiras, em que a precisão preditiva sob incerteza é crucial, os métodos de avaliação convencionais revelam-se frequentemente insuficientes. As práticas standard baseiam-se geralmente em métricas de desempenho agregadas sobre conjuntos históricos de teste, o que resulta na omissão do comportamento dos modelos em cenários raros, altamente voláteis ou ambíguos. Esta limitação torna os modelos previsionais suscetíveis a excesso de confiança e a degradação do desempenho em períodos de stress. Para enfrentar este desafio, apresentamos a `GASTeF` (Generative Adversarial Stress Test for Forecasting), uma nova estrutura generativa concebida para testar modelos de previsão de stress em aplicações financeiras. O `GASTeF` estende a arquitetura da Rede Adversária Generativa (GAN) ao incorporar as previsões de um dado modelo de previsão no ciclo de treino do gerador. O gerador é otimizado não só para produzir séries temporais financeiras estatisticamente realistas, mas também para identificar regiões no espaço de entrada onde o modelo apresenta uma elevada incerteza ou erro de previsão. Isto é conseguido através de uma função de perda sensível à incerteza que orienta o gerador para amostras geradoras de stress, mas plausíveis.

Inspirando-se no quadro GASTeN, originalmente proposto para testes de stress em tarefas de classificação no domínio da visão por computador, o `GASTeF` generaliza esses princípios para a previsão de séries temporais financeiras. O método proposto é avaliado empiricamente com base em dados diários de rentabilidade de fundos cotados setoriais (ETFs) do S&P 500, abrangendo o período de 2000 a 2023. Os resultados experimentais demonstram que o `GASTeF` gera sequências sintéticas que degradam significativamente o desempenho preditivo, mantendo simultaneamente a credibilidade estatística. Estes resultados evidenciam a eficácia do método na identificação de vulnerabilidades dos modelos e o seu potencial enquanto ferramenta para o desenvolvimento de sistemas de inteligência artificial mais resilientes e transparentes no setor financeiro.

Palavras-chave: Redes Adversariais Generativas (GANs), Dados sintéticos, Aumento de dados, Testes de stress, Séries temporais financeiras, Previsão

Abstract

With the increasing deployment of machine learning (ML) and deep learning (DL) models in critical domains such as financial forecasting, the establishment of rigorous and transparent evaluation frameworks has become imperative. Despite their increasing predictive accuracy, these models frequently exhibit limited interpretability due to their complex, non-linear architectures, often rendering them "black boxes" in practical applications. This opacity poses significant challenges in discerning the conditions under which model predictions become unreliable or systematically degrade. Stress testing, defined as the systematic evaluation of model performance under rare, adverse, or high-uncertainty conditions, constitutes a vital mechanism for assessing model robustness and ensuring accountability. Nonetheless, the exhaustive enumeration of all plausible stress scenarios is computationally infeasible, and prevailing evaluation methodologies often inadequately capture these critical edge cases, thereby obscuring potential vulnerabilities.

In the domain of financial time series forecasting, where predictive accuracy under conditions of uncertainty is critical, conventional evaluation methodologies often prove inadequate. Standard practices typically rely on aggregated performance metrics over historical test datasets, thereby failing to capture model behavior under rare, high-volatility, or ambiguous scenarios. This limitation leaves forecasting models susceptible to overconfidence and performance degradation during stress events. To address this challenge, we introduce `GASTeF` (Generative Adversarial Stress Test for Forecasting), a novel generative framework designed for stress testing forecasting models in financial applications. `GASTeF` extends the Generative Adversarial Network (GAN) architecture by embedding predictions of a given forecasting model within the training loop of the generator. The generator is optimized not only to produce statistically realistic financial time series, but also to identify regions in the input space where the model exhibits high predictive uncertainty or error. This is achieved through an uncertainty-aware loss function that guides the generator toward stress-inducing, yet plausible, samples.

Drawing inspiration from the GASTeN framework, originally proposed for classification stress testing in the context of computer vision, `GASTeF` generalizes these principles to financial time series forecasting. The proposed method is empirically evaluated on daily return data from S&P 500 sector exchange-traded funds (ETFs) spanning the period from 2000 to 2023. Experimental findings demonstrate that `GASTeF` generates synthetic sequences that significantly impair forecasting performance while preserving statistical credibility. These results demonstrate the method's effectiveness in revealing model vulnerabilities and its potential as a tool for developing more resilient and transparent AI systems in finance.

Keywords: Generative Adversarial Networks (GANs), Synthetic data, Data Augmentation, Stress Testing, Financial Time Series, Forecasting

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Carlos Soares and Vítor Cerqueira, for their invaluable guidance and the insightful discussions, which have greatly enriched this work and contributed meaningfully to its development.

A special acknowledgment is extended to my husband, Hugo, whose unwavering support, patience, and steadfast presence have been a constant source of strength. His encouragement has been truly instrumental in the successful completion of this journey.

I am deeply grateful to my parents, grandparents, and sister for their unconditional love and continual support throughout every stage of my life. Their belief in me has been a foundation upon which this achievement was built.

Also, I would like to thank my friends, especially those I had the pleasure of meeting during the course of this journey. Your companionship and encouragement have made this experience all the more rewarding and memorable.

Finally, to everyone who has played a part, big or small, in shaping my academic and personal growth, I am sincerely grateful. Thank you for being a part of this incredible journey.

Vânia Ribeiro

"Start by doing what is necessary,
then do what is possible,
and suddenly you are doing the impossible."

Saint Francis of Assisi

Contents

List of Figures

List of Tables

Abbreviations

AR	Autoregressive
ARCH	Autoregressive Conditional Heteroscedastic
ARIMA	Autoregressive Integrated Moving Average
ARIMAX	Autoregressive Integrated Moving Average with explanatory variables
cGAN	Conditional Generative Adversarial Network
C-WGAN	Conditional Wasserstein Generative Adversarial Network
DL	Deep Learning
ETF	Exchange-Traded Funds
GAN	Generative Adversarial Network
GARCH	Generalized Autoregressive Conditional Heteroscedastic
GASTeF	Generative Adversarial Stress Testing for Forecasting
GASTeN	Generative Adversarial Stress Testing Network
GMMN	Generative Moment Matching Network
LSTM	Long Short-Term Memory
ML	Machine Learning
MSE	Mean Squared Error
MVO	Markowitz's mean-variance Optimisation
PCA	Principal Component Analysis
PReLU	Parametric Rectified Linear Unit
RCGAN	Recurrent Conditional Recurrent Generative Adversarial Network
RNN	Recurrent Neural Network
SARIMA	Seasonal Autoregressive Integrated Moving Average
SARIMAX	Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors
SigCWGAN	Signature Conditional Wasserstein Generative Adversarial Network
SDG	Sustainable Development Goals
TCN	Temporal Convolutional Network
TimeGAN	Time Series Generative Adversarial Network
t-SNE	T-Distributed Stochastic Neighbor Embedding
VAR	Vector Autoregressive
VECM	Vector Error Correction Models
WGAN	Wasserstein Generative Adversarial Network
WGAN-GP	Wasserstein Generative Adversarial Network with Gradient Penalty

Chapter 1

Introduction

In recent years, the adoption of machine learning (ML) and deep learning (DL) in finance has grown substantially [125], driven by the increasing availability of financial data and advances in computational resources. These technologies have achieved strong predictive performance in various financial tasks, including time series forecasting for high-frequency stock market prediction [167], portfolio allocation [101], and algorithmic trading [83]. Despite their effectiveness, DL models are often criticized for their lack of interpretability. Their highly parameterized, nonlinear architectures function as "black boxes," making it difficult for practitioners and regulators to understand or trust their decision-making process [36].

Deploying ML models in high-stakes environments requires careful consideration of transparency, robustness, and ethical implications. This need has given rise to Responsible AI [12], which promotes methodologies that ensure the safe, interpretable, and accountable use of ML systems in practice. One such approach includes the use of model cards [103], which are structured documents that outline a model's intended use, performance, and known limitations.

A particularly pressing concern in finance is how forecasting models perform under stress conditions characterized by high volatility, abrupt market shifts, or rare and extreme events [95]. Traditional evaluation techniques, such as historical backtesting, typically average results across test sets [28], thereby failing to represent such stress conditions adequately. Stress testing addresses this by evaluating model behavior in adverse or uncertain scenarios. In this context, Generative Adversarial Networks (GANs) have emerged as a powerful tool for generating stress-inducing series [27, 140, 54].

1.1 Motivation

The responsible application of ML and DL in finance requires not only high predictive accuracy but also transparency regarding model behavior under uncertainty. Financial time series are characterized by non-stationarity [79, 67], structural breaks, and rare but impactful events [95]. Understanding and adequately evaluating model performance in such stress scenarios is essential

to identifying vulnerabilities such as overconfidence and lack of robustness [127, 28], as conceptually illustrated in Figure 1.1. Despite advances in responsible AI, stress testing methodologies for forecasting under such adverse conditions remain underdeveloped [27, 140].

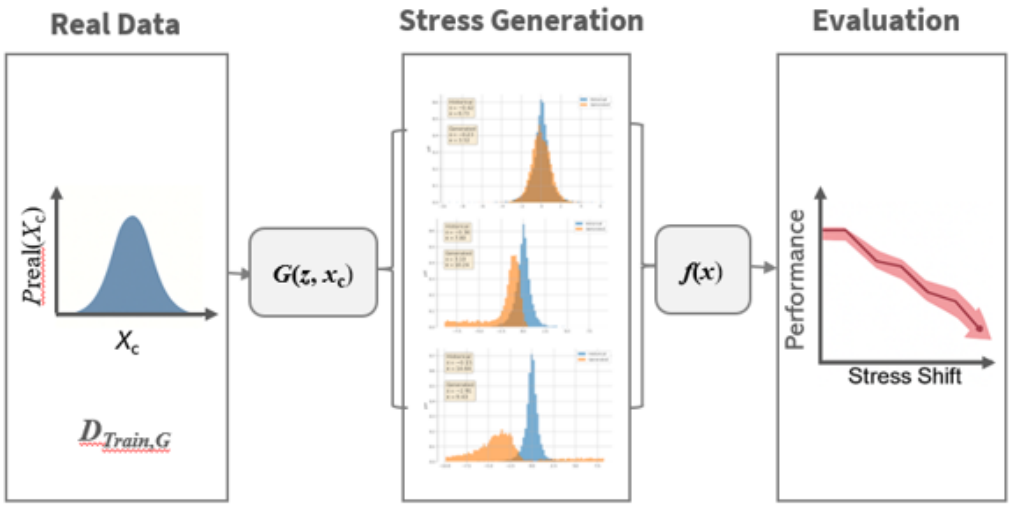

Figure 1.1: Conceptual illustration of stress testing in forecasting models. Real data (X_c) are transformed by the generator $G(z, X_c)$ to produce stressed variants, which are then evaluated by the forecasting model $f(x)$. The decline in performance under increasing stress shifts reveals the model's robustness and sensitivity to adverse conditions.

Robustness under stress is critical for decision-making in finance, where model overconfidence can lead to misallocation of capital or underestimation of risk. Understanding when and why models fail enables practitioners to identify weaknesses in model generalization, improve reliability, and better inform risk management processes. Documentation approaches such as model cards [103] aim to improve transparency by documenting limitations and appropriate use cases. By developing methodologies for stress testing, specifically through the identification of scenarios in which models exhibit reduced reliability, such approaches can support the responsible deployment of AI systems. This includes the development of more robust evaluation tools to assess model performance under stress conditions, ultimately enhancing the safety, transparency, and sustainability of AI applications in finance and other sensitive domains.

1.2 Goals and Research Questions

The primary objective of this work is to develop a generative adversarial framework for stress testing forecasting models in the context of financial time series. The proposed methodology, referred to as `GASTeF` (Generative Adversarial Stress Testing for Forecasting), is inspired by the work of Cunha et al. [27], who introduced GASTeN, a generative approach for stress testing image classifiers by synthesizing adversarial yet realistic samples that expose model vulnerabilities. Adapting this concept to a regression setting, `GASTeF`, aims to generate synthetic, yet statistically

plausible, financial return sequences that are explicitly designed to induce low-confidence and/or high-error predictions from a target pre-trained forecasting model. By doing so, the framework enables the systematic identification of model vulnerabilities under stress, particularly in scenarios characterized by high uncertainty or rare market behavior.

Unlike a standard GAN, which optimizes the generator solely to produce realistic financial time series sequences, our methodology extends the generator's loss with an uncertainty term derived from the output of a pre-trained forecasting model. This integration enables the generator to not only enhance realism but also account for predictive uncertainty directly during the GAN training process. This uncertainty-aware objective encourages the generator to synthesize stress-inducing samples that reveal regions in the input space where the forecasting model becomes less reliable. This design extends the foundational idea proposed in GASTeN [27], which targeted classification models by generating realistic images that fall near the classifier's decision boundary, where predictive confidence is inherently low.

Through this approach, this work aims to contribute to the development of more robust and transparent forecasting systems in finance by advancing methodologies for stress testing and model evaluation under uncertainty. The resulting framework offers an interpretable and data-driven means of diagnosing model weaknesses, with potential applications in risk management, responsible AI, and the construction of more resilient financial decision-support systems.

To achieve these main goals, we aim to answer two specific research questions:

- How can a GAN architecture be adapted to generate synthetic financial time series that consistently trigger low-confidence predictions from a chosen probabilistic model?
- Do these generated synthetic time series offer valuable insights into the behaviour of the selected machine learning model when applied to real-world financial indices, particularly under conditions leading to low-confidence decisions?

This work also seeks to support the United Nations Sustainable Development Goals (SDGs) [147], with a particular emphasis on Goal 9: Industry, Innovation, and Infrastructure. By advancing the reliability and accountability of artificial intelligence models, the study contributes to the development of resilient technological infrastructure and the promotion of innovation. These efforts underpin sustainable industrialization and are expected to facilitate long-term economic development. Furthermore, by fostering the responsible use of AI, the work aims to mitigate potential adverse societal impacts, thereby aligning with broader objectives of sustainable and equitable progress.

1.3 Outline

In addition to the current chapter, the subsequent sections of this work comprise the following sections:

- **Chapter 2: Literature Review** introduces the key concepts and relevant background on time series forecasting, generative adversarial networks, uncertainty-aware forecasting, and

stress testing. It reviews existing approaches and advancements in the field and identifies gaps that this work aims to address.

- **Chapter 3: `GASTeF`: Generative Adversarial Stress Testing for Forecasting** formalizes the research problem and presents the proposed methodological framework. It introduces the `GASTeF` architecture, outlines its design principles, and explains how predictive uncertainty is integrated into the generative process to enable adversarial stress scenario synthesis.

- **Chapter 4: Experimental Setup and Results** presents the empirical evaluation of the proposed `GASTeF` framework. It describes the dataset, baseline forecasting model, training protocol, and experimental configuration, followed by an analysis of the results. The chapter assesses both the statistical fidelity of the generated time series and their effectiveness in inducing predictive degradation and uncertainty in the forecaster, highlighting the trade-off between data realism and stress intensity.

- **Chapter 5: Conclusion** summarizes the main contributions and findings of the study, discusses its limitations, and outlines directions for future research. It revisits the research questions and demonstrates how the proposed methodology advances the field of robustness evaluation in financial forecasting through data-driven stress scenario generation.

Chapter 2

Literature Review

This chapter presents the theoretical background underlying this research, with a focus on financial time series forecasting, generative modeling, and stress testing. Section 2.1 introduces the statistical properties (subsection 2.1.1) and forecasting tasks of financial time series based on horizon, dimensionality, and model granularity (subsection 2.1.2). Subsection 2.1.3 describes labeling schemes and training strategies, outlining supervised, unsupervised, semi-supervised, and reinforcement learning approaches. Subsection 2.1.4 reviews classical and machine learning-based forecasting methods, while subsection 2.1.5 covers evaluation metrics for point predictions. Subsection 2.1.6 discusses uncertainty quantification techniques, including conformal prediction. Section 2.2 surveys generative adversarial networks (GANs), with an emphasis on conditional architectures for sequential data. Finally, Section 2.3 introduces stress testing methodologies, highlighting the GASTeN framework for diagnosing forecasting model vulnerabilities under uncertainty.

2.1 Financial Time Series Forecasting

2.1.1 Definition and Characteristics of Financial Time Series

A time series is defined as a sequence of observations recorded at regular time intervals and indexed chronologically, typically denoted as $Y_t = y_1, \ldots, y_n$, where t represents time. These observations can be collected in continuous or discrete time and are commonly analysed for forecasting future values, understanding underlying phenomena, or supporting data-driven decision-making [5].

In the financial domain, time series data typically consist of periodic observations of asset prices, such as opening, closing, high, and low prices, transaction volumes, and adjusted prices that account for dividends and stock splits [168]. Assets may include stock indices, individual equities, commodities, or exchange rates [138]. Despite the heterogeneity in asset types and measurement units, financial time series often exhibit common structural components such as trends, cycles, seasonality, and irregular variations. These components can be described as follows [3]:

- **Trend**: Represents the long-term movement in a time series, reflecting its general tendency to increase, decrease, or remain stable over an extended period.

- **Cyclical Component**: Captures medium-term fluctuations caused by recurring economic or financial conditions. Unlike seasonal effects, cycles occur over longer intervals and are not strictly periodic.
- **Seasonality**: Denotes systematic, calendar-related fluctuations that repeat at regular intervals within a year. These are often driven by climatic conditions, holidays, or social conventions and are essential for accurate short-term forecasting.
- **Irregular Component**: Comprises unpredictable, non-recurring disturbances in the time series. These are typically caused by exogenous shocks such as geopolitical events, natural disasters, or economic crises.

Financial time series are particularly challenging to precise modelling and accurate forecasting due to several empirical properties, commonly referred to as stylized facts [168, 6]. These include:

- **Autocorrelation**: Refers to the correlation between a time series and its own past values. It quantifies the degree of linear dependence between lagged observations, indicating potential temporal structure or memory in the data. While raw financial returns typically exhibit low autocorrelation, nonlinear transformations such as squared or absolute returns often reveal significant autocorrelations, particularly at short lags [26].
- **Heteroscedasticity**: The variance of returns is not constant over time; instead, it fluctuates, often increasing during periods of market stress or economic uncertainty [168].
- **Volatility Clustering**: Describes the empirical observation that large or small movements in asset returns tend to be temporally clustered. Periods of high volatility are likely to be followed by further high volatility, and similarly for low volatility. While raw returns may exhibit little to no autocorrelation, their squared or absolute values often show significant autocorrelation, indicating persistence in volatility dynamics [6].
- **Nonlinearity**: The dynamics of financial time series cannot be fully captured by linear models, as they exhibit nonlinear dependencies that vary over time and across regimes [168].

Traditional statistical approaches typically assume stationarity or stable relationships between past and future values. However, the presence of non-stationarity necessitates preprocessing techniques such as detrending or differencing [113]. In contrast, modern DL methods offer greater flexibility in capturing nonlinear dependencies and regime changes without requiring explicit stationarity assumptions [108].

2.1.2 Forecasting Dimensions in Financial Time Series

Forecasting in this context refers to estimating future values Y_{n+h}, where h denotes the forecast horizon. Two primary dimensions characterize forecasting problems: the number of variables considered and the length of the forecast horizon.

- **Univariate vs. Multivariate Forecasting:** Univariate models focus on forecasting a single variable, such as the daily closing price or return of a financial asset. In contrast, multivariate models incorporate multiple related time series (e.g., returns of different sectors or macroeconomic indicators), leveraging their joint dynamics to improve forecast accuracy.

- **One-step vs. Multi-step Forecasting:** One-step-ahead forecasting targets the immediate next value (y_{n+1}), whereas multi-step forecasting estimates several values ahead ($y_{n+1}, y_{n+2}, \ldots, y_{n+h}$). However, multi-step forecasting may give rise to potential problems, including error accumulation across prediction horizons, reduced forecast accuracy, and heightened predictive uncertainty [143, 153, 134].

Furthermore, many financial applications involve time series databases $\mathscr{D} = \{Y_1, Y_2, \ldots, Y_n\}$, where each Y_i corresponds to a distinct but related univariate time series (e.g., different sector ETFs or asset classes). In this context, forecasting approaches are categorized as either local or global [71]:

- **Local Models:** These models are trained independently for each time series. Classical statistical approaches such as ARIMA and exponential smoothing typically follow this approach [68].

- **Global Models:** A single model is trained across all time series in the dataset. Global models capture shared patterns across related time series, enabling improved generalization by leveraging cross-series dependencies, whereas local models, trained independently on each series, are restricted to learning temporal dynamics in isolation. This approach has shown superior performance in various forecasting tasks [53].

2.1.3 Labels and Training Strategies

Machine learning encompasses distinct approaches - supervised, unsupervised, semi-supervised and reinforcement learning - each providing distinct capabilities for modeling and analyzing financial time series data:

- **Supervised Learning:** Frequently employed in the context of financial time series forecasting, models learn a functional mapping from input features to target variables using fully labeled datasets. Supervised machine learning consists of two main classes of problems: classification and regression. In a classification problem, the labels constitute a discrete set, while in regression problems the labels are continuous. The goal of machine learning regression algorithms is to discern the function that maps objects to continuous labels, rendering it apt for diverse applications encompassing the prediction of asset prices, estimation of market indices, or forecasting of financial performance metrics [120, 75].

- **Unsupervised Learning:** Seeks to identify hidden patterns and structures in unlabeled financial data [120, 20].

- **Semi-supervised Learning:** Presents a compelling alternative for financial modeling, especially in settings where large quantities of unlabeled time series data are available, but only a subset is labeled.
- **Reinforcement Learning:** Models sequential decision-making problems, where an agent learns optimal actions (e.g., buy, sell, hold) through interaction with a financial environment to maximize cumulative reward (e.g., returns) [11].

2.1.4 Supervised Machine Learning Models

Time series forecasting is not a novel concept because it has long roots in regression. The standard for such techniques is Autoregressive Integrated Moving Average, also known as ARIMA. Meanwhile, variations of this model have been developed, such as SARIMA (or Seasonal ARIMA), and ARIMAX (ARIMA with explanatory variables). For decades, linear regression methods such as autoregressive (AR) models have dominated the forecasting landscape [77, 7]. AR models aim to produce the next time step Y_{t+1} in sequence as a function of the previous n time steps, where n represents the order of the model. Doing this not only predicts future values but also preserves the temporal dynamics of a time series. Although auto-regressive-based approaches offer reasonable predictive performance, they exhibit certain limitations [37]. The models are inherently linear, making them unsuitable for capturing data with nonlinear relationships. Additionally, they rely upon certain assumptions about the data, such as the constant standard deviation, which may not hold in all cases. For these reasons, AR models are by definition deterministic and non-generative, making them less effective at capturing the inherent randomness and uncertainty in real-world time series data [56].

In response, several nonlinear time series models emerged in the late 1970 and early 1980s, including the bi-linear model [114], the threshold autoregressive model [146, 145], the autoregressive conditional heteroscedastic (ARCH) model [40] and the generalised autoregressive conditional heteroscedastic (GARCH) model [14]. The ARCH/GARCH has been particularly influential in financial econometrics, where it is widely used to model and forecast volatility by capturing time-varying conditional variance. These models represent time series as stochastic processes with time-dependent variance, offering greater flexibility in modeling nonlinear dependencies and volatility clustering. Despite their success, the theoretical development and analytical tools for nonlinear time series analysis and forecasting still lags behind its linear counterpart [56]. Also, since they are deterministic, do not allow probabilistic forecasting, or have strong assumptions on the distribution of the target variable, they often fail to fully capture the intricacies of financial markets, particularly the well-known stylised facts [99].

In recent decades, machine learning models and, more notably, DL-based approaches have gained traction and have emerged as powerful competitors to classical statistical methods in the forecasting community [4]. One such model is NHITS [19], short for Neural Hierarchical Interpolation for Time Series Forecasting. These data-driven models, unlike traditional model-driven approaches, are non-parametric and nonlinear, learning the stochastic relationship between past

and future values directly from historical data. As a result, they can replicate many of the stylised facts observed in financial markets.

For time-series forecasting, Recurrent Neural Networks (RNNs) have emerged as a powerful and effective approach, particularly for capturing the sequential nature and long-range dependencies of time-dependent data. One prominent variant, Long Short-Term Memory (LSTM) networks, has become widely used in financial time series forecasting due to its ability to handle complex dependencies [44]. Lee and Yoo [85] introduced an RNN-based approach to predict stock returns, Güresen et al. [59] used neural networks for stock market forecasting, while Sirignano and Cont [130] utilised LSTMs to argue for the universality of asset price features. Deep learning models, particularly LSTM networks, have been shown to outperform traditional ARIMA-based models in time series forecasting, especially for long-term prediction tasks [128]. The M4 competition [96] provided a key benchmark for assessing forecasting methodologies, where the top-performing hybrid ES-RNN model combined exponential smoothing and globally trained LSTMs [133]. Subsequently, the M5 competition [97], focused on retail time series, reinforced these findings by demonstrating the superior performance of machine learning methods over classical techniques.

2.1.5 Measures of Forecasting Accuracy

Forecast performance is commonly assessed using error metrics that quantify the difference between observed and predicted values. The forecast error at horizon h is defined as the difference between the actual value y_{t+h} and the predicted value $\hat{y}_{t+h|t}$, as shown in Equation (2.1):

$$e_{t+h} = y_{t+h} - \hat{y}_{t+h|t} \tag{2.1}$$

According to Shcherbakov et al. [126], these metrics can be broadly categorized as:

- **Absolute error metrics:** measure the magnitude of errors in the same scale as the data;
- **Percentage error metrics:** evaluate errors as a proportion of actual values;
- **Relative error metrics:** assess forecast performance relative to a benchmark;
- **Scaled error metrics:** provide scale-independent assessments for model comparisons across series.

These are the most commonly used point forecast evaluation metrics:

- **Root Mean Squared Error (RMSE):** A widely used metric for scale-dependent data, suitable for comparing forecasts within a single time series or across series with the same scale. Interpreted in the same unit as the original data, facilitating intuitive understanding. Computes the square root of the average of squared errors:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2} \tag{2.2}$$

- **Mean Absolute Percentage Error (MAPE):** Belongs to the class of percentage-based error measures. It is scale-independent and commonly used for model comparison across different datasets. However, becomes undefined when $y_t = 0$, can be heavily influenced by small denominators or outliers and assumes that percentage errors are symmetric, which is not always the case. MAPE expresses forecast errors as a percentage of the actual values:

$$MAPE = \frac{100}{n} \sum_{t=1}^{n} \left| \frac{e_t}{y_t} \right| \tag{2.3}$$

- **Mean Absolute Scaled Error (MASE):** Scales the forecast error relative to the in-sample mean absolute error of a naive forecast. MASE provides a consistent scale-independent measure of forecast accuracy across series of different magnitudes.

$$MASE = \frac{1}{n} \sum_{t=1}^{n} \frac{|e_t|}{\frac{1}{n-1} \sum_{i=2}^{n} |y_i - y_{i-1}|} \tag{2.4}$$

2.1.6 Uncertainty Quantification

Machine learning models often exhibit state-of-the-art performance; however, they frequently overlook the critical aspect of prediction uncertainty in ML-assisted and/or automated decision-making [1, 124]. In critical fields such as finance, the quantification of uncertainty is as important as the prediction itself; therefore, accurate predictions and robust uncertainty quantification are essential for high-stakes decision-making. Well-calibrated and effectively communicated uncertainty can increase transparency, guide stakeholders in trusting model predictions, and address fairness concerns [144, 165].

Transparency in machine learning encompasses procedural transparency, offering insights into model development (code development, model cards, dataset details) [10, 49, 103, 115], and algorithm transparency, revealing a model behaviour to stakeholders [82, 117]. While explainability has been a primary focus for algorithm transparency [93], understanding specific model behaviour alone might be insufficient for stakeholders to assess accuracy or knowledge adequacy. Since 2018, deep learning research has started to focus on understanding system comprehension and identifying areas of uncertainty, giving rise to key concepts such as "uncertainty quantification" and "uncertainty estimation" [92, 98, 141, 148].

While there are many sources of uncertainty [149], in our work we focus on those we can quantify in machine learning models: aleatoric and epistemic uncertainty [32, 80, 48]. Aleatoric uncertainty, also known as irreducible uncertainty, arises from inherent randomness or noise in inputs or targets. On the other hand, epistemic uncertainty originates from a lack of knowledge about the optimal function explaining the observed data. While machine learning models typically account for aleatoric uncertainty through the specification of a noise model or likelihood function, they are less likely to account for epistemic uncertainty. The latter is instrumental in detecting dataset shifts [110], identifying adversarial inputs [161], and guiding exploration-centric methods such as Bayesian Optimisation [64] and reinforcement learning [45].

The discourse on uncertainty quantification has notably gained momentum, particularly in domains such as computer vision and natural language processing, with a primary focus on classification problems [58, 63, 106, 148]. It is essential to recognise that methods for uncertainty quantification exhibit significant variations between classification and regression problems. Classification approaches often initiate the process with probability estimates, contributing to uncertainty quantification, out-of-distribution detection, and open-set recognition [50]. Regression problems are often characterised by the prevalent use of point predictors that provide a single summary statistic like a conditional mean. More sophisticated methods adopt prediction intervals to express uncertainty [17], where wider intervals signify heightened uncertainty.

Prediction intervals are essential for defining bounds on random variables with specified probabilities. Various approaches, such as Bayesian methods [52, 157] and ensemble methods [76, 164], can model the conditional distribution for exact or approximate prediction intervals. Gaussian processes find applications in finance, including option pricing [139] and volatility forecasting [91]. An alternative is a direct estimation without modelling the conditional distribution, achieved through methods like quantile regression [81] or conformal prediction [118, 150]. In their study, [33] conducted a comprehensive comparison of uncertainty quantification methodologies for regression problems, evaluating Bayesian methods, ensemble methods, direct interval estimation methods, and conformal prediction as general approaches. The study revealed that Gaussian processes, commonly applied in finance, assume a (conditional) normal distribution, leading to invalid models when this assumption is violated. In contrast, conformal prediction was identified as a method with guaranteed performance without presuming any specific data distribution. While machine learning literature traditionally assumes data independence and identical distribution (iid), this assumption often does not hold in financial applications. Furthermore, [107] demonstrated the applicability and theoretical guarantees of split conformal prediction, particularly in handling dependent data, prevalent in financial time series. Other studies, such as [158, 74, 21, 51] exemplified diverse applications of conformal prediction. These applications ranged from generating prediction sets for a market maker's net position to forecasting short-term electricity prices, predicting stock returns based on realised volatility, and developing an adaptive method inspired by conformal prediction to address distribution shifts for market volatility prediction.

2.2 Synthetic Time Series Data Generation

The scarcity of high-frequency financial data and the limited number of extreme market events, such as market crashes, have motivated a growing interest in the generation of realistic synthetic financial time series [156, 38, 34].

2.2.1 Synthetic Time Series Generation Methods

Data augmentation and, in particular, resampling techniques have been explored to enrich limited financial datasets and improve the diversity of training samples. The simplest approaches, such as random resampling or bootstrapping, reuse existing observations to expand the dataset but often

lead to redundancy without improving information diversity [111]. In contrast, traditional data augmentation techniques generate new synthetic samples by manipulating existing time-series observations through deformation, shortening, enlargement, or modification. Examples include jittering, dynamic time warping, window slicing, and permutation. In contrast, traditional data augmentation techniques generate new synthetic samples by manipulating existing time-series observations through deformation, shortening, enlargement, or modification. Examples include jittering, dynamic time warping, window slicing, and permutation [69]. However, since these methods rely exclusively on existing data, they tend to generate synthetic series with limited variability and can distort long-term temporal dependencies, thus reducing the fidelity of intertemporal and cross-feature relationships in the resulting time series [154].

Beyond these transformation-based methods, statistical and simulator-based approaches, such as Monte Carlo simulations [31], generate time series through predefined rules or simulation environments, allowing for interpretable and controlled data creation, but with limited flexibility for capturing real-world complexity [136]. In contrast, data-driven approaches rely on observed data to learn the underlying temporal structure and generate new synthetic sequences. These include deep generative models such as Generative Adversarial Networks (GANs) [162, 42, 90], Variational Autoencoders (VAEs) [2], diffusion models [137], and Large Language Models (LLMs) [72], which have shown promise in generating realistic financial time series that reproduce many of the stylized facts observed in real market data [156, 38, 34].

2.2.2 Generative Adversarial Networks

As a simple but powerful framework, GANs have been applied to many contexts and are currently state-of-the-art in image and text generation [73, 116]. Although these computer vision advances have garnered much attention, GANs applications have diversified across disciplines such as time series [42] and sequence generation, namely those involving natural language processing [151], music generation [104] and speech and audio analysis [112].

2.2.2.1 GANs Architectures

In the realm of generative modelling techniques, Generative Adversarial Network (GAN) is an unsupervised generative model first introduced by [55] where two players, a generator and a discriminator, are battling in a zero-sum game[1]. In GANs, both the generator G and the discriminator D are differentiable functions concerning their parameters and are usually represented by neural networks. The generator G takes some noise as input, usually Gaussian, and returns what it believes to be a realistic-looking sample. The discriminator D takes as input either the generated sample from G or a real sample coming from the data and determines where the sample given comes from. The goal of the generator G is to learn to deceive the discriminator D by producing synthetic data points closely mimicking real data, while the discriminator D goal is to learn to

[1] A zero-sum game is a mathematical representation in game theory of a situation that involves two sides, wherein the gain experienced by one side corresponds to an equivalent loss suffered by the other. When aggregating the total gains and subtracting the total losses of all participants, the sum will equal zero.

properly determine the realism of a given data point, discerning between the real data samples and those coming from the generator G. GANs are trained through an adversarial process, wherein the simultaneous training of the two different models, the generator G and the Discriminator D, takes place. The visual representation of this process can be found in Figure 2.1.

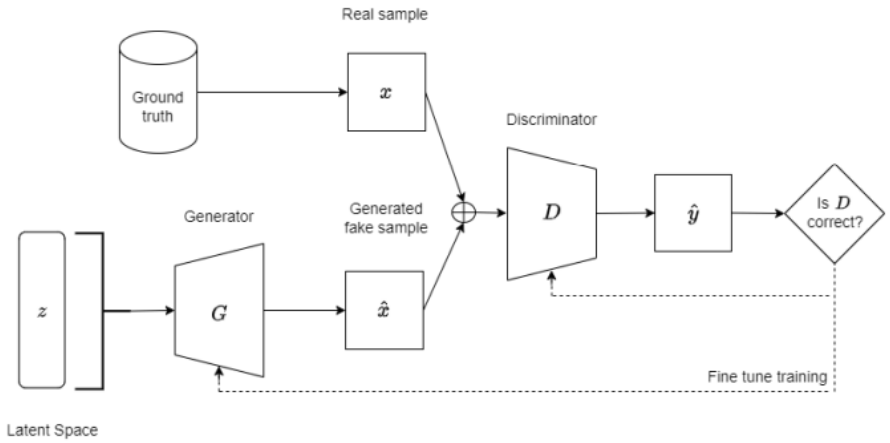

Figure 2.1: GAN architecture

The dynamic interplay between the two networks constitutes a min-max game, where the discriminator D is trained to maximise the probability of assigning the correct label to both samples coming from real data and those coming from the generator G. The generator G is trained to either minimise the log-probability of the discriminator D correctly assigning its outputs as coming from the generator G, or to maximise the log-probability of the discriminator D being mistaken. Competition drives both networks to improve their performance until the genuine data in indistinguishable from the generated one. This two-player min-max game is set in motion by optimising the joint minimax loss function in equation (2.1):

$$\min_G \max_D \mathscr{L}(D,G) = \mathbb{E}_{x\sim\mathbb{P}_{\mathrm{r}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p(z)}[\log(1-D(G(z)))] \tag{2.5}$$

where x is a real data sample drawn from the true data distribution $p(x)$, and z represents a latent variable sampled from the latent space distribution $p(z)$. The latent space provides the random input for the generator, serving as an abstract and typically lower-dimensional representation space from which the generator $G(z)$ maps latent vectors into the data space to produce synthetic samples. The discriminator $D(x)$ outputs the probability that a given input is real and $\mathbb{E}$ denotes the expectation over the respective distributions.

Furthermore, as demonstrated by [55], a unique solution at $\mathbb{P}_{\mathrm{r}} = \mathbb{P}_{\mathrm{g}}$ exists when $D^*(x) = \frac{1}{2}$. This is due to the optimization of the objective function in Equation (2.1) under the optimal generator $D^*(x)$ being equivalent to minimising the cost function:

$$-\log 4 + 2 \cdot JSD(\mathbb{P}_{\mathrm{r}} \parallel \mathbb{P}_{\mathrm{g}}) \tag{2.6}$$

where $JSD(\mathbb{P}_r \parallel \mathbb{P}_g)$ is the Jensen-Shannon divergence between the real data distribution $\mathbb{P}_r$ and the generated data distribution $\mathbb{P}_g$.

and the Kullback–Leibler (KL) divergence, denoted by $\mathrm{KL}(\mathbb{P}_X \parallel \mathbb{P}_Y)$, is defined as:

$$\mathrm{KL}(\mathbb{P}_X \parallel \mathbb{P}_Y) = \int_{\mathscr{X}} P_X(x) \log\left(\frac{P_X(x)}{P_Y(x)}\right) dx \tag{2.7}$$

over the joint support for P_X, P_Y denoted by $\mathscr{X}$.

2.2.2.2 Applications in Time Series

Unconditional GANs, while effective for image and text generation, necessitate a more refined methodology when applied to time series modelling. Specifically, time series generation necessitates a sampling process that considers the series' previous state to capture its statistical attributes, such as autocorrelation, trends, and seasonality. Conditional GANs [102] address this challenge by explicitly incorporating auxiliary conditioning v into the generator and discriminator's decision-making process. This information v can represent various factors, such as class labels, categorical features, or current market conditions. Consequently, cGANs (Conditional Generative Adversarial Network) aim to learn an implicit conditional generative model, particularly suitable for sequential data (e.g., time series, text data) or scenarios involving "what-if" inquiries, such as stress tests [84].

Most cGANs applications related to this work have focused on synthesising data to enhance supervised learning models. Noteworthy instances include works addressing imbalanced classification, specifically in fraud detection [35, 43]. These studies demonstrate that cGANs compare favourably to traditional oversampling techniques. Another pertinent application is found in medical time series generation and anonymization [42], where cGANs are employed to generate realistic synthetic medical data. The cGANs also prove beneficial in the optimization of hyperparameters (fine-tuning) for trading strategies algorithms, using samples generated by the cGAN [84].

Formally defining a cGAN involves incorporating the conditional variable v into the original formulation. The generator and discriminator networks now receive both noise z and conditioning information v as input. The discriminator is trained to maximise the correct labelling of real and generated data, while the generator is trained to minimise the log-likelihood of the discriminator classifying generated data as real. The minimax game between the generator and discriminator is defined as:

$$\min_G \max_D \mathscr{L}(D,G) = \mathbb{E}_{x\sim\mathbb{P}_r(x)}[\log D(x|v)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z|v)))] \tag{2.8}$$

In contrast to our proposed methodology, the aforementioned approaches rely only on binary adversarial feedback for the learning process. This single reliance may not be deemed adequate to ensure that the network comprehensively captures the temporal dynamics in the training data. A notable advancement was made in [159], where the authors introduce Time Series Generative

Adversarial Network (TimeGAN), a novel framework comprising four network components: embedding function, recovery function, sequence generator, and sequence discriminator. TimeGAN integrates unsupervised adversarial loss on real and synthetic data with step-wise supervised loss on the original data. It significantly surpasses previous architectures in terms of discriminative and predictive scores on benchmark datasets. However, TimeGAN produces mini-sequences of fixed length and exhibits impractical training times in contrast to the arbitrary length and time-efficient training regimen demanded by our approach. [135] introduced FIN-GAN, an adaptation of the conventional GAN [55] methodology specifically tailored for the finance domain. The primary objective is to generate synthetic time series that faithfully replicate the stylised facts observed in authentic financial data, without the need for explicit assumptions or complex mathematical formulations. For these reasons, FIN-GAN stands in contrast to stochastic processes and agent-based modelling, which often necessitate non-trivial assumptions to achieve similar outcomes, representing a significant milestone for GANs in the financial domain. [155] presented QuantGAN, a GAN variant that employs temporal convolutional networks (TCNs) to effectively capture extended temporal dependencies, such as volatility clustering. Additionally, innovative transformer-based architectures are incorporated within a GAN framework [89].

These methodologies, however, do not address the simultaneous generation of multiple stocks and their correlation dynamics, lacking realism, crucial for testing diversification strategies in portfolio management [39]. In the financial realm, managing risks relies on understanding correlation dynamics to ensure individual assets remain uncorrelated under various market conditions. Factors such as sector-specific influences, macroeconomic changes, exogenous events, and investor sentiment contribute to stock correlations. Composite financial instruments like Exchange-Traded Funds (ETFs) introduce additional correlation sources [122, 66]. [88] made a pioneering approach to synthesising realistic and high-fidelity stock market order streams using Conditional Wasserstein GAN (C-WGAN). A Conditional Wasserstein GAN (C-WGAN) is derived by integrating a novel cost function utilising the Wasserstein distance. This formulation facilitates performance enhancement for the generator by leveraging the continuity and differentiability inherent in the Wasserstein distance between distributions. Additionally, the model incorporates a conditional variable v to effectively capture temporal dynamics in sequential data over time. The cost function of C-WGAN, simplified through Kantorovich-Rubinstein duality, is formulated as:

$$\min_{G} \max_{D \in \mathscr{D}} \mathbb{E}_{\mathbf{x} \sim p_{\mathrm{d}}}[D(x|v)] - \mathbb{E}_{z \sim p_z}[D(G(z|v))] \tag{2.9}$$

where $\mathscr{D}$ represents the collection of all 1-Lipschitz functions. The Lipschitz conditions are imposed through weight clipping. According to [9] the discriminator's gradient in WGAN behaves more favourably than that of a vanilla GAN, by improving the Optimisation of the generator. Subsequently, [57] proposed introducing a Wasserstein-1 distance with a Gradient Penalty, instead of a standard cross-entropy or zero-sum GAN, to gently enforce the Lipschitz constraint without weight clipping. This method adds a gradient penalty to the training objective of the discriminator, improving training stability. However, a comprehensive study conducted by [94] concludes that

there is limited evidence demonstrating the consistent superiority of WGAN-GP over WGAN, even after thorough hyperparameter tuning and random restarts.

Conditional Sig-Wasserstein GAN (SigCWGAN) [90] provides a robust and theoretically principled mechanism for generating realistic financial time series. By aligning the truncated signatures of real and synthetic trajectories in conditional feature space, it captures both distributional structure and temporal dependencies, outperforming architectures such as TimeGAN, RCGAN (Recurrent Conditional Recurrent Generative Adversarial Network), and GMMN (Generative Moment Matching Network) in capturing long-term dynamics and heavy-tailed behavior [90, 86]. This makes it particularly suitable for financial forecasting and stress testing.

2.2.2.3 Training challenges

Nevertheless, achieving the impressive performance of GANs is a complex undertaking. The training process of GANs is notably unstable, demanding a meticulous choice of architecture, loss functions, and training strategies to ensure stability [8]. Challenges encompass issues such as insufficient variation in simulated outputs (referred to as 'mode collapse'), non-convergence during training, and the inadequacy of training losses as a reliable metric for evaluating GAN output quality.

Mode collapse in GAN training arises from two main factors: discriminator overfitting, where it becomes too adept at identifying generated samples, leading to a reduction in the diversity of generated outputs; and vanishing discriminator gradients, where weak gradients from the discriminator cause learning stagnation for the generator. Convergence does not guarantee training success if mode collapse occurs. Non-convergence in the adversarial training of conventional GANs often indicates failure, potentially caused by oscillatory behaviour and non-convexity of the joint loss function, resulting in unstable representations of the generative distribution. Underfitting or vanishing discriminator gradients can also contribute to non-convergence. Strategies to address those issues includes: regularisation through discriminator learning from stochastically corrupted inputs [8], by introducing noise to the real and generated samples presented to the discriminator; loss function reshaping such as weight regularisation, to prevent overconfidence in the discriminator [119]; utilisation of alternative loss functions like Wasserstein loss (WGAN) to enhance stability [9]; application of gradient penalties in the discriminator learning process (WGAN-GP) to encourage it to produce gradients of a specific magnitude and prevent overfitting [57]; employing a quantile regression loss function to implicitly guide the generator learning the inverse of a cumulative density function [109]; reformulating the objective function using Mean Squared Error (MSE) to minimise the χ^2-distance [100], and considering the discriminator as an energy function that assigns low energy values to high-density regions, directing the generator to sample from these areas [166].

2.2.2.4 Evaluation challenges

Evaluating GAN-generated time series data, particularly financial time series data, remains an open challenge due to the temporal nature of time series data and the lack of established evaluation metrics [15, 16]. Traditional GAN metrics, such as Inception Score [121], Mode Score [60] and Fréchet Inception Distance [65], are not directly applicable to time series data due to their focus on image-related tasks. To address this challenge, several approaches have been proposed. Some studies covert time series data into images and apply traditional GAN metrics [62, 152], however, as [142] argued, the quality of a GAN should be evaluated based on its originally intended purpose. Others develop metrics specifically tailored for time series data. TimeGAN [162] introduces two novel metrics: the Discriminative Score and the Predictive Score. The Discriminative Score evaluates the ability of an external classifier to distinguish between real and synthetic time series data, with the classification error serving as the corresponding metric. The Predictive Score measures the predictive performance of a model trained on synthetic data when applied to real data, with the metric being represented by the Mean Absolute Error (MAE). Another approach directly assesses the quality of the generated time series without using external classifiers. Dimensionality reduction techniques [105, 70], such as t-distributed stochastic neighbour embedding (t-SNE) and Principal Component Analysis (PCA), can be used to visualise and compare the distributions of real and synthetic multivariate time series data, providing a qualitative assessment of their similarity.

In the context of evaluating GANs for financial time series, in addition to the methodologies applied, especially for time series, three general approaches are commonly employed:

1. **Stylised Facts Evaluation**: This approach assesses the extent to which the generated financial time series exhibits stylised facts [24, 25, 18], which are well-established heuristics for the time series behaviour of financial asset returns. This evaluation approach was taken by [135, 155] and by [84], for multivariate data.

2. **Structural Time Series Modelling Evaluation**: This approach compares the performance of the generated financial time series data to the performance of structural time series modelling benchmarks, such as Autoregressive Integrated Moving Average (ARIMA) [163], Generalised Autoregressive Conditional Heteroscedasticity (GARCH) [14], Vector Autoregressive (VAR) and Vector Error Correction Models (VECM) [41] for multivariate data. These evaluation approaches were explored by [135, 155, 84, 29, 47, 129, 46].

3. **'Train on Real, Test on Synthetic' Cross-Validation Evaluation**: This approach [42, 29] involves training a generative model on a portion of training data to produce synthetic data to augment a validation set. Two separate classifiers are then trained, one on the original validation set and another on an augmented validation set. If the classifier performance improves due to augmentation, the model is considered to generate data with the same distribution as real data.

The assessment of GAN-generated time-series data, especially within the realm of financial time series, continues to be a subject of ongoing research. A holistic evaluation, incorporating diverse methodologies such as traditional GAN metrics, domain-specific metrics, and similarity measurements, proves essential for a comprehension and effective appraisal of the generated data.

2.3 Stress Testing

Stress testing is a systematic approach for evaluating the reliability and robustness of machine learning (ML) models in the presence of atypical or adverse conditions [28]. In high-stakes domains such as finance, models are susceptible to performance degradation due to distributional shifts, data irregularities, or misaligned inductive biases. Traditional evaluation metrics often obscure such vulnerabilities. Stress testing addresses this by examining model behavior beyond standard training distributions, offering diagnostic insights into generalization capacity and failure modes, thereby supporting more reliable and responsible ML deployment [131].

Recent advancements in stress testing encompass methodologies like Instance Space Analysis (ISA), which enable a more detailed assessment of model performance by examining behavior across diverse input conditions through techniques such as dimensionality reduction and footprint estimation [132]. Due to the impracticality of comprehensive testing across all input scenarios [131], the literature recommends employing synthetic test instances with controllable properties. This enables systematic and scalable stress testing, allowing for the identification of model weaknesses and enhancing reliability in real-world applications.

One notable contribution in this space is the Generative Adversarial Stress Testing Network (GASTeN), proposed by Cunha et al. [27, 140], which adapts generative adversarial networks (GANs) to synthesize input samples that are simultaneously realistic and challenging for a given classifier. Specifically, GASTeN extends the conventional GAN framework by integrating a target classifier into the training loop of the generator. This enables the generation of samples that maximize classifier uncertainty (i.e., examples associated with low-confidence predictions) while preserving data plausibility. The architecture of the GASTeN framework is depicted in Figure 2.2.

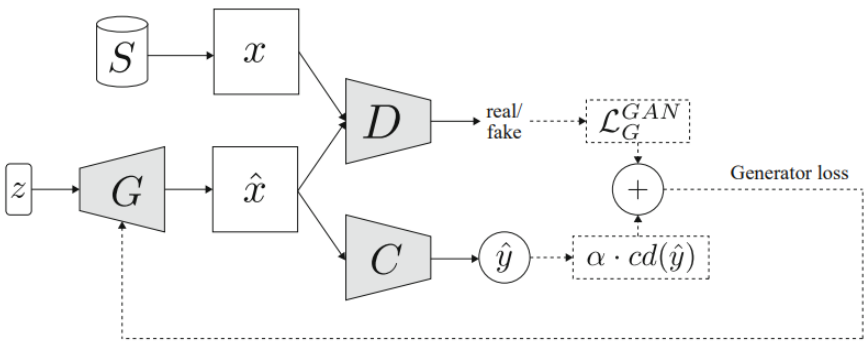

Figure 2.2: GASTeN architecture [27]

To operationalize this, GASTeN augments the generator loss function with a confusion distance term, measuring the divergence of prediction probabilities from the decision boundary (e.g., 0.5 in binary classification). This term incentivizes the generator to synthesize samples located near the model's decision boundary, effectively probing the frontier of classification competence. The framework is empirically validated on benchmark datasets (e.g., MNIST, FashionMNIST), and evaluated using metrics such as Fréchet Inception Distance (FID) for sample realism and Average Confusion Distance (ACD) for classifier uncertainty. However, the authors identify several limitations, including the instability of GAN training, misalignment between valuation metrics (e.g., FID and ACD), and the necessity of post-processing filtering to isolate stress-inducing samples.

The rationale behind GASTeN stems from the inadequacy of traditional evaluation metrics to capture model reliability under distributional shifts or edge-case scenarios. By systematically generating and analyzing samples in regions of high epistemic uncertainty, stress testing frameworks like GASTeN contribute to the diagnostic assessment of ML models, supporting more transparent deployment in high-stakes domains. Furthermore, stress testing aligns with the emerging paradigm of responsible AI by facilitating the development of model cards [103]—structured documentation that communicates model performance across diverse conditions. In parallel, the increasing complexity of ML pipelines has prompted parallel efforts in model and data debugging [131], as even subtle data errors or inductive biases can lead to silent failures in production systems. Stress testing addresses this challenge by exposing brittle regions of the input space and highlighting the conditions under which model generalization deteriorates.

In general, GASTeN represents a foundational step toward scalable, data-driven stress testing and provides a framework that could be extended to domains beyond computer vision, including, as in the present work, for financial time series.

2.4 Summary

This chapter consolidated the theoretical and methodological groundwork for the research, offering a structured overview of the essential components relevant to forecasting financial time series under uncertainty. It began in 2.1 by characterizing financial time series and the complexities inherent in forecasting tasks of varying granularity and horizon. A review of classical and modern supervised learning approaches was provided in 2.1.4, with emphasis on the limitations of traditional statistical techniques and the advantages introduced by nonlinear machine learning models. Evaluation metrics for point forecasts were introduced in 2.1.5, followed by an exploration of uncertainty quantification techniques in 2.1.6, including the role of conformal prediction in providing calibrated prediction intervals. The chapter 2.2.2 then examined generative adversarial networks (GANs) and their potential for generating realistic sequential data in financial contexts. Finally, stress testing methodologies were presented in 2.3 as critical tools for assessing model robustness under atypical conditions, culminating in a review of the GASTeN framework, which serves as a conceptual foundation for the generative stress testing methodology proposed in this

work. Together, these elements establish the conceptual and methodological landscape necessary for developing and evaluating the `GASTeF` framework introduced in the subsequent chapters.

Chapter 3

GASTeF: Generative Adversarial Stress Testing for Forecasting

This chapter formalizes the research problem and presents the proposed methodological framework. Section 3.1 refines the problem formulation introduced in Chapter 1, providing additional context and motivation. Section 3.2 introduces and details the proposed generative architecture, GASTeF, outlining its key components and underlying design rationale.

3.1 Problem Statement

This work proposes a stress testing framework for financial time series forecasting models, leveraging a generative adversarial architecture built on top of the SigCWGAN [90]. The objective is to generate synthetic, yet statistically credible, input sequences that induce high predictive uncertainty or elevated forecasting error in a target model, thereby enabling a systematic stress testing framework for model robustness assessment.

The proposed methodology is guided by the following design desiderata:

- **D1: Statistical plausibility:** Generated sequences should preserve the temporal and distributional characteristics of real financial data to ensure realistic stress scenarios.
- **D2: Uncertainty-driven generation:** The generator is explicitly optimized to generate financial time series sequences that correspond to low-confidence or high-error forecasts from a pre-trained forecasting model, enhancing its utility for the model's stress testing.
- **D3: Flexible stress criteria:** The loss formulation supports modular integration of different uncertainty or error metrics, enabling tailored stress definitions aligned with specific evaluation objectives.

3.1.1 Scope and Application Domain

The methodological scope is defined as follows:

- **Domain:** Multi-step forecasting of daily asset returns across a set of N financial instruments, each represented by its historical price time series. This domain is selected due to its high economic relevance, inherent volatility, and the presence of structural breaks and stress episodes. Sector ETFs offer a diversified yet interdependent representation of market behavior, capturing both broad systemic movements and sector-specific dynamics. These characteristics provide a rich and realistic testing ground for assessing the robustness and generalization capacity of forecasting models under heterogeneous and regime-shifting conditions.

- **Generative Task:** The generator is trained to produce future return sequences that (i) conform to the statistical properties of the real data and (ii) induce high predictive uncertainty or error in the target forecaster.

- **Forecasting Setting:** The forecasting model is pre-trained in a supervised learning setup and held fixed during the generator's training. Forecasting targets consist of multi-step-ahead return vectors for all N assets, conditioned on a historical window of equal length.

- **Evaluation Criteria:** Effectiveness is assessed using uncertainty quantification metrics, degradation of predictive performance, and qualitative inspection of generated stress scenarios.

Furthermore, the framework relies on model-derived uncertainty estimates and does not require access to ground-truth prediction intervals.

By applying GASTeF in this setting, we aim to simulate realistic stress scenarios that challenge model generalization and predictive reliability, thereby offering actionable insights for model validation, risk management, and the development of more resilient decision-support systems in quantitative finance.

3.1.2 Research Hypothesis

This work is guided by the following research hypothesis:

> *Adding an additional uncertainty term into the SigCWGAN [90] generator loss function enables the generation of statistically plausible financial time series that induce higher predictive uncertainty and/or elevated error rates in a pre-trained forecasting model, while preserving the statistical fidelity of the generated data.*

This hypothesis asserts that incorporating an uncertainty-aware loss into the generator's objective enables the synthesis of statistically plausible input sequences that expose regions where the forecasting model exhibits low confidence or increased error. The approach leverages the stability and temporal consistency of SigCWGAN while maintaining architectural simplicity to support scalable stress testing of multistep forecasting models in financial domains.

3.2 GASTeF

This section presents the proposed GASTeF architecture.

3.2.1 Base Architecture: SigCWGAN

At the core of GASTeF lies the SigCWGAN framework [90], which leverages rough path theory [22, 23] and Wasserstein geometry to enable efficient and distributionally consistent generation of high-dimensional, non-stationary sequences. This subsection reviews the essential mechanisms of SigCWGAN [90] to contextualize the architectural and objective-driven extensions introduced in GASTeF.

The SigCWGAN proposed by Ni et al. [90], presents a generative framework tailored for high-dimensional, sequential, multivariate financial time series data. These data are characterized by non-stationarity, volatility clustering, fat tails, and latent data-generating processes, which render both classical parametric simulation (e.g., Monte Carlo methods) and resampling techniques insufficient for realistic and robust scenario generation [123]. Although parametric Monte Carlo methods remain prevalent for applications such as volatility estimation and derivative pricing, they are constrained by model specification risk, limiting their robustness under unforeseen or out-of-distribution scenarios [123]. SigCWGAN addresses these limitations through a signature-based metric and an autoregressive generative architecture that jointly preserves temporal structure and statistical and distributional fidelity without relying on adversarial neural discriminators.

Signature Representation of Time Series. At the core of SigCWGAN lies the use of the *signature transform* from rough path theory, which provides a universal and unique representation of continuous paths [22, 23].

Formally, let $\Omega(J,\mathbb{R}^{1+d})$ denote the space of time-augmented continuous paths, i.e.,

$$\Omega(J,\mathbb{R}^{1+d}) = \left\{ t \mapsto (t,x_t) \mid x \in \mathscr{C}(J,\mathbb{R}^d) \right\},$$

where $J \subset \mathbb{R}$ is a compact time interval and x_t is a continuous, bounded-variation path in $\mathbb{R}^d$, with d representing the number of dimensions (e.g., asset returns). The augmentation with time ensures that the path is uniquely identified up to reparameterization, a crucial property for learning meaningful signatures of sequential data.

The *signature* of a path $x \in \mathscr{C}(J,\mathbb{R}^d)$ over interval $[s,t] \subset J$ is the infinite sequence of iterated integrals:

$$\mathrm{Sig}(x)_{s,t} := \left(1, \int_s^t dx_u, \int_s^t \int_s^{u_1} dx_{u_2} \otimes dx_{u_1}, \dots\right) \in T((\mathbb{R}^d)),$$

where $T((\mathbb{R}^d))$ is the tensor algebra over $\mathbb{R}^d$.

In practice, we truncate the series at level M, yielding the *truncated signature of order M*, denoted $\mathrm{Sig}_M(x) \in \mathbb{R}^{D_M}$:

$$D_M = \sum_{k=0}^{M} d^k = \frac{d^{M+1}-1}{d-1}.$$

where, d represents the input channel dimension, M is the truncation depth, and D_M denotes the total number of terms in the truncated signature, corresponding to the output dimensionality. This formula applies specifically to the implementation in the `signatory` package, which computes the full tensor signature without quotienting by the shuffle relations.

Example for $M = 2$:

$$\mathrm{Sig}_2(x) = \left(1, \underbrace{\int_s^t dx_u}_{\text{Level 1}}, \underbrace{\int_s^t \int_s^{u_1} dx_{u_2} \otimes dx_{u_1}}_{\text{Level 2}}\right).$$

This transformation satisfies two essential properties:

- **Universality:** Any continuous functional on path space can be approximated arbitrarily well by a linear functional of the signature [87].
- **Uniqueness:** The signature uniquely identifies the path up to tree-like equivalence, which in time-augmented paths corresponds to reparameterization invariance [61, 13].

Signature Wasserstein-1 Metric. To compare conditional distributions of real and synthetic future time series, SigCWGAN minimizes the distance between their expected signature representations. Specifically, for a given path segment $\mathbf{x}_{t-p+1:t} \in \mathbb{R}^{p\times d}$ (the conditioning window), let $\mathbf{x}_{t+1:t+q} \in \mathbb{R}^{q\times d}$ denote the corresponding future window. The truncated signature transform of degree M applied to a path $\mathbf{x} \in \mathbb{R}^{\ell\times d}$ is denoted $\mathrm{Sig}_M(\mathbf{x}) \in \mathbb{R}^{D_M}$, where $D_M = \sum_{k=0}^{M} d^k$ is the dimension of the signature space.

To estimate the conditional expectation $\mathbb{E}_\mu[\mathrm{Sig}_{M_2}(\mathbf{x}_{t+1:t+q}) \mid \mathrm{Sig}_{M_1}(\mathbf{x}_{t-p+1:t})]$, we follow [90] and fit a linear projection $\hat{L} \in \mathbb{R}^{D_{M_2}\times D_{M_1}}$ using ridge-regularized least squares:

$$\hat{L} = \arg\min_L \sum_t \left\|\mathrm{Sig}_{M_2}(\mathbf{x}_{t+1:t+q}) - L\cdot \mathrm{Sig}_{M_1}(\mathbf{x}_{t-p+1:t})\right\|_2^2 + \lambda\|L\|_F^2, \tag{3.1}$$

where:

- $\mathbf{x}_{t-p+1:t}$ is the standardized historical (conditioning) window of p time steps;
- $\mathbf{x}_{t+1:t+q}$ is the standardized real future trajectory over q steps;
- $\mathrm{Sig}_{M_1}(\cdot)$ and $\mathrm{Sig}_{M_2}(\cdot)$ denote the truncated signature transforms of order M_1 and M_2, applied to past and future segments, respectively;
- $\lambda > 0$ is a regularization hyperparameter;
- $\|\cdot\|_2$ denotes the Euclidean norm;
- $\|\cdot\|_F$ is the Frobenius norm.

This yields a fixed operator $\hat{L}$ that estimates the conditional expected signature of real future trajectories.

During training, the generator G_θ receives the past window $\mathbf{x}_{t-p+1:t}$ and noise $\mathbf{z} \sim \mathcal{N}(0, I)$ to produce synthetic future samples $\hat{\mathbf{x}}^{(j)}_{t+1:t+q}$. For each training instance, the expected signature of the generated futures is estimated via Monte Carlo sampling (MC):

$$\mathbb{E}_{\nu(\theta, \mathbf{x}_{t-p+1:t})}[\mathrm{Sig}_{M_2}(\hat{\mathbf{x}}_{t+1:t+q})] \approx \frac{1}{N_{\mathrm{MC}}} \sum_{j=1}^{N_{\mathrm{MC}}} \mathrm{Sig}_{M_2}(\hat{\mathbf{x}}^{(j)}_{t+1:t+q}), \tag{3.2}$$

where $\nu(\theta, \mathbf{x}_{t-p+1:t})$ denotes the generator's conditional distribution.

The final loss is the *Signature Wasserstein-1* distance (Sig-W_1), computed as the empirical average of deviations between real and synthetic conditional expectations:

$$\mathscr{L}^{(N)}_{\text{Sig-W1}}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left\| \hat{L} \cdot \mathrm{Sig}_{M_1}(\mathbf{x}^{(i)}_{t-p+1:t}) - \mathbb{E}_{\nu}[\mathrm{Sig}_{M_2}(\hat{\mathbf{x}}^{(i)}_{t+1:t+q})] \right\|_2. \tag{3.3}$$

where:

- N is the number of training instances (e.g., batch size or number of available windows);
- i indexes the individual samples within the batch;
- $\mathbf{x}^{(i)}_{t-p+1:t} \in \mathbb{R}^{p \times d}$ is the standardized conditioning window (past context) for sample i;
- $\hat{L} \in \mathbb{R}^{D_{M_2} \times D_{M_1}}$ is a fixed linear projection matrix estimating the conditional expectation of the future signature given the past.

This metric-based formulation eliminates the need for a neural critic, improves stability, and aligns the conditional laws of generated and real paths in signature space. It aligns with the method introduced in [90], where signature-based metrics are used to align conditional laws of path distributions.

Conditional Autoregressive Generator. The generator G_θ in SigCWGAN is implemented as a *Conditional Autoregressive Feed-Forward Neural Network* (AR-FNN) [90], designed to recursively generate future trajectories conditioned on past observations. Given a historical window $\mathbf{x}_{t-p+1:t} \in \mathbb{R}^{p \times d}$ and a stochastic noise vector $\mathbf{z}_t \sim \mathcal{N}(0, \mathbf{I})$, the generator produces a forecasted path $\hat{\mathbf{x}}^{(t)}_{t+1:t+q} \in \mathbb{R}^{q \times d}$ as:

$$\hat{\mathbf{x}}^{(t)}_{t+1:t+q} = G_\theta(\mathbf{x}_{t-p+1:t}, \mathbf{z}_t). \tag{3.4}$$

Architecturally, G_θ is implemented using a residual multi-layer perceptron (MLP). At each autoregressive step $s = 1, \ldots, q$, the generator concatenates the latent noise $z_s \in \mathbb{R}^m$ with the flattened conditioning history $\mathbf{x}_{t-p+1:t} \in \mathbb{R}^{p \cdot d}$ and feeds the result into a residual feed-forward network. This network consists of a sequence of residual blocks, each comprising a linear transformation followed by a PReLU (Parametric Rectified Linear Unit) activation and optional residual skip connection (when input and output dimensions match). The output $\hat{x}_{t+s}$ is then appended to the history, and the process is repeated recursively for q steps.

This autoregressive generation scheme enables the model to capture temporal dependencies across multiple future time steps, while the residual architecture enhances gradient flow and stability during training. The generator outputs a full sequence $\hat{\mathbf{x}}_{t+1:t+q}$ of predicted returns, conditioned both on historical context and noise, supporting stochastic sampling for distributional forecasting.

Training Procedure. In contrast to conventional GAN architectures that rely on a learned adversarial critic, SigCWGAN employs a metric-based training objective grounded in pathwise signature features. Specifically, the generator G_θ is trained to minimize the discrepancy between the expected signatures of real and generated future trajectories, conditioned on the same historical context. This approach leverages the expressive power and time-reparameterization invariance of the signature transform mentioned previously, enabling robust modeling of sequential structure and long-range dependencies.

The training objective is defined via the Signature Wasserstein-1 distance:

$$\mathscr{L}_G = \mathscr{L}_{\text{Sig-W1}}(\theta), \tag{3.5}$$

where $\mathscr{L}_{\text{Sig-W1}}(\theta)$ quantifies the expected ℓ_2-distance between the future signature predicted by a linear estimator $\hat{L} \cdot \text{Sig}_{M_1}(\bar{\mathbf{x}}_{t-p+1:t})$ and the empirical expected signature of samples generated from G_θ. The conditional expectation is estimated via Monte Carlo sampling of synthetic futures, as described in Algorithm 1 (adapted from [90]).

Algorithm 1 Training Procedure of SigCWGAN (adapted from [90])

Input: Standardized time series $\bar{\mathbf{X}} \in \mathbb{R}^{T\times d}$, signature orders M_1, M_2, window lengths of past path$p$ and future path q, learning rate η, batch size B, number of epochs N, and number of Monte Carlo samples N_{MC}.

Output: Generator parameters θ

Preprocessing:

Segment $\bar{\mathbf{X}}$ into windows: $\bar{X}_{t-p+1:t}$ and $\bar{X}_{t+1:t+q}$

Compute $\text{Sig}_{M_1}(\bar{X}_{t-p+1:t})$ and $\text{Sig}_{M_2}(\bar{X}_{t+1:t+q})$

Fit regression map $\hat{L}$ from past to future signatures

Training Loop:

for epoch $i = 1,\dots,N$ **do**

  for each minibatch $\mathscr{B} = \{t_1,\dots,t_B\}$ **do**

    Initialize loss $\mathscr{L}_{\mathscr{B}}(\theta) \leftarrow 0$

    for each $t \in \mathscr{B}$ **do**

      Sample N_{MC} futures: $\hat{X}^{(j)}_{t+1:t+q} \sim G_\theta(\bar{X}_{t-p+1:t})$

      Compute empirical mean of future signatures:

$$\mathbb{E}[\text{Sig}_{M_2}] = \frac{1}{N_{\text{MC}}}\sum_{j=1}^{N_{\text{MC}}} \text{Sig}_{M_2}(\hat{X}^{(j)}_{t+1:t+q})$$

      Accumulate loss:

$$\mathscr{L}_{\mathscr{B}}(\theta) \mathrel{+}= \left\| \hat{L}\cdot \text{Sig}_{M_1}(\bar{X}_{t-p+1:t}) - \mathbb{E}[\text{Sig}_{M_2}] \right\|^2$$

    end for

    Update parameters: $\theta \leftarrow \theta - \eta \nabla_\theta \mathscr{L}_{\mathscr{B}}(\theta)$

  end for

end for

return θ

This framework eliminates the need for a trainable neural discriminator, thereby improving training stability and computational efficiency. By aligning conditional expected signatures, SigCWGAN directly matches the statistical properties of real and synthetic paths in feature space, preserving both distributional structure and temporal dynamics [90, 86]. Its use of signature features—grounded in rough path theory—provides a mathematically rigorous similarity measure that is invariant to time reparameterizations and capable of capturing higher-order temporal dependencies [22, 78].In addition, SigCWGAN offers scalability to high-dimensional, multivariate time series and supports long-range forecasting.

In summary, SigCWGAN offers a robust and theoretically grounded framework for learning conditional distributions over future trajectories in time series data. This makes it particularly suit-

able for applications in financial forecasting and stress testing, where preserving temporal structure and tail behavior is essential. The entire SigCWGAN pipeline is implemented in PyTorch, with efficient signature computation provided by the `signatory` library [78].[1]

3.2.2 Uncertainty-Aware Extension in GASTeF

To adapt this architecture for stress testing a given forecasting model, we extend the base SigCWGAN model into a framework denoted GASTeF. Figure 3.1 provides a schematic overview of the GASTeF architecture.

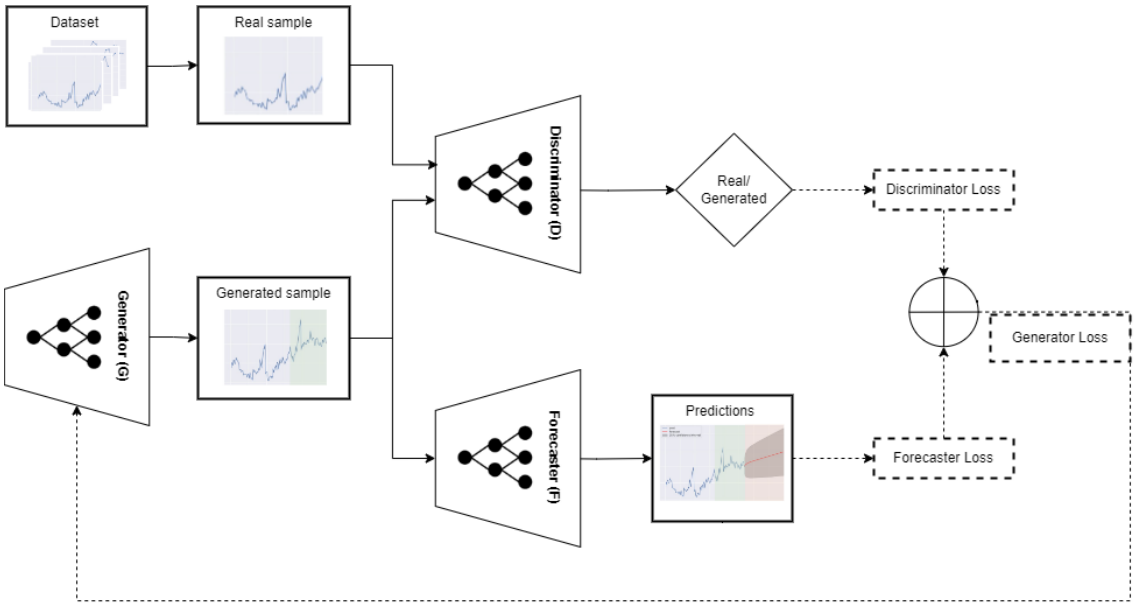

Figure 3.1: Overview of the GASTeF architecture. The generator (G) produces synthetic futures that are evaluated by a discriminator (D) for realism and by a fixed probabilistic forecaster (F) for predictive behavior. Dashed arrows denote loss signals aggregated in the generator objective (signature Wasserstein, uncertainty-aware term, and auxiliary components).

The extensions are structured along three axes: (i) a probabilistic forecasting model, (ii) an uncertainty-aware loss function, and (iii) a modified training procedure.

Probabilistic Forecasting Model. To evaluate the robustness of forecasting predictions under adversarial stress scenarios, GASTeF integrates a pre-trained probabilistic forecasting model f_ϕ to predict conditional prediction intervals of future trajectories. By incorporating the outputs of f_ϕ into the generator's loss function, the GAN is guided to produce synthetic stress scenarios that deliberately perturb the input distribution of the forecasting model, enabling a systematic assessment of how predictive confidence and performance deteriorate under adverse or distributionally shifted conditions.

[1] For reproducibility and implementation details, we refer the reader to the official repository: https://github.com/SigCGANs/Conditional-Sig-Wasserstein-GANs.

Formally, for a generated sequence $\hat{\mathbf{x}}^{(i)}_{t+1:t+q} \in \mathbb{R}^{q\times d}$, the model produces lower and upper predictive bounds:

$$\left[\underline{\mathbf{y}}^{(i)}, \overline{\mathbf{y}}^{(i)}\right] = f_\phi(\hat{\mathbf{x}}^{(i)}_{t+1:t+q}), \tag{3.6}$$

where f_ϕ is implemented as a two-headed feed-forward neural network trained via quantile regression. The training objective minimizes the pinball loss at a confidence level $\alpha \in (0,1)$, resulting in an estimated prediction interval:

$$\mathscr{L}_{\text{quantile}} = \mathbb{E}\left[\rho_{\alpha/2}\left(\underline{\mathbf{y}} - \mathbf{y}\right) + \rho_{1-\alpha/2}\left(\mathbf{y} - \overline{\mathbf{y}}\right)\right], \tag{3.7}$$

where $\rho_q(u) = \max\{(q-1)u, qu\}$ is the standard asymmetric quantile (pinball) loss.

This formulation avoids distributional assumptions such as as the Gaussian distribution, which often fail in financial contexts where return distributions exhibit heavy tails and skewness. In addition, each bound is estimated independently, allowing the model to capture asymmetric uncertainty patterns, an essential property for identifying directional vulnerabilities and tail risks under synthetic stress scenarios. Once trained, f_ϕ is held fixed and used solely to evaluate the uncertainty of generated sequences.

This design allows GASTeF to stress test the forecasting model by actively generating inputs that yield wider predictive intervals or challenge the model's predictive confidence, thereby exposing vulnerabilities in the model's generalization to distributional shifts or rare scenarios.

Uncertainty-Aware Loss Function. To enable stress testing of predictive models, we incorporate an auxiliary loss term that encourages the generator to produce samples which elicit high epistemic uncertainty from a fixed probabilistic forecasting model. Given a generated trajectory $\hat{\mathbf{x}}^{(i)}_{t+1:t+q} \in \mathbb{R}^{q\times d}$, we pass it through a frozen pre-trained forecasting model f_ϕ that outputs prediction intervals:

$$[\underline{\mathbf{y}}^{(i)}, \overline{\mathbf{y}}^{(i)}] = f_\phi\left(\hat{\mathbf{x}}^{(i)}_{t+1:t+q}\right). \tag{3.8}$$

The auxiliary uncertainty-aware term is designed to reward the generator when the forecasting model exhibits wide prediction intervals:

$$\mathscr{L}_{\text{uncertainty}} = \lambda \cdot \mathbb{E}_i\left[\left\|\overline{\mathbf{y}}^{(i)} - \underline{\mathbf{y}}^{(i)}\right\|_1\right], \tag{3.9}$$

where $\lambda \geq 0$ is a tunable hyperparameter that controls the influence of uncertainty. This term does not penalize uncertainty, but rather incentivizes the generation of inputs that lead to increased predictive dispersion.

To balance the contribution of this term with respect to the main signature matching loss $\mathscr{L}_{\text{Sig-W1}}$, we apply a normalization factor:

$$\tilde{\mathscr{L}}_{\text{uncertainty}} = \lambda \cdot \frac{\delta}{\mathscr{L}_{\text{Sig-W1}} + \varepsilon} \cdot \mathbb{E}_i\left[\left\|\overline{\mathbf{y}}^{(i)} - \underline{\mathbf{y}}^{(i)}\right\|_1\right], \tag{3.10}$$

where δ is a predefined baseline signature loss and $\varepsilon \ll 1$ ensures numerical stability.

Additionally, to prevent the generator from trivially increasing uncertainty by deviating from the data distribution realism, we include a penalty term:

$$\mathscr{L}_{\text{penalty}} = \lambda^2 \cdot \kappa \cdot \max\left(0, \frac{\mathscr{L}_{\text{Sig-W1}}}{\delta} - 1\right), \tag{3.11}$$

where $\kappa > 0$ penalizes synthetic sequences whose signatures fall below the baseline plausibility threshold. The inclusion of λ^2 in the penalty formulation causes the regularization strength to increase more rapidly as the uncertainty emphasis grows. This promotes training stability by discouraging the generator from producing excessive deviations from the true data distribution, even under high stress-test scenarios.

The total generator objective is thus composed of three components:

1. The **signature matching loss** $\mathscr{L}_{\text{Sig-W1}}$, which aligns the expected pathwise signature of generated sequences with those of real data using the Wasserstein-1 metric in signature space.

2. The **uncertainty-aware reward** $\mathscr{L}_{\text{uncertainty}}$, which encourages the generation of inputs that induce large predictive intervals from the forecasting model, signaling epistemic uncertainty.

3. A **distributional plausibility penalty** $\mathscr{L}_{\text{penalty}}$, which regularizes the generator by penalizing unrealistic trajectories whose signature statistics deviate excessively from the empirical baseline.

Formally, the full generator objective in GASTeF is given by:

$$\mathscr{L}_{\text{GASTeF}} = \mathscr{L}_{\text{Sig-W1}} - \tilde{\mathscr{L}}_{\text{uncertainty}} + \mathscr{L}_{\text{penalty}}. \tag{3.12}$$

This revised training objective balances three competing goals: (i) preserving distributional fidelity via the signature matching loss, (ii) discovering high-uncertainty input regions that expose forecasting vulnerabilities, and (iii) maintaining training stability through penalized divergence. Overall, the generation of statistically realistic but structurally adverse scenarios that maximize epistemic uncertainty in the predictive model aligns with the goals of stress testing. It enables the construction of adversarial-but-realistic scenarios useful for risk evaluation and stress testing in financial forecasting systems.

GASTeF Training Procedure. To enable the adversarial generation of stress-testing scenarios, we extend the baseline SigCWGAN optimization loop (see Algorithm 1) by incorporating an uncertainty-aware regularizer into the generator's training objective. This extension drives the generator to produce trajectories that are both statistically consistent with historical data and diagnostically informative, in the sense that they elicit heightened epistemic uncertainty or predictive degradation from a downstream forecasting model.

The forecasting model f_ϕ, which is used solely for uncertainty quantification during GAN training, is implemented as a quantile-based predictor trained to estimate prediction intervals at a given confidence level $1-\alpha$. The model is pretrained independently on real data using the pinball loss (see Algorithm 2), capturing distributional asymmetries and heavy-tailed behavior without relying on parametric assumptions (e.g., Gaussian distributions). Once trained, the forecasting model remains frozen and non-trainable during generator training, thereby serving as a diagnostic tool to assess the epistemic uncertainty elicited by synthetic sequences.

Algorithm 2 Pretraining Procedure of the Probabilistic Forecasting Model

Input: Real future trajectories $\mathbf{X}^{(i)}_{t+1:t+q} \in \mathbb{R}^{q\times d}$, confidence level $\alpha \in (0,1)$, learning rate η, number of epochs N, forecasting model f_ϕ with outputs $[\underline{\mathbf{y}}, \overline{\mathbf{y}}]$

Output: Trained parameters ϕ of forecasting model

Training Loop:

for epoch $e = 1, \ldots, N$ **do**
  for each minibatch $\mathscr{B} = \{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(B)}\}$ **do**
    Flatten each $\mathbf{X}^{(i)} \in \mathbb{R}^{q\times d}$ to a vector $\mathbf{y}^{(i)} \in \mathbb{R}^{qd}$
    Forward pass through model: $[\underline{\mathbf{y}}^{(i)}, \overline{\mathbf{y}}^{(i)}] = f_\phi(\mathbf{y}^{(i)})$
    Compute quantile targets:

$$\mathscr{L}_{\text{quantile}} = \sum_{i=1}^{B} \rho_{\alpha/2}(\underline{\mathbf{y}}^{(i)} - \mathbf{y}^{(i)}) + \rho_{1-\alpha/2}(\mathbf{y}^{(i)} - \overline{\mathbf{y}}^{(i)})$$

    where $\rho_q(u) = \max((q-1)\cdot u, q\cdot u)$
    Backpropagate and update parameters: $\phi \leftarrow \phi - \eta \nabla_\phi \mathscr{L}_{\text{quantile}}$
  end for
end for
Freeze model parameters ϕ for downstream GASTeF training
return ϕ

Algorithm 3 outlines the complete training procedure of the GASTeF model. It extends the baseline SigCWGAN loop (cf. Algorithm 1) by introducing an uncertainty-aware loss and a plausibility-preserving penalty to guide adversarial scenario generation for stress testing.

The training process is structured in two main phases: (i) *preprocessing and model initialization*, and (ii) *adversarial generator training*.

In the preprocessing stage (lines 1–4), the standardized time series $\bar{\mathbf{X}} \in \mathbb{R}^{T\times d}$ is segmented into overlapping input-output windows of length p (past) and q (future). Signature transforms of both segments are computed using truncation orders M_1 and M_2, respectively. A linear regression map $\hat{L}$ is then fitted to predict future signatures from past ones. Simultaneously, the probabilistic forecasting model f_ϕ is pretrained using quantile regression on real data (Algorithm 2) and remains frozen during generator training.

In the main training loop (lines 5–22), the generator G_θ is optimized to minimize the GASTeF objective, which balances three components:

- **Signature matching loss:** At each timestep $t \in \mathscr{B}$, the generator produces N_{MC} future samples $\hat{X}^{(j)}_{t+1:t+q}$. The expected future signature $\mathbb{E}[\text{Sig}_{M_2}]$ is compared against the target signature $\hat{L} \cdot \text{Sig}_{M_1}(\bar{X}_{t-p+1:t})$ to compute the primary loss $\mathscr{L}_{\text{Sig-W1}}$.
- **Uncertainty-aware reward:** Each generated trajectory is passed through the fixed forecasting model f_ϕ, which returns interval bounds $[\underline{\mathbf{y}}^{(j)}, \overline{\mathbf{y}}^{(j)}]$. The L1-width of the prediction interval is scaled by $\lambda \cdot \delta/(\mathscr{L}_{\text{Sig-W1}} + \varepsilon)$, forming the normalized reward term $\mathscr{L}_{\text{uncertainty}}$. This term promotes the generation of trajectories that elicit high epistemic uncertainty.
- **Plausibility penalty:** To discourage the generator from deviating too far from realistic data regimes, a penalty is imposed when $\mathscr{L}_{\text{Sig-W1}} > \delta$. This term is weighted by $\lambda^2 \cdot \kappa$, ensuring greater regularization pressure at higher uncertainty weights.

The total generator loss is then computed as:

$$\mathscr{L}_{\texttt{GASTeF}} = \mathscr{L}_{\text{Sig-W1}} - \tilde{\mathscr{L}}_{\text{uncertainty}} + \mathscr{L}_{\text{penalty}},$$

which is minimized with respect to θ using gradient descent (line 21). The subtraction of the uncertainty reward reflects the objective of generating diagnostically informative sequences that maximize model stress.

This formulation enables GASTeF to synthesize statistically realistic yet adversarial trajectories that expose vulnerabilities in fixed forecasting pipelines, making it a principled tool for scenario-based financial stress testing.

Algorithm 3 Training Procedure of GASTeF

Input: Standardized time series $\bar{\mathbf{X}} \in \mathbb{R}^{T \times d}$, signature orders M_1, M_2, window lengths p, q, learning rate η, batch size B, number of epochs N, Monte Carlo samples N_{MC}, baseline signature loss δ, uncertainty weight λ, penalty coefficient κ, pretrained probabilistic forecasting model f_ϕ.

Output: Trained generator parameters θ

Preprocessing:

Segment $\bar{\mathbf{X}}$ into windows: $\bar{X}_{t-p+1:t}$ and $\bar{X}_{t+1:t+q}$

Compute $\text{Sig}_{M_1}(\bar{\mathbf{x}}_{t-p+1:t})$, $\text{Sig}_{M_2}(\bar{\mathbf{x}}_{t+1:t+q})$

Fit regression map $\hat{L}$ from past to future signatures

Pretrain forecasting model f_ϕ on real data (see Algorithm 2)

Training Loop:

for epoch $i = 1, \ldots, N$ **do**

  for minibatch $\mathscr{B} = \{t_1, \ldots, t_B\}$ **do**

    Initialize loss terms: $\mathscr{L}_{\text{Sig-W1}} \leftarrow 0$, $\tilde{\mathscr{L}}_{\text{uncertainty}} \leftarrow 0$, $\mathscr{L}_{\text{penalty}} \leftarrow 0$

    for each $t \in \mathscr{B}$ **do**

      Sample N_{MC} futures: $\hat{X}^{(j)}_{t+1:t+q} \sim G_\theta(\bar{X}_{t-p+1:t})$

      Compute empirical mean of future signatures:

$$\mathbb{E}[\text{Sig}_{M_2}] = \frac{1}{N_{\text{MC}}} \sum_{j=1}^{N_{\text{MC}}} \text{Sig}_{M_2}(\hat{\mathbf{x}}^{(j)}_{t+1:t+q})$$

      Accumulate signature matching loss:

$$\mathscr{L}_{\text{Sig-W1}} \mathrel{+}= \left\| \hat{L} \cdot \text{Sig}_{M_1}(\bar{X}_{t-p+1:t}) - \mathbb{E}[\text{Sig}_{M_2}] \right\|^2$$

      for each synthetic future $\hat{\mathbf{x}}^{(j)}_{t+1:t+q}$ **do**

        Flatten and pass through f_ϕ to obtain intervals $[\underline{\mathbf{y}}^{(j)}, \overline{\mathbf{y}}^{(j)}]$

        Accumulate uncertainty-aware loss:

$$\mathscr{L}_{\text{uncertainty}} \mathrel{+}= \frac{\lambda \cdot \delta}{\mathscr{L}_{\text{Sig-W1}} + \varepsilon} \cdot \left\| \overline{\mathbf{y}}^{(j)} - \underline{\mathbf{y}}^{(j)} \right\|_1$$

      end for

    end for

    Combine total generator loss:

$$\mathscr{L}_{\text{GASTeF}} = \mathscr{L}_{\text{Sig-W1}} - \mathscr{L}_{\text{uncertainty}} + \mathscr{L}_{\text{penalty}}$$

    Update generator parameters:

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathscr{L}_{\text{GASTeF}}$$

  end for

end for

return θ

3.3 Summary

This chapter introduced GASTeF, a generative adversarial framework for stress testing multivariate financial time series forecasting models. The problem was formally defined in section 3.1 as the generation of statistically realistic input sequences that systematically induce high epistemic uncertainty or elevated forecast error in a fixed predictive model. To this end, the framework builds upon the Conditional Signature Wasserstein GAN (SigCWGAN) [90] and extends it with an uncertainty-aware loss component that promotes the generation of diagnostically informative stress scenarios. A probabilistic forecasting model, implemented as a quantile regressor, is pretrained on real data and integrated into the generator's training loop as a frozen diagnostic mechanism. The complete training procedure, including signature matching, uncertainty amplification, and plausibility regularization, was formalized and described in algorithmic detail in section 3.2. This approach enables targeted exploration of forecasting model vulnerabilities under adverse yet statistically credible conditions, supporting systematic validation, robustness analysis, and risk-aware model deployment in financial time series applications.

Chapter 4

Experimental Setup and Results

This chapter presents the empirical evaluation of the proposed `GASTeF` framework, detailing both the experimental setup and the resulting analysis. The objective is to assess the framework's ability to generate statistically plausible yet adversarial financial time series that expose weaknesses in probabilistic forecasting models. Section 4.1 introduces the dataset of sector ETF returns, including the input features, temporal partitioning scheme, and preprocessing pipeline. Section 4.2 outlines the experimental design, baseline configuration, hyperparameter settings, training protocol, and implementation environment. Section 4.3 defines the evaluation metrics used to measure both the statistical fidelity of the generated data and the degradation in forecasting performance under synthetic stress conditions. The subsequent subsections, 4.4.1 and 4.4.2, present the main results: the first examines the statistical quality of generated sequences—covering marginal distributional alignment, temporal dependencies, and structural coherence—while the second evaluates the stress-testing effectiveness of `GASTeF` through predictive degradation and uncertainty analysis. Finally, subsection 4.4.3 synthesizes these findings, highlighting the trade-off between data realism and adversarial strength, and assessing the overall robustness and practical relevance of the framework.

4.1 Dataset

4.1.1 Dataset description

This study utilizes a multivariate time series dataset comprising daily log returns from nine sector-based Exchange-Traded Funds (ETFs), each representing a distinct economic sector within the S&P 500 index. The raw price data were retrieved from Yahoo Finance [160], covering the period from January 3, 2000, to December 31, 2023. The complete dataset spans over two decades, encompassing various market regimes, including periods of expansion, recession, and systemic stress. The presence of extreme events is not incidental but rather a desired feature of the data, as it enables the assessment of the model's ability to capture tail behavior and distributional shifts. This is particularly relevant for the SigCWGAN framework, which explicitly addresses the known limitations of generative models in replicating the tails of financial return distributions [90].

The ETF tickers and their corresponding sector designations are listed in Table 4.1. Let $X_t \in \mathbb{R}^d$ denote the vector of daily log returns at time t, where $d = 9$ corresponds to the number of sector ETFs included in the study.

Table 4.1: Constituent Sector ETFs of the S&P 500 used in the `GASTeF` Framework

Ticker	**Sector Name**
XLE	Energy
XLU	Utilities
XLK	Technology
XLB	Materials
XLP	Consumer Staples
XLY	Consumer Discretionary
XLI	Industrials
XLV	Health Care
XLF	Financials

4.1.2 Temporal Partitioning

To ensure a realistic and robust evaluation of the generative model, the dataset is partitioned using a chronologically consistent, rolling window approach. Specifically, standardized log return sequences are segmented into fixed-length paths of length $p+q$ (condition and prediction windows), resulting in a tensor $\mathbf{X} \in \mathbb{R}^{N\times(p+q)\times d}$, where N denotes the number of non-overlapping samples and $d = 9$ is the number of ETF sectors.

This collection of sequences is then split into training and testing subsets using a temporal 80/20 split:

$$\mathbf{X}_{\text{train}} = \mathbf{X}_{1:\lfloor 0.8N \rfloor}, \quad \mathbf{X}_{\text{test}} = \mathbf{X}_{\lfloor 0.8N \rfloor + 1:N}, \tag{4.1}$$

ensuring that training samples precede test samples in time, thus avoiding data leakage and preserving causal structure. This setup ensures the statistical integrity of evaluation under a realistic stress-testing framework, particularly when evaluating time-dependent properties such as autocorrelation, cross-correlation, and signature similarity.

4.1.3 Preprocessing Pipeline

Let $P_t \in \mathbb{R}^d$ denote the adjusted closing price vector of $d = 9$ sector ETFs at time t, and define the daily log return as:

$$X_t = \log(P_t) - \log(P_{t-1}) = \log\left(\frac{P_t}{P_{t-1}}\right). \tag{4.2}$$

The resulting log return matrix is denoted $\mathbf{X}_{\text{raw}} \in \mathbb{R}^{T\times d}$, where T is the total number of trading days (time steps) and $d = 9$ is the number of sector ETF features.

To normalize scale differences across time and variables, we apply a temporal standardization step using `StandardScalerTS`, which computes the global mean and standard deviation over all time steps and features:

$$\mu = \frac{1}{T \cdot d} \sum_{t=1}^{T} \sum_{j=1}^{d} X_{t,j}, \quad \sigma = \sqrt{\frac{1}{T \cdot d} \sum_{t=1}^{T} \sum_{j=1}^{d} (X_{t,j} - \mu)^2}. \tag{4.3}$$

The standardized log return series is given by $\bar{X}_{t,j} = \frac{X_{t,j} - \mu}{\sigma}$, yielding a matrix $\bar{\mathbf{X}} \in \mathbb{R}^{T \times d}$.

Following standardization, the data is segmented into rolling windows of length $p+q$, resulting in a tensor $\bar{\mathbf{X}}_{\text{windowed}} \in \mathbb{R}^{N \times (p+q) \times d}$, where N is the number of extracted samples.

Each sample is structured as:

- **Input (conditioning window):** $\bar{X}_{t-p+1:t} \in \mathbb{R}^{p \times d}$, the standardized past log returns;
- **Target (prediction window):** $\bar{X}_{t+1:t+q} \in \mathbb{R}^{q \times d}$, the standardized future returns.

Overall, this setup supports conditional generative modeling by enabling the learning of mappings from short-term temporal contexts to plausible future sequences, while maintaining statistical coherence and facilitating efficient learning. It also provides a consistent interface for evaluating forecasting models under standardized input conditions, enhancing both interpretability and generalization of results.

4.2 GASTeF Experiments

This section outlines the experimental setup used to validate the `GASTeF` framework described in Section 3.2. It includes a detailed description of the baseline model configuration (Section 4.2.1), hyperparameter settings, training protocol, and implementation environment (Section 4.2.2), establishing the reproducible conditions under which the subsequent evaluation will be conducted.

4.2.1 Baseline Model

To rigorously quantify the incremental contribution of the uncertainty-aware regularization and plausibility penalty in the `GASTeF` framework, we define a baseline model that preserves the overall architecture, including the use of a fixed probabilistic forecasting model, but excludes the uncertainty and penalty components from the generator's training objective. Specifically, the generator is optimized solely to align synthetic trajectories with the statistical properties of the real data, as measured by the signature-based Wasserstein-1 distance. It is not explicitly incentivized to induce predictive uncertainty or forecast degradation.

Formally, the generator in the baseline model is trained to minimize only the signature-based loss:

$$\mathscr{L}_{\text{baseline}} = \mathscr{L}_{\text{Sig-W1}}, \tag{4.4}$$

where $\mathscr{L}_{\text{Sig-W1}}$ denotes the conditional signature Wasserstein-1 loss described in Section 3.2.

The probabilistic forecasting model f_ϕ is independently pretrained on real data using quantile regression and remains fixed throughout GAN training, consistent with the full `GASTeF` setup. However, in this baseline configuration, f_ϕ plays no role in shaping the generator's training dynamics; it is employed solely at evaluation time to measure the predictive dispersion and performance degradation caused by the generated sequences.

This setup functions as a control condition, isolating the effects of the proposed uncertainty-aware regularization. Comparative evaluation across the baseline and `GASTeF` variants is conducted along the following axes:

- **Statistical plausibility:** the degree to which generated trajectories preserve the temporal, distributional, and structural properties of real financial data.
- **Epistemic stress potential:** the ability of generated samples to induce elevated predictive uncertainty in the downstream forecasting model.
- **Forecast performance degradation:** the extent to which generated inputs degrade the accuracy or reliability of the forecaster under synthetic stress conditions.

By benchmarking against this reduced variant, we empirically validate whether the integration of uncertainty-aware loss terms enhances the adversarial stress-testing capability of the model, without compromising generative fidelity. This baseline thus defines the lower bound of epistemic impact achievable under the `GASTeF` architecture in the absence of explicit stress-inducing objectives.

4.2.2 Hyperparameter Settings and Training Protocol

All models were implemented in `PyTorch` (v2.0) and executed with CUDA acceleration (v11.8). Experiments were conducted using a single random seed to ensure reproducibility. The training protocol is organized into the following components:

Training Configuration:

- **Optimizer:** Adam optimizer with learning rate $\eta = 10^{-3}$
- **Batch size:** 200 samples per iteration
- **Training epochs:** 1,000 steps (no early stopping)
- **Historical and prediction horizon:** $p = q = 3$ days

Signature Feature Configuration:

- **Signature depth:** $M_1 = M_2 = 2$ (for past and future)
- **Monte Carlo samples:** $N_{\text{MC}} = 1000$ per forecast window

Forecasting Model (Quantile Regressor):

- **Training epochs:** 100 (fixed)
- **Learning rate:** 10^{-3}
- **Confidence levels:** $1-\alpha \in \{85\%, 90\%, 95\%\}$, i.e., $\alpha \in \{0.15, 0.10, 0.05\}$
- **Loss function:** Pinball loss applied independently at lower and upper quantiles

`GASTeF`-Specific Hyperparameters:

- **Uncertainty loss weight:** $\lambda \in \{0.0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06\}$
- **Signature matching baseline:** $\delta = 0.5$
- **Penalty coefficient:** κ (fixed across runs, tuned via cross-validation)

Unless stated otherwise, these hyperparameter values are used consistently across all experimental conditions to enable fair comparisons between the baseline and uncertainty-aware variants of the `GASTeF` model.

4.3 Evaluation Metrics

This section presents the primary evaluation metrics employed to assess the performance of the proposed approach.

4.3.1 Synthetic Time Series Quality

The selection of appropriate evaluation metrics for GANs remains an open research question, particularly in the context of time-series data [15, 16]. While most existing literature focuses on evaluating GANs for image generation tasks [121, 60, 65, 62, 152], the assessment of synthetic time-series data introduces unique challenges. Specifically, a comprehensive evaluation must account not only for the fidelity of the marginal (static) distributions but also for the preservation of temporal dependencies and inter-variable dynamics intrinsic to the data-generating process. To this end, we adopt a multifaceted evaluation framework designed to reflect three core desiderata for high-quality generative models (diversity, fidelity, and usefulness), drawing inspiration from the evaluation methodology proposed in TimeGAN [162]:

- **Diversity:** Reflects the extent to which generated samples capture the full variability of the real data distribution. This is evaluated using statistical moment-based metrics, including absolute histogram distance, skewness, and kurtosis, which collectively assess marginal distributional alignment and tail behavior.

- **Fidelity:** Measures the statistical indistinguishability of synthetic data from real data. It is quantified via the discriminative score, alongside temporal and cross-sectional similarity metrics such as lag-1 autocorrelation and inter-variable cross-correlation.

- **Usefulness:** Assesses the utility of synthetic data in downstream tasks, particularly forecasting. The predictive score evaluates whether models trained on real data generalize to generated outputs, while the signature metric captures the preservation of complex sequential dynamics.

By jointly considering these aspects—diversity, fidelity, and usefulness—our evaluation strategy provides a set of complementary metrics to evaluate the quality of the generated time series. Each metric is formally defined and discussed as follows. The implementation of these metrics is based on the evaluation framework proposed in SigCWGAN [90, 30].

- **Discriminative Score**: For a quantitative measure of similarity between real and generated returns, we adopt a post-hoc time-series classification approach. Specifically, a two-layer LSTM network is trained in a supervised setting to classify real versus generated sequences by minimizing cross-entropy loss. Let Acc_{test} be the accuracy on a held-out test set. The discriminative score is computed as:

$$\text{Discriminative Score} = \left|0.5 - \text{Acc}_{\text{test}}\right|. \tag{4.5}$$

  Interpretation: A lower score indicates greater indistinguishability between real and synthetic returns, suggesting high distributional similarity.

- **Predictive Score**: To assess the model's ability to preserve input-output relationships, we compute a predictive consistency metric based on linear regression [162, 42]. Specifically, a linear regression model is trained on real input-output pairs $(X_{t-p+1:t}, X_{t+1})$, where $X_{t-p+1:t} \in \mathbb{R}^{p\times d}$ is a window of past p time steps of a d-dimensional time series, and $X_{t+1} \in \mathbb{R}^d$ is the subsequent time step. The quality of this fit on real returns is evaluated using the coefficient of determination, denoted as R^2_{TRTR}, which stands for *Train on Real, Test on Real*. This value reflects the predictive power of a model trained and evaluated on real returns. To evaluate whether the same input-output relationship is preserved in the generated returns, the trained linear model is applied to synthetic outputs $\hat{X}_{t+1}$ produced by feeding real inputs $X_{t-p+1:t}$ into the GAN generator. The resulting R^2 score is denoted as R^2_{TSTR}, standing for *Train on Synthetic, Test on Real*, indicating that the model's predictions on generated returns are being evaluated against the real underlying process. The predictive score is then computed as the absolute difference between the two coefficients of determination:

$$\text{Predictive Score} = \left|R^2_{\text{TRTR}} - R^2_{\text{TSTR}}\right|. \tag{4.6}$$

Interpretation: A lower predictive score suggests that the generative model has successfully captured the underlying temporal structure and predictive relationships present in the real returns, suggesting high temporal consistency.

- **Absolute Histogram Distance**: This metric compares the marginal distributions of real and generated returns using the absolute difference between empirical histograms, and averaged across all dimensions. For each dimension d, histograms are computed using a fixed number of bins, and the loss is given by:

$$\text{Absolute Histogram Distance} = \frac{1}{D}\sum_{i=1}^{D}\left\|\hat{P}_i - P_i\right\|_1, \tag{4.7}$$

where P_i and $\hat{P}_i$ are the normalized histograms of the real and generated returns in dimension i, respectively. This metric reflects how well the model replicates the marginal distributions.

Interpretation: A lower value indicates that the model accurately reproduces the empirical marginal distributions of the real time series, suggesting high distributional similarity.

- **Autocorrelation at Lag 1**: To assess whether the temporal dependencies are preserved, we compute the absolute difference in the in lag-1 autocorrelation coefficients between real and generated returns, averaged over all time series dimensions:

$$\text{Autocorrelation at Lag 1} = \frac{1}{D}\sum_{i=1}^{D}\left|\rho_i^{(1)} - \hat{\rho}_i^{(1)}\right|, \tag{4.8}$$

where $\rho_i^{(1)}$ and $\hat{\rho}_i^{(1)}$ denote the lag-1 autocorrelations for dimension i in the real and generated returns, respectively. This metric is crucial for validating short-term temporal dynamics.

Interpretation: A lower value suggests that the generative model has successfully captured short-term temporal dependencies present in the real returns.

- **Skewness Error**: Skewness measures the asymmetry of a distribution. For each time series, skewness is computed as:

$$\text{Skewness}(X) = \frac{\mathbb{E}[(X-\mu)^3]}{\sigma^3}, \tag{4.9}$$

where μ is the mean and σ is the standard deviation. The metric reports the absolute difference in skewness between real and generated data, capturing asymmetries in return distributions:

$$\text{Skewness Error} = \left|\text{Skewness}(X) - \text{Skewness}(\hat{X})\right|. \tag{4.10}$$

Interpretation: A low skewness error indicates that the GAN can accurately model asymmetric properties of the return distributions.

- **Kurtosis Error**: Kurtosis quantifies the "tailedness" of the distribution. The (excess) kurtosis is defined as:

$$\text{Kurtosis}(X) = \frac{\mathbb{E}[(X-\mu)^4]}{\sigma^4} - 3. \tag{4.11}$$

This metric compares the excess kurtosis of real and synthetic data:

$$\text{Kurtosis Error} = \left|\text{Kurtosis}(X) - \text{Kurtosis}(\hat{X})\right|, \tag{4.12}$$

which is critical for understanding how well the model captures heavy tails and extreme events.

Interpretation: A smaller kurtosis error implies that the generative model captures the heavy tails or extremal behavior typical of financial returns.

- **Cross-Correlation**: For multivariate time series, the model should preserve the interdependencies across dimensions. Let C and $\hat{C}$ be the empirical cross-correlation matrices of the real and generated data, respectively. The metric measures the total deviation in off-diagonal correlations:

$$\text{Cross\_correlation} = \sum_{i<j} \left|C_{ij} - \hat{C}_{ij}\right|, \tag{4.13}$$

where C_{ij} denotes the lag-0 correlation between dimension i and j. This metric evaluates the model's ability to reproduce cross-sectional relationships.

Interpretation: A low cross-correlation error indicates that the model effectively reproduces the inter-asset dependencies present in the original dataset, which is essential for multivariate financial modeling.

- **Signature Metric:** The signature of a path, denoted by $S(\cdot)$, is a feature map from the space of continuous paths into a tensor algebra, capturing the ordered structure and higher-order interactions of the time series over a fixed window. It provides a faithful and compact representation of sequential dynamics that is invariant to time reparameterization. In this metric, we first compute truncated path signatures $S(X_{t-p+1:t})$ and $S(X_{t:t+q})$ from real data over historical and future time intervals, respectively. A linear map $\hat{L}$ is learned via regression to approximate the mapping:

$$\hat{L} : S(X_{t-p+1:t}) \mapsto S(X_{t:t+q}). \tag{4.14}$$

Next, the generator is conditioned on the real historical segment $X_{t-p+1:t}$ to produce synthetic future paths $\hat{X}_{t:t+q}$. The signature of the generated sequence, $S(\hat{X}_{t:t+q})$, is then compared to the predicted signature $\hat{L}(S(X_{t-p+1:t}))$ using the ℓ_1 norm. The final score is given by:

$$\text{Signature Metric} = \left\|S(\hat{X}_{t:t+q}) - \hat{L}(S(X_{t-p+1:t}))\right\|_1. \tag{4.15}$$

Interpretation: A lower signature metric indicates that the generator produces future sequences whose path-level structure is consistent with those of the real data, as learned

through the signature mapping. This suggests that the generative model preserves complex sequential dependencies and temporal orderings beyond simple autocorrelation, making it especially suitable for evaluating the fidelity of time series dynamics in financial applications.

4.3.2 Forecasting Performance Under Stress

In the context of stress testing, it is essential to evaluate not only the statistical fidelity of the generated scenarios, but also their capacity to systematically uncover vulnerabilities in financial forecasting models. This dual objective supports the advancement of more robust, uncertainty-aware decision-making frameworks under adverse conditions. To this end, we assess the response of a probabilistic regression model—trained to produce prediction intervals at a predefined confidence level across multiple future time steps—when subjected to synthetic inputs designed to simulate stress scenarios.

Let $\hat{X}_{t+1:t+q} \in \mathbb{R}^{q \times d}$ denote the generated future time series over a prediction horizon of length q, with d dimensions. Let the probabilistic regression model, when given $\hat{X}_{t-p+1:t}$ as input, produce lower and upper prediction bounds $\tilde{X}^{\text{lower}}, \tilde{X}^{\text{upper}} \in \mathbb{R}^{q \times d}$.

We evaluate forecasting performance under stress using three complementary metrics that jointly capture the accuracy, calibration, and informativeness of the predicted intervals:

- **Forecast Error:** Quantifies the deviation between the predicted midpoint of the probabilistic prediction interval and the corresponding generated future sequence. Let each generated future sample be flattened into a vector $\hat{X}_i \in \mathbb{R}^D$, where $D = q \cdot d$, and let $\tilde{X}_i^{\text{mid}} = \frac{1}{2}\left(\tilde{X}_i^{\text{lower}} + \tilde{X}_i^{\text{upper}}\right) \in \mathbb{R}^D$ denote the predicted midpoint. The forecast error is then defined as:

$$\text{Forecast Error} = \frac{1}{N} \sum_{i=1}^{N} \left\| \hat{X}_i - \tilde{X}_i^{\text{mid}} \right\|^2, \tag{4.16}$$

  where N is the number of generated samples. In practice, this corresponds to the MSE between predicted and target vectors, as implemented using `F.mse_loss` in PyTorch.

  Interpretation: A lower forecast error indicates that the probabilistic regression model produces accurate point estimates for future trajectories under synthetic inputs. Elevated errors may suggest that the generated scenarios contain stress features that meaningfully challenge the forecasting model.

- **Average Prediction Interval Width (APIW):** Captures the mean width of the prediction interval at a specified confidence level $1 - \alpha$, across all time steps, variables, and samples:

$$\text{Average Prediction Interval Width}_{(\alpha)} = \frac{1}{N \cdot q \cdot d} \sum_{i=1}^{N} \sum_{t=1}^{q} \sum_{j=1}^{d} \left(\tilde{X}_{i,t,j}^{\text{upper}} - \tilde{X}_{i,t,j}^{\text{lower}} \right). \tag{4.17}$$

Interpretation: Wider intervals indicate greater predictive uncertainty. When interpreted together with coverage, this metric provides insight into the informativeness of the model's uncertainty quantification under stress.

- **Empirical Coverage:** Evaluates the proportion of generated future values that fall within the predicted interval bounds:

$$\text{Coverage}_{(\alpha)} = \frac{1}{N \cdot q \cdot d} \sum_{i=1}^{N} \sum_{t=1}^{q} \sum_{j=1}^{d} \mathbb{I}\left[\tilde{X}_{i,t,j}^{\text{lower}} \leq \hat{X}_{i,t,j} \leq \tilde{X}_{i,t,j}^{\text{upper}}\right], \tag{4.18}$$

where $\mathbb{I}[\cdot]$ is the indicator function.

Interpretation: Coverage quantifies how well-calibrated the prediction intervals are. For a confidence level $1-\alpha$, ideal calibration implies empirical coverage close to $1-\alpha$. Deviations from this target indicate either overconfidence (under-coverage) or excessive conservatism (over-coverage).

Collectively, these metrics offer a rigorous and multidimensional assessment of the forecasting model's robustness when subjected to synthetically generated stress scenarios. High predictive accuracy, coupled with narrow and well-calibrated prediction intervals, indicates that the generative model exposes the forecasting system to plausible yet controlled perturbations. In contrast, elevated forecast errors or excessively wide intervals signal that the generated scenarios effectively challenge the model's assumptions and reveal its vulnerabilities, fulfilling the fundamental objective of stress testing.

4.4 Results and Discussion

This section presents the main empirical findings of the study, evaluating the performance of the proposed `GASTeF` framework across different uncertainty regularization settings. The analysis focuses on two complementary aspects: the statistical quality of the generated synthetic time series and their effectiveness in inducing meaningful stress on the forecasting model. All extended numerical results, training diagnostics, and distributional analyses supporting these findings are reported in Appendix A.

4.4.1 Synthetic Time Series Quality

The experiments conducted across different values of the uncertainty regularization parameter (λ) and prediction confidence levels ($\alpha \in \{0.15, 0.10, 0.05\}$) provide a detailed view of how the `GASTeF` framework balances statistical realism and stress-inducing capability. This subsection summarizes the quantitative results, covering metrics of statistical fidelity, distributional structure, discriminative realism, predictive consistency, and pathwise similarity through the signature Wasserstein distance. The baseline configuration ($\lambda = 0$) for each confidence level serves as the

control condition for assessing the incremental effects of uncertainty-aware regularization. The complete numerical results for all configurations are reported in Appendix A.

At the baseline ($\lambda = 0$), the generator produces synthetic financial returns that closely replicate the empirical properties of real data, with moderate kurtosis error (≈ 1.97), slight positive skewness error (≈ 0.44), and low histogram distance (≈ 0.01)(Figures 4.1–4.2). Increasing marginally the uncertainty regularization term to ($0.01 \leq \lambda \leq 0.03$) slightly alters these statistics while maintaining overall plausibility. For instance, at $\lambda = 0.02$ kurtosis error values range between 2.5 and 3.1 across all confidence levels, and the lag-1 autocorrelation remains close to 0.1, indicating realistic persistence without over-smoothing.

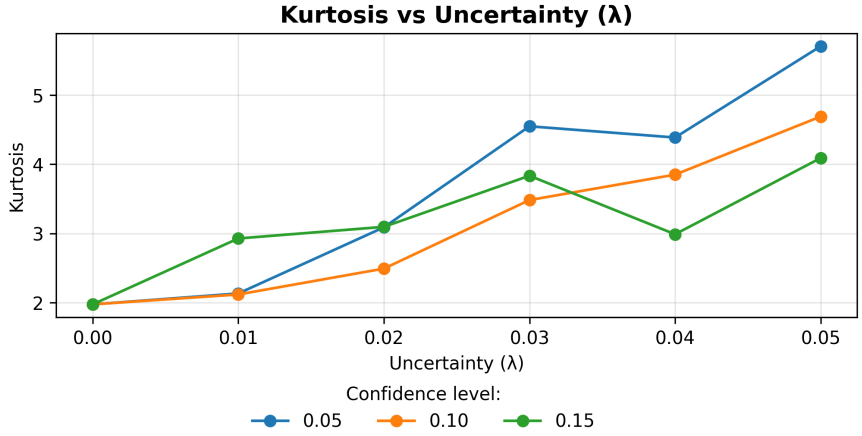

Figure 4.1: Evolution of kurtosis error as a function of the uncertainty regularization parameter (λ) for different prediction confidence levels ($\alpha \in \{0.15, 0.10, 0.05\}$).

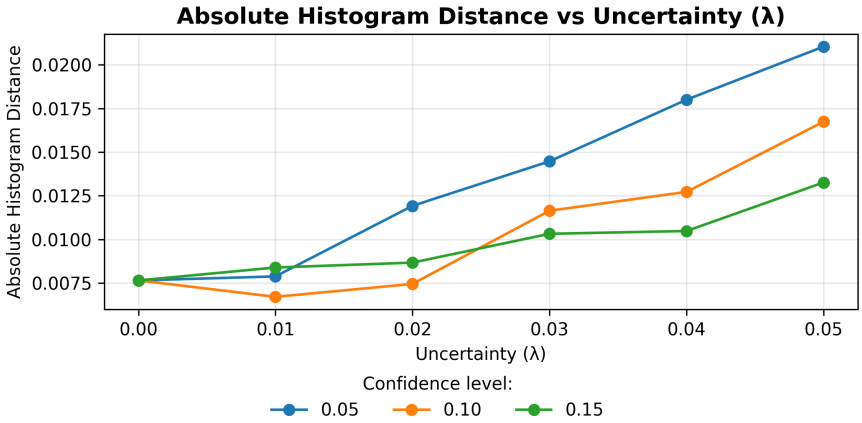

Figure 4.2: Evolution of the absolute histogram distance as a function of the uncertainty regularization parameter (λ) for different prediction confidence levels ($\alpha \in \{0.15, 0.10, 0.05\}$).

Cross-correlation distances (Figure 4.3) and histogram distances also remain almost unchanged, confirming that inter-series relationships are largely preserved. In this scenario, the model achieves a healthy balance between realism and stress generation: the outputs remain statistically coherent while introducing subtle perturbations that challenge downstream forecasting models.

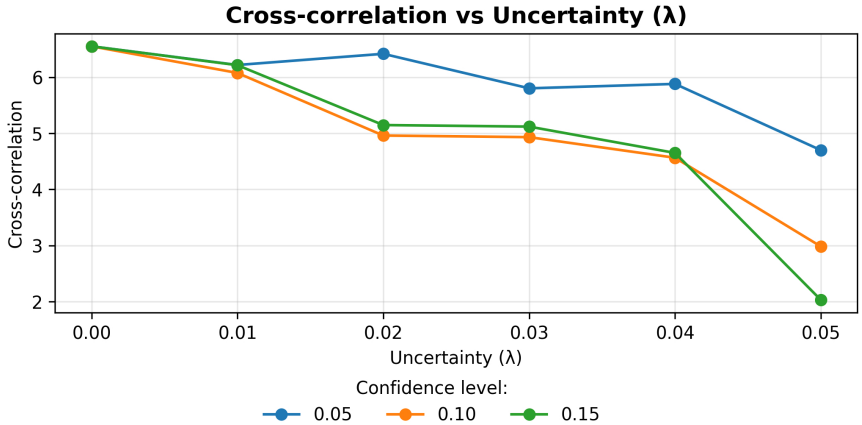

Figure 4.3: Evolution of cross-correlation distance as a function of the uncertainty regularization parameter (λ) for different prediction confidence levels ($\alpha \in \{0.15, 0.10, 0.05\}$).

However, as λ increases beyond 0.05, fidelity begins to deteriorate. kurtosis and skewness errors rise sharply (Figures 4.1 and 4.4), with some configurations (e.g., $\lambda = 0.06, \alpha = 0.05$) reaching values of 10.43 and 3.77, respectively—far exceeding empirical norms. Cross-correlation distances drop (Figure 4.3), implying that asset correlations strengthen under stress, a behavior consistent with market downturns, where assets move more synchronously. These distortions indicate that excessive uncertainty weighting drives the generator into over-regularization, producing extreme and statistically implausible stress scenarios that no longer reflect meaningful financial dynamics.

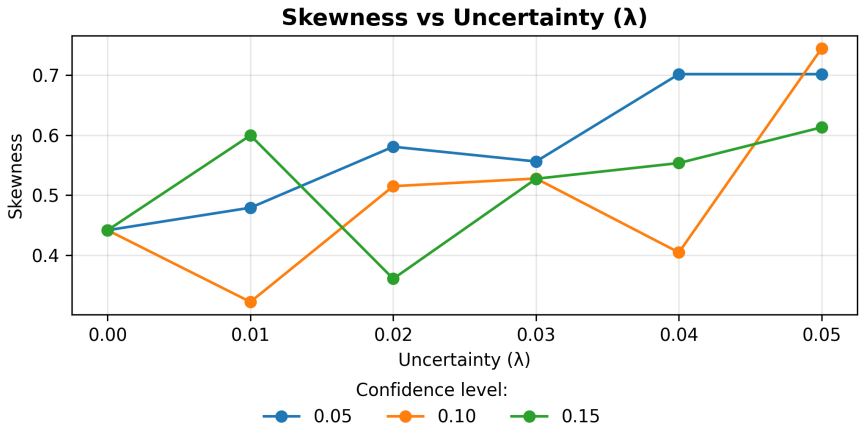

Figure 4.4: Evolution of skewness error as a function of the uncertainty regularization parameter (λ) for different prediction confidence levels ($\alpha \in \{0.15, 0.10, 0.05\}$).

The discriminative score, measuring how easily a classifier can distinguish synthetic data from real samples, provides an intuitive view of realism (Figure 4.5). Across all confidence levels, the baseline ($\lambda = 0$) achieves a moderate score of about 0.31. As uncertainty regularization is introduced, this score improves, reaching a minimum around 0.20–0.23 for $\lambda = 0.02$–0.03, indicating that the generated series have become harder to distinguish from real ones. This suggests that mild uncertainty weighting actually enhances realism by encouraging the generator to explore underrepresented but plausible regions of the data distribution. Beyond this range, however, discriminative performance worsens again, signaling that the synthetic sequences diverge from empirical characteristics.

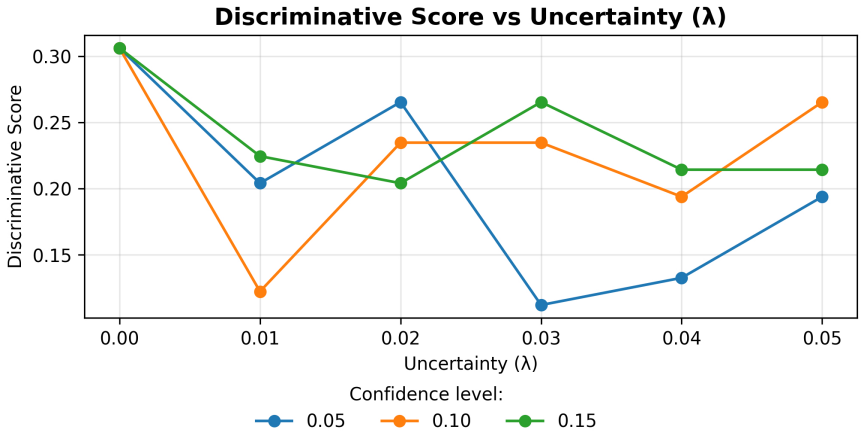

Figure 4.5: Evolution of the discriminative score as a function of the uncertainty regularization parameter (λ) for different prediction confidence levels ($\alpha \in \{0.15, 0.10, 0.05\}$).

Predictive consistency shows a similar pattern (Figure 4.6). For small λ, the gap between R^2_{TSTR} (trained on real, tested on synthetic) and R^2_{TRTR} (trained and tested on real) remains minimal, confirming that the synthetic data preserve the functional relationships learned from the real data. This is particularly important for stress-testing applications, as it demonstrates that the synthetic series remain informative while modestly degrading the predictive model's stability. At high λ, however, predictive coherence diminishes, and the generative process begins to overemphasize uncertainty at the expense of meaningful structure.

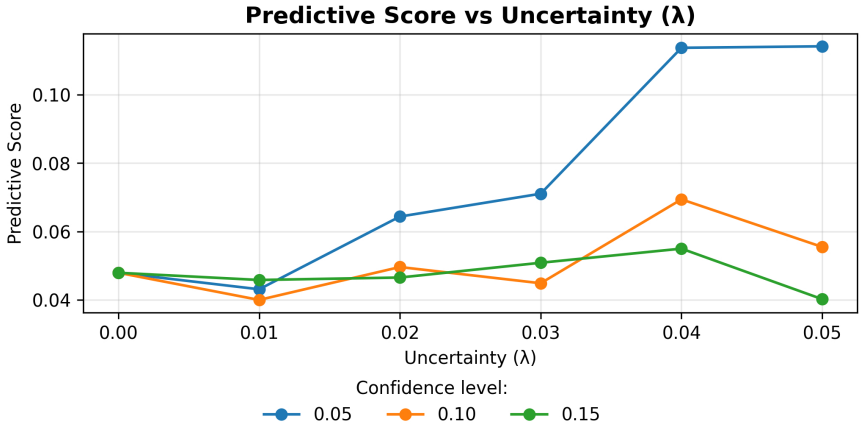

Figure 4.6: Evolution of the predictive score as a function of the uncertainty regularization parameter (λ) for different prediction confidence levels ($\alpha \in \{0.15, 0.10, 0.05\}$).

The signature Wasserstein-1 (Sig-W1) metric (Figure 4.7) captures discrepancies in temporal structure between real and synthetic trajectories. At baseline ($\lambda = 0$), this metric stabilizes around 0.50 across all α levels, confirming that the generator accurately reproduces sequential dependencies. For moderate uncertainty levels ($\lambda \leq 0.04$), Sig-W1 remains stable (0.49–0.53), showing that the generated paths retain realistic evolution patterns. Only at higher λ values does the metric begin to drift upward, reaching 0.68 at $\lambda = 0.06, \alpha = 0.05$, suggesting that structural coherence weakens as uncertainty regularization dominates the training dynamics.

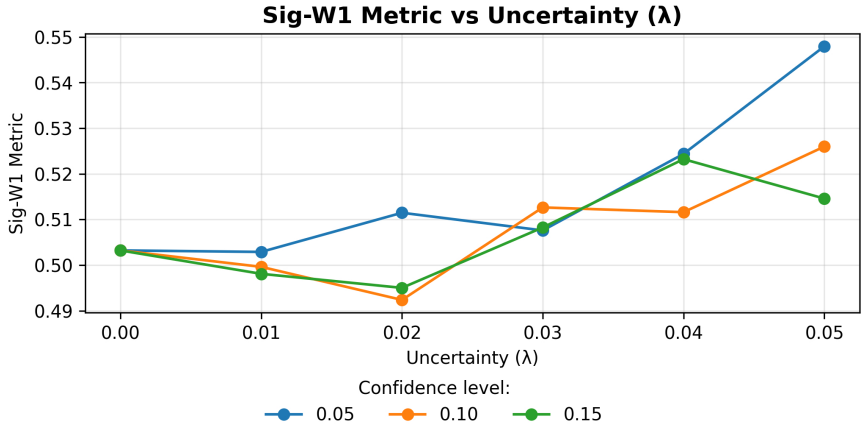

Figure 4.7: Evolution of the signature Wasserstein-1 (Sig-W1) metric as a function of the uncertainty regularization parameter (λ) for different prediction confidence levels ($\alpha \in \{0.15, 0.10, 0.05\}$).

Comparing across confidence levels reveals that the effects of uncertainty weighting are more pronounced at lower α (i.e., tighter confidence intervals). For example, when $\alpha = 0.05$, increases in λ produce larger deviations in kurtosis error, skewness error, and Sig-W1 than at $\alpha = 0.15$, indicating that high-confidence predictive regimes amplify the influence of uncertainty-aware regularization. In practice, this means that `GASTeF` becomes more sensitive when the forecaster's output intervals are narrower, leading to stronger but potentially less realistic stress responses.

Taken together, these results demonstrate that `GASTeF` maintains strong generative fidelity and structural realism for small to moderate uncertainty regularization ($0.01 \leq \lambda \leq 0.04$). In this range, synthetic trajectories are statistically coherent, visually plausible, and capable of inducing measurable stress in predictive models. The baseline ($\lambda = 0$) remains the most realistic configuration but lacks the capacity to expose model vulnerabilities, whereas large uncertainty weights ($\lambda \geq 0.05$) produce distortions that compromise both realism and interpretability. These findings highlight the importance of balancing uncertainty awareness and generative plausibility, an essential consideration when using adversarial generation for financial stress testing.

4.4.2 Stress Testing Effectiveness

The second objective of this study is to assess the effectiveness of the `GASTeF` framework in generating synthetic sequences that expose weaknesses in probabilistic forecasting models. This evaluation focuses on three complementary metrics introduced in Subsection 4.3.2: (i) the forecast error, (ii) the average prediction interval width (APIW), and (iii) the empirical coverage. Together,

these measures provide a comprehensive characterization of model degradation, uncertainty expansion, and calibration stability under progressively stressed input conditions.

Forecast error. The forecast error, defined as the mean squared deviation between the generated target and the midpoint of the model's predictive interval, exhibits a systematic dependence on the uncertainty regularization parameter λ across all confidence levels ($\alpha \in \{0.15, 0.10, 0.05\}$). Figure 4.8 illustrates this relationship, showing a monotonic increase in forecast error as λ grows, consistent across confidence levels.

At the baseline configuration ($\lambda = 0$), the forecast error remains low—ranging between 0.09 and 0.16—indicating that the forecasting model performs reliably when exposed to data closely aligned with the empirical distribution. As λ increases, the forecast error rises progressively, reaching 0.12–0.22 for $\alpha = 0.15$, 0.14–0.21 for $\alpha = 0.10$, and up to 0.31 for $\alpha = 0.05$ at $\lambda = 0.05$. This monotonic increase reflects the growing difficulty of the forecasting task as the generator introduces controlled deviations from the real data distribution.

At higher regularization levels ($\lambda = 0.06$), the forecast error escalates sharply—up to 1.35 for $\alpha = 0.05$—indicating that the excessive regularization results in substantial deviation from historical behavior, thereby inducing pronounced forecasting failures.

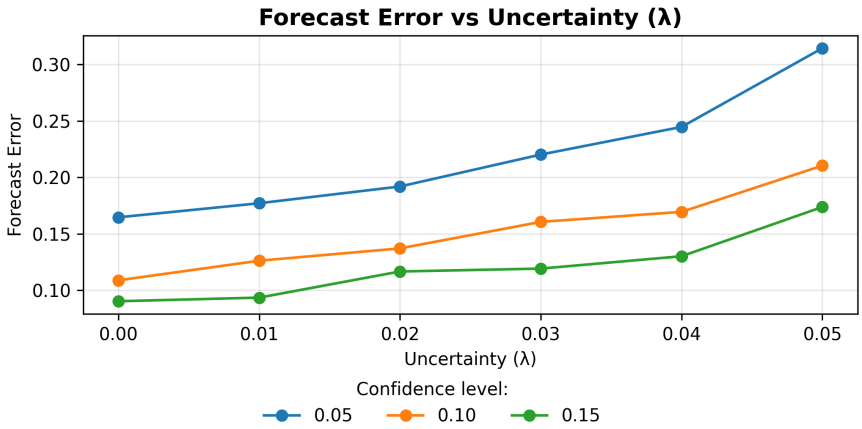

Figure 4.8: Evolution of forecast error as a function of the uncertainty regularization parameter (λ) for different prediction confidence levels ($\alpha \in \{0.15, 0.10, 0.05\}$).

Average prediction interval width. The APIW quantifies the degree of epistemic uncertainty induced in the forecasting model. As shown in Figure 4.9, a consistent widening of prediction intervals is observed as λ increases, reflecting greater uncertainty when the model encounters more challenging or atypical synthetic inputs.

At the baseline configuration ($\lambda = 0$), APIW values range from 1.10 ($\alpha = 0.15$) to 2.12 ($\alpha = 0.05$), reflecting a well-calibrated probabilistic forecaster operating under conditions consistent with the historical data distribution. Under moderate uncertainty regularization ($0.02 \leq \lambda \leq 0.05$), interval widths increase gradually—reaching 1.23 for $\alpha = 0.15$, 1.66 for $\alpha = 0.10$, and 2.81 for $\alpha = 0.05$—while remaining within plausible limits. This controlled expansion demonstrates that the `GASTeF` generator successfully induces meaningful epistemic stress without compromising statistical coherence, thereby enabling robust evaluation of the forecasting model's uncertainty calibration.

Beyond this regime ($\lambda > 0.05$), however, APIW values increase sharply, reaching 6.79 for $\alpha = 0.05$. Such pronounced widening suggests that excessive uncertainty regularization causes the generator to produce implausibly volatile stress scenarios, leading to over-dispersed and less interpretable predictive intervals.

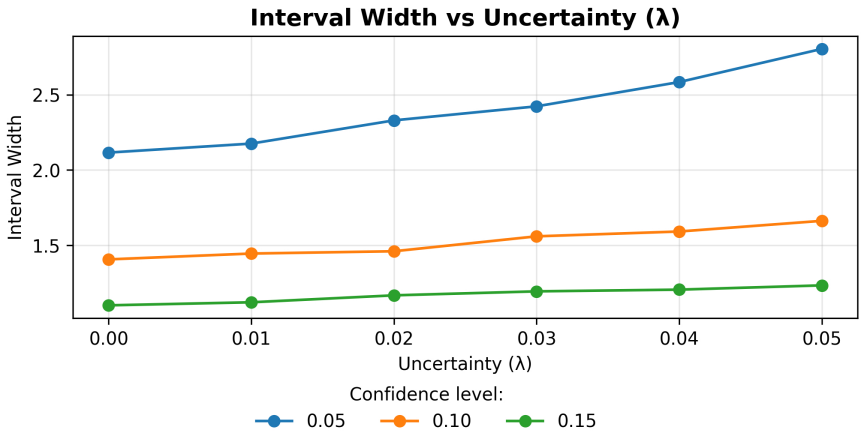

Figure 4.9: Evolution of the average prediction interval width (APIW) as a function of the uncertainty regularization parameter (λ) for different confidence levels ($\alpha \in \{0.15, 0.10, 0.05\}$).

Empirical coverage. Coverage measures the proportion of true observations falling within the predicted intervals, assessing calibration reliability. As shown in Figure 4.10, for $\lambda \leq 0.05$, coverage remains close to the nominal levels ($1 - \alpha$), ranging from 0.85–0.92 for $\alpha = 0.15$, 0.88–0.94 for $\alpha = 0.10$, and 0.96–0.97 for $\alpha = 0.05$. These results indicate that the forecaster maintains proper calibration under moderate perturbations. However, when λ increases to 0.06, deviations emerge: coverage declines to 0.82 for $\alpha = 0.15$ and 0.88 for $\alpha = 0.10$, while slightly exceeding the target (0.98) for $\alpha = 0.05$. This asymmetric behavior highlights instability in uncertainty estimation under extreme stress.

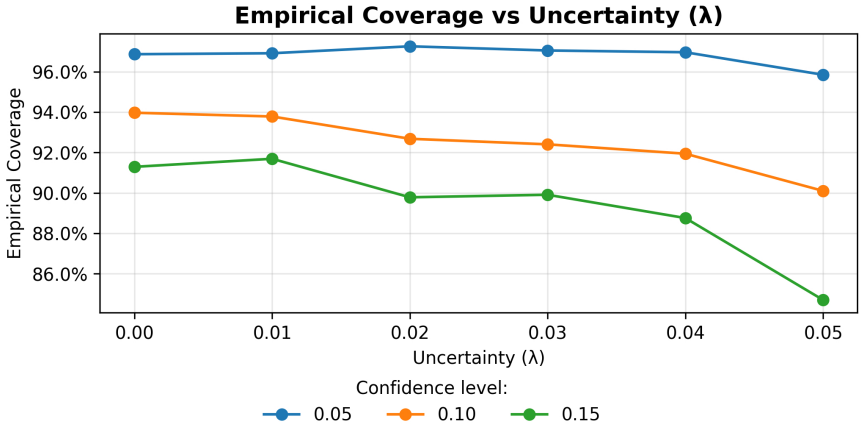

Figure 4.10: Empirical coverage (C_{emp}) as a function of the uncertainty regularization parameter (λ) for different confidence levels ($\alpha \in \{0.15, 0.10, 0.05\}$).

The joint evolution of forecast error, APIW, and empirical coverage reveals a clear trade-off between statistical fidelity and stress intensity. For low and moderate values of λ ($0.01 \leq \lambda \leq 0.05$), `GASTeF` generates synthetic data that remain statistically coherent yet sufficiently stressful to induce controlled performance degradation and uncertainty expansion. Compared to the baseline, these configurations effectively highlight vulnerabilities in the forecasting model without violating the underlying data realism. When λ exceeds this range, the generator begins to produce highly distorted trajectories, leading to sharp increases in forecast error and excessive uncertainty, thereby compromising interpretability and realism. Excessive regularization underscores the importance of selecting a suitable regularization range that maintains the trade-off between realism and stressfulness.

Overall, the results confirm that the proposed uncertainty-aware adversarial training mechanism enables `GASTeF` to produce plausible yet stress-inducing synthetic scenarios. Moderate levels of uncertainty regularization yield the optimal balance between realism and adversarial strength, validating the framework's capacity to support systematic stress testing of probabilistic forecasting models.

4.4.3 Results Discussion

Subsections 4.4.1 and 4.4.2 presented a comprehensive empirical assessment of the proposed `GASTeF` framework, evaluating its ability to generate statistically coherent yet adversarial time series for stress testing probabilistic forecasting models. The evaluation was structured along two complementary dimensions: (*i*) the statistical quality and realism of the synthetic data, and (*ii*) the

effectiveness of these data in inducing controlled degradation of the forecasting model predictive performance.

From a statistical quality perspective, the results demonstrated that for low to moderate levels of uncertainty regularization ($\lambda \in [0.01, 0.04]$), the generated sequences maintain high fidelity with respect to empirical financial time series. Distributional characteristics—such as skewness error, kurtosis error, and histogram distance—remained within realistic bounds, while lag-1 autocorrelation and cross-correlation distance structures were preserved. Path-wise coherence, quantified through the signature Wasserstein-1 distance, remained stable across confidence levels, confirming that temporal dynamics and inter-series dependencies were faithfully reproduced. The discriminative and predictive utility metrics further support these findings, showing minimal divergence between real and synthetic samples and a preserved functional structure across time series. At $\lambda = 0.02$, the discriminator's accuracy reached its minimum, indicating that synthetic and real samples were statistically indistinguishable and that the generator achieved optimal realism.

Beyond this optimal regime, however, statistical degradation became evident. Increasing λ beyond 0.05 led to heavy-tailed and asymmetric distributions, unrealistically strong inter-series co-movements, and weakened sequential consistency. These distortions were reflected in sharp increases in kurtosis, skewness, and cross-correlation, as well as in unstable discriminative and predictive scores. The observed deterioration highlights a fundamental trade-off between the realism of the generated data and the strength of the induced stress: while higher regularization enhances adversarial pressure, it also risks departing from the manifold of plausible market behavior.

The analysis of stress-testing effectiveness reinforced these observations. As λ increased, both forecast error and average prediction interval width (APIW) rose monotonically, indicating that the generated data progressively challenged the forecasting model's generalization ability. Importantly, for moderate regularization ($\lambda \leq 0.04$), this degradation occurred without compromising empirical coverage, suggesting that the forecaster's uncertainty estimates remained well-calibrated under realistic stress. In this range, `GASTeF` successfully induced epistemic stress—forcing the forecasting model to widen its predictive intervals and adapt to more uncertain environments—without producing implausible inputs. At higher regularization levels ($\lambda \geq 0.05$), however, the model exhibited signs of breakdown, with explosive forecast errors, inflated interval widths, and declining coverage. These results indicate that excessive uncertainty penalization generates stress scenarios that are statistically unstable and less informative from a risk management standpoint.

Taken together, the findings confirm that `GASTeF` achieves its dual objective: maintaining statistical coherence while generating synthetic scenarios that meaningfully stress probabilistic forecasting models. Within a moderate regularization regime, the framework produces plausible synthetic trajectories that degrade predictive accuracy and expand uncertainty in a controlled and interpretable manner. Beyond a critical threshold, however, the adversarial dynamics become excessively dominant, leading to unrealistic distortions that compromise interpretability and practical applicability.

Limitations and Threats to Validity. Despite these encouraging results, several limitations should be acknowledged.

First, the evaluation relies on a finite historical dataset that may not capture all possible market regimes, structural breaks, or tail dependencies. Consequently, the observed generalization and stress-testing performance might vary under alternative datasets or macro-financial conditions.

Second, the realism of the generated stress scenarios is inherently constrained by the representational capacity of the underlying generator. If the generative model fails to capture rare but plausible market dynamics, it may underrepresent systemic risks or extreme co-movements. In particular, as illustrated in Figure 4.11, the baseline configuration ($\lambda = 0$, $\alpha = 0.05$) reveals a notable limitation in the generator's ability to reproduce realistic cross-asset dependencies. The synthetic correlation matrix exhibits overly homogeneous relationships compared to the more heterogeneous cross-asset dependencies observed in the real data, indicating that the current formulation of `GASTeF` does not fully capture joint dynamics across assets. This limitation constrains the realism of system-wide stress propagation and motivates future extensions toward explicitly multivariate or dependency-aware architectures.

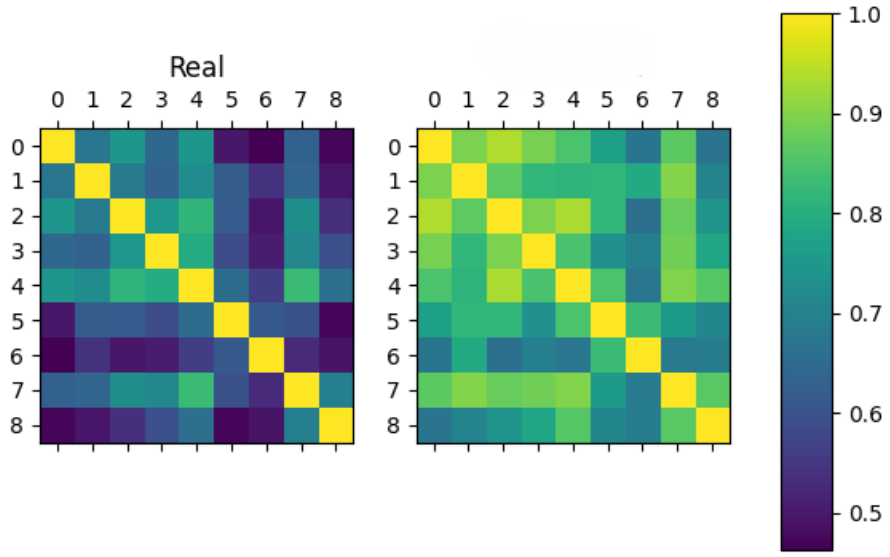

Figure 4.11: Cross-asset correlation matrices for real data (left) and synthetic data generated by `GASTeF` ($\lambda = 0$, $\alpha = 0.05$). The figure highlights the generator's limitation in reproducing realistic inter-asset dependencies, motivating future extensions toward explicitly multivariate modeling.

Third, while the selected metrics—such as forecast error, interval width, and empirical coverage—offer a rigorous view of predictive degradation and uncertainty calibration, they do not exhaustively represent the behavior of forecasting models under stress. Complementary analyses,

such as tail-risk sensitivity, dependence structure breakdown, or cross-regime robustness, could provide additional insight into model resilience.

Another limitation concerns the interpretability of high-λ stress scenarios. Although such configurations can expose latent weaknesses in forecasting systems, the resulting synthetic trajectories may deviate substantially from realistic market dynamics, limiting their use for policy evaluation or risk management. This tension between adversarial strength and statistical plausibility represents a fundamental trade-off in uncertainty-aware generative modeling and highlights the need for adaptive regularization or realism constraints to maintain domain relevance.

Finally, potential threats to validity arise from the experimental design and modeling choices. The forecaster used for evaluation represents a specific architecture and loss function; different model families—such as recurrent networks, Bayesian ensembles, or regime-switching forecasters—could exhibit distinct sensitivity patterns. Similarly, hyperparameter tuning and training stability may influence both generative quality and stress magnitude. Future work should therefore focus on robustness analyses across architectures, datasets, and market conditions to confirm the generalizability of the present findings.

Summary. In summary, `GASTeF` demonstrates a strong capacity to produce statistically consistent and functionally meaningful stress scenarios. The framework effectively balances realism and adversarial intensity within a well-defined regularization range, providing a principled foundation for controlled stress testing of forecasting systems. Nevertheless, careful calibration of the uncertainty regularization strength and systematic validation across models and datasets remain essential to ensuring both robustness and interpretability in real-world financial applications.

Chapter 5

Conclusion

This work addressed the need for systematic stress testing methodologies in financial forecasting by introducing GASTeF—a generative adversarial framework designed to synthesize statistically plausible yet adversarial financial time series to stress test forecasting models. Motivated by the need for robust and responsible AI in finance, where models must operate reliably under uncertainty, GASTeF leverages generative adversarial learning to synthesize stressful financial time series that expose vulnerabilities in probabilistic forecasting models that elicit degraded confidence and/or performance.

Built upon a GAN architecture, the proposed framework integrates predictive uncertainty directly into the generator's training objective, enabling targeted synthesis of stress-inducing inputs. The design is guided by three core desiderata: statistical plausibility, uncertainty-driven generation, and flexible stress criteria.

The empirical results confirmed that GASTeF achieves its dual objective. For moderate values of the uncertainty regularization parameter $\lambda \in [0.01, 0.04]$, the generated time series retained high statistical fidelity, matching real data in terms of distributional characteristics, autocorrelation structures, cross-asset relationships, and path-wise consistency. At the same time, these sequences effectively degraded forecasting performance by increasing predictive error and uncertainty, while maintaining near-nominal coverage. Beyond a critical threshold ($\lambda > 0.05$), the adversarial strength of the generator compromised data realism, resulting in heavy-tailed distributions, implausible dependencies, and destabilized prediction intervals.

These findings position GASTeF as a promising and practical tool for stress testing forecasting systems. The framework offers a tunable, data-driven mechanism for examining model robustness across the entire uncertainty spectrum, ranging from benign perturbations to extreme stress events. In doing so, it contributes to the development of more transparent, resilient, and trustworthy forecasting systems.

5.1 Key Contributions

This work makes the following scientific and methodological contributions:

- **Introduction of the `GASTeF` framework with uncertainty-aware adversarial generation:** A novel generative adversarial architecture for stress testing forecasting models using synthetic financial time series that induce epistemic stress. The framework integrates model uncertainty directly into the generator's loss function, enabling targeted synthesis of stress scenarios within a regression setting and allowing the generator to learn adversarial perturbations that expose weaknesses in probabilistic forecasting models.

- **Comprehensive empirical evaluation:** Development and implementation of a rigorous multi-criteria evaluation protocol assessing fidelity, predictive generalization, discriminative similarity, and structural coherence. The results demonstrate a quantifiable trade-off between statistical plausibility and stress-inducing effectiveness—modulated by the uncertainty regularization parameter λ—and empirically validate the framework on S&P 500 sector ETF data, confirming its effectiveness and practical relevance in real-world financial modeling contexts. Beyond this empirical setting, the proposed architecture can be readily applied to other asset classes, market environments, and forecasting models, underscoring its versatility as a general stress-testing framework.

5.2 Future Work

While the current study demonstrates the feasibility and utility of `GASTeF`, several extensions could significantly enhance its scope, interpretability, and practical relevance:

- **Generalization to multivariate forecasting:** Future work should extend `GASTeF` to better capture cross-asset dependencies and joint stress dynamics across multiple financial series. This would allow the generation of system-wide stress scenarios that more accurately reflect interconnected market behavior.

- **Application to broader financial domains:** Testing the framework on diverse asset classes, such as fixed income, commodities, and digital assets, could evaluate its robustness and transferability beyond equity-based forecasting tasks.

- **Integration with portfolio optimization:** A promising direction for future research involves extending the generator objective to jointly optimize stress realism and portfolio robustness. Specifically, beyond the current combination of the signature-based Wasserstein distance and uncertainty regularization, the generator loss could incorporate an additional term reflecting portfolio performance under generated stress scenarios. This would align data generation with downstream decision-making, enabling `GASTeF` to produce trajectories that not only challenge forecasting models but also test the resilience of portfolio allocations. Figure 5.1 illustrates this extended architecture, in which the forecaster is coupled with a portfolio optimizer that feeds back an optimizer loss signal to guide the generator toward adversarial yet financially meaningful scenarios.

- **Automated hyperparameter and regularization tuning:** Developing adaptive or meta-learning strategies for dynamically adjusting uncertainty and stress parameters (e.g., λ) could improve scalability and stability. This includes Bayesian optimization or reinforcement-based tuning of regularization strength during training.
- **Establishment of standardized evaluation benchmarks and validation protocols:** Defining consistent metrics, reference datasets, and experimental protocols would enable systematic comparison among stress-testing frameworks and promote reproducibility across studies.
- **Scenario validation with historical crises:** Comparing synthetic stress trajectories with known market dislocations (e.g., the 2008 Global Financial Crisis or COVID-19 pandemic) could provide interpretable validation and enhance practitioner confidence in model outputs.
- **Deployment and open-source implementation:** Developing and releasing a GitHub package containing the full `GASTeF` pipeline, pre-trained components, and evaluation tools would facilitate reproducibility, transparency, and integration into financial stress-testing and model validation workflows.

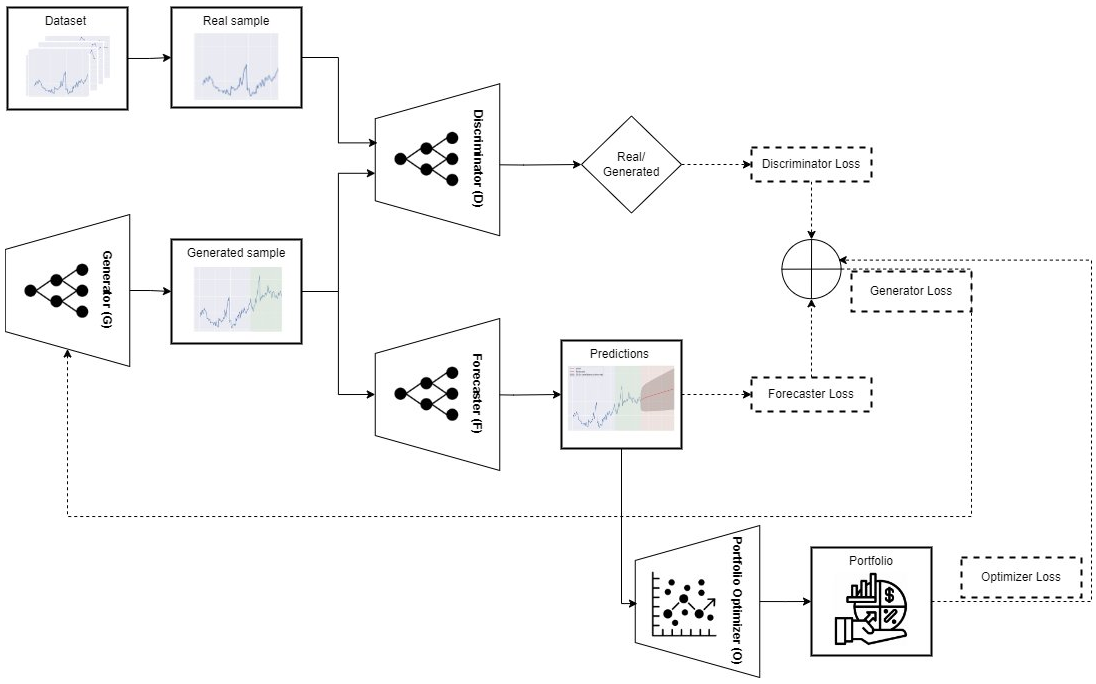

Figure 5.1: Extended GASTeF framework integrating portfolio optimization into the stress-generation process. The generator (G) produces synthetic stress trajectories evaluated by the discriminator (D) and forecaster (F), as in the base model. In the extended setting, a portfolio optimizer (O) receives the forecaster's predictions and updates allocations under the generated scenarios. The resulting optimizer loss is propagated back to the generator, enriching its objective with a portfolio-aware term that encourages the synthesis of stress conditions revealing vulnerabilities in both forecasting and allocation strategies.

Overall, GASTeF represents a step toward proactive and explainable financial stress testing. By generating realistic, multi-dimensional adversarial scenarios, it offers a foundation for developing forecasting and portfolio construction approaches that remain resilient under uncertainty and extreme market conditions.

Appendix A

Extended Experimental Results and Diagnostics

This appendix provides complementary material supporting the empirical findings presented in Chapter 4. It includes: (i) detailed numerical results for all experimental configurations, (ii) training diagnostics illustrating optimization stability and loss component behavior, and (iii) distributional and temporal comparisons between real and generated data. Together, these materials ensure transparency and reproducibility of the `GASTeF` experiments.

A.1 Numerical Results Tables

This section provides the complete numerical results for all combinations of uncertainty weighting λ and confidence level α, supporting the analyses discussed in Chapter 4.

Table A.1: Statistical fidelity metrics as a function of the uncertainty-weight λ and confidence level α.

λ	α	Abs. Hist. Dist.	ACF (lag-1)	Kurtosis	Skewness	Cross-Corr.
0.00	0.15	0.01	0.07	1.97	0.44	6.55
	0.10	0.01	0.07	1.97	0.44	6.55
	0.05	0.01	0.07	1.97	0.44	6.55
0.01	0.15	0.01	0.11	2.93	0.60	6.22
	0.10	0.01	0.09	2.12	0.32	6.08
	0.05	0.01	0.10	2.13	0.48	6.22
0.02	0.15	0.01	0.06	3.09	0.36	5.15
	0.10	0.01	0.11	2.49	0.52	4.96
	0.05	0.01	0.09	3.09	0.58	6.42
0.03	0.15	0.01	0.08	3.83	0.53	5.12
	0.10	0.01	0.11	3.48	0.53	4.93
	0.05	0.01	0.12	4.55	0.56	5.80
0.04	0.15	0.01	0.11	2.99	0.55	4.65
	0.10	0.01	0.07	3.85	0.40	4.56
	0.05	0.02	0.08	4.39	0.70	5.88
0.05	0.15	0.01	0.06	4.09	0.61	2.03
	0.10	0.02	0.05	4.69	0.74	2.98
	0.05	0.02	0.08	5.70	0.70	4.70
0.06	0.15	0.02	0.05	3.45	0.56	2.13
	0.10	0.02	0.05	3.13	0.64	2.06
	0.05	0.02	0.04	10.43	3.77	12.04

Table A.2: Discriminative, Predictive, and Signature-based Metrics as a function of the uncertainty-weight λ and confidence level α.

λ	α	Discriminative Score	R^2_{TSTR}	R^2_{TRTR}	Predictive Score	Sig-W1 Metric
0.00	0.15	0.31	0.02	0.05	0.01	0.50
	0.10	0.31	0.02	0.05	0.01	0.50
	0.05	0.31	0.02	0.05	0.01	0.50
0.01	0.15	0.22	0.02	0.05	0.01	0.50
	0.10	0.12	0.02	0.04	0.01	0.50
	0.05	0.20	0.02	0.04	0.01	0.50
0.02	0.15	0.20	0.02	0.05	0.01	0.50
	0.10	0.23	0.02	0.05	0.01	0.49
	0.05	0.27	0.02	0.06	0.01	0.51
0.03	0.15	0.27	0.02	0.05	0.01	0.51
	0.10	0.23	0.02	0.04	0.01	0.51
	0.05	0.11	0.02	0.07	0.01	0.51
0.04	0.15	0.21	0.02	0.06	0.01	0.52
	0.10	0.19	0.02	0.07	0.01	0.51
	0.05	0.13	0.02	0.11	0.02	0.52
0.05	0.15	0.21	0.02	0.04	0.01	0.51
	0.10	0.27	0.02	0.06	0.01	0.53
	0.05	0.19	0.02	0.11	0.02	0.55
0.06	0.15	0.01	0.02	0.05	0.02	0.53
	0.10	0.28	0.02	0.07	0.02	0.55
	0.05	0.47	0.02	16.54	0.02	0.68

Table A.3: Surrogate model performance as a function of the uncertainty-weight λ and confidence level α.

λ	α	Forecast Error	Interval Width	Coverage
0.00	0.15	0.09	1.10	0.91
	0.10	0.11	1.41	0.94
	0.05	0.16	2.12	0.97
0.01	0.15	0.09	1.12	0.92
	0.10	0.13	1.44	0.94
	0.05	0.18	2.18	0.97
0.02	0.15	0.12	1.17	0.90
	0.10	0.14	1.46	0.93
	0.05	0.19	2.33	0.97
0.03	0.15	0.12	1.19	0.90
	0.10	0.16	1.56	0.92
	0.05	0.22	2.42	0.97
0.04	0.15	0.13	1.20	0.89
	0.10	0.17	1.59	0.92
	0.05	0.24	2.59	0.97
0.05	0.15	0.17	1.23	0.85
	0.10	0.21	1.66	0.90
	0.05	0.31	2.81	0.96
0.06	0.15	0.22	1.30	0.82
	0.10	0.26	1.66	0.88
	0.05	1.35	6.79	0.98

A.2 Training Diagnostics

This section complements the numerical tables in section A.1 with training diagnostics and per-run visualizations. Each figure reports (top to bottom) the evolution of the following metrics: (i) loss terms (total, signature, and uncertainty terms), (ii) forecaster model average prediction interval width (APIW), (iii) absolute histogram distance, (iv) lag-1 autocorrelation, and (v) cross-correlation across training iterations. For illustration purposes, these visual diagnostics are presented for $\alpha = 0.05$ across different values of the uncertainty regularization parameter λ, as this configuration provides representative insight into the model's training behavior and stability patterns observed across confidence levels.

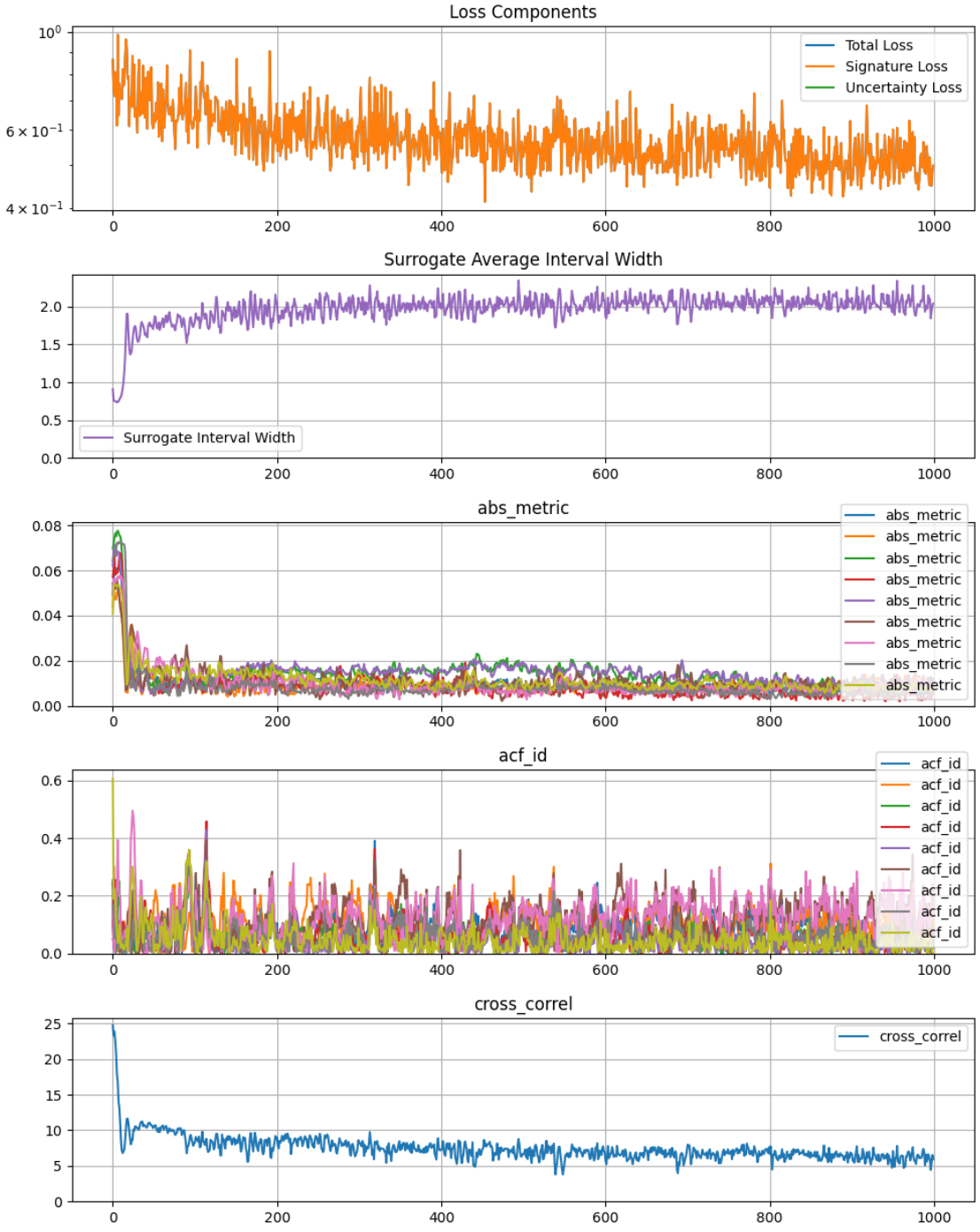

Figure A.1: Training diagnostics for $\lambda = 0$, $\alpha = 0.05$. Panels show (top to bottom): loss components (total, signature, uncertainty), forecaster APIW, absolute histogram distance (`abs_metric`), lag-1 autocorrelation (`acf_id`), and cross-correlation (`cross_correl`) over training iterations.

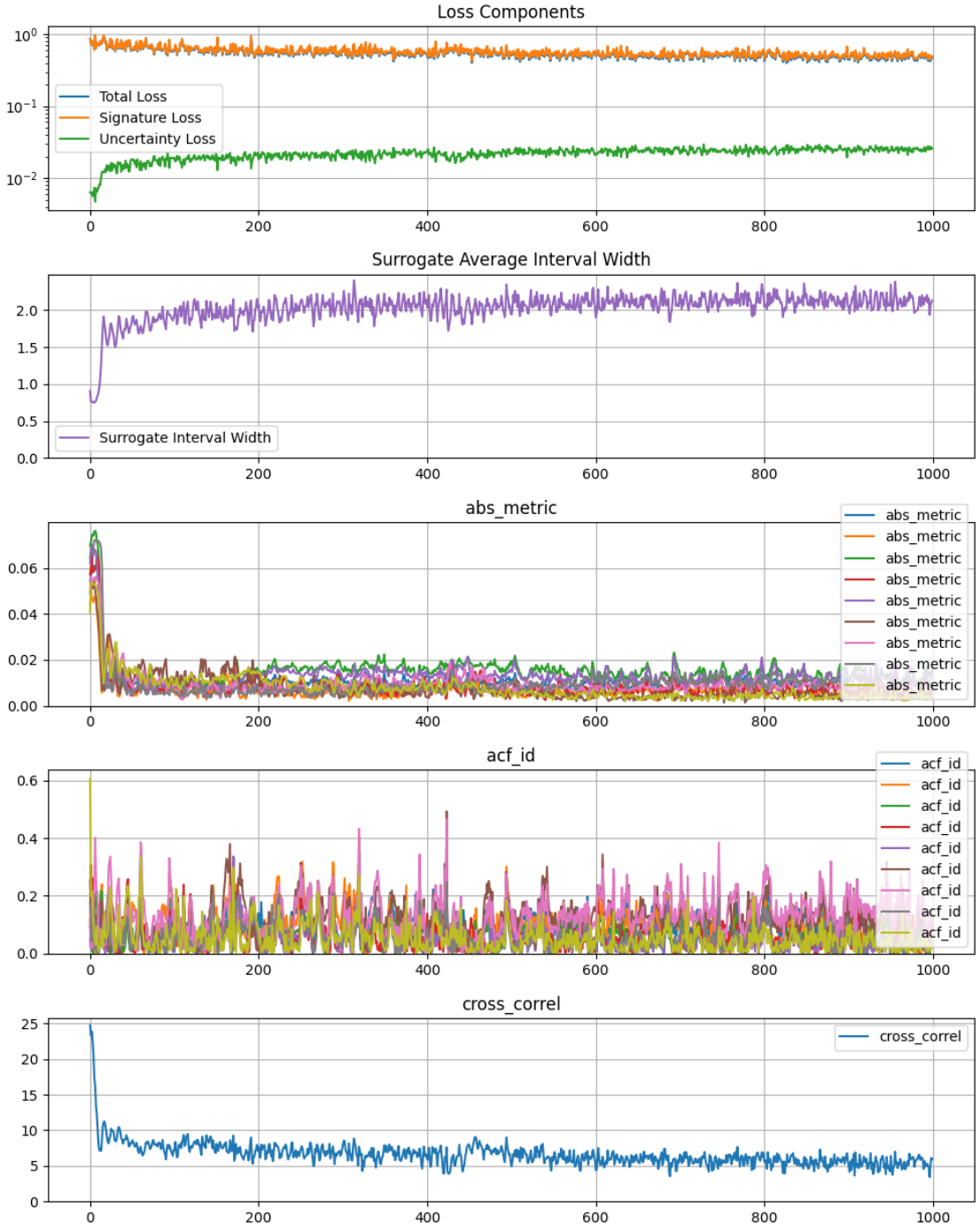

Figure A.2: Training diagnostics for $\lambda = 0.01$, $\alpha = 0.05$. Panels show (top to bottom): loss components (total, signature, uncertainty), forecaster APIW, absolute histogram distance (`abs_metric`), lag-1 autocorrelation (`acf_id`), and cross-correlation (`cross_correl`) over training iterations.

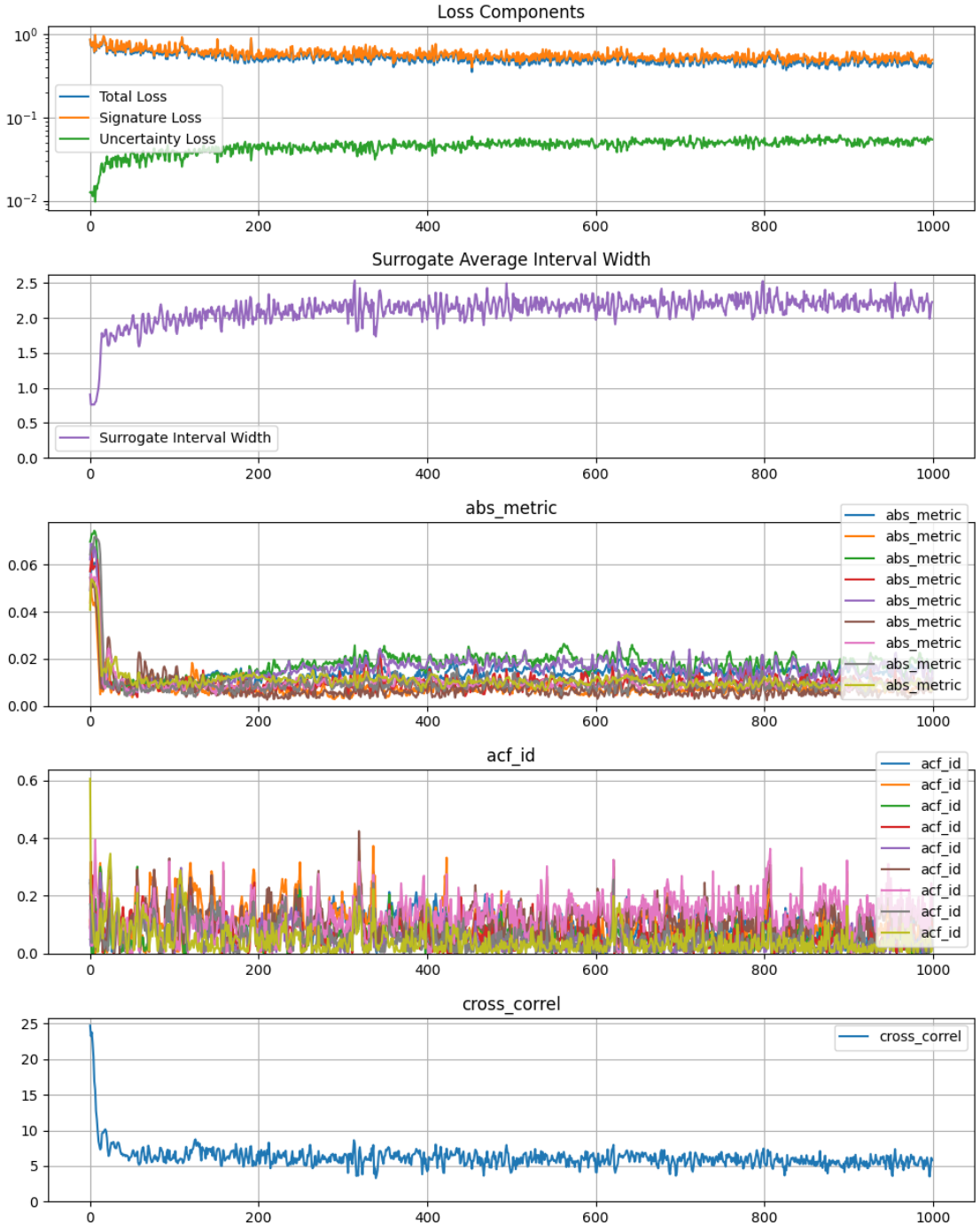

Figure A.3: Training diagnostics for $\lambda = 0.02$, $\alpha = 0.05$. Panels show (top to bottom): loss components (total, signature, uncertainty), forecaster APIW, absolute histogram distance (`abs_metric`), lag-1 autocorrelation (`acf_id`), and cross-correlation (`cross_correl`) over training iterations.

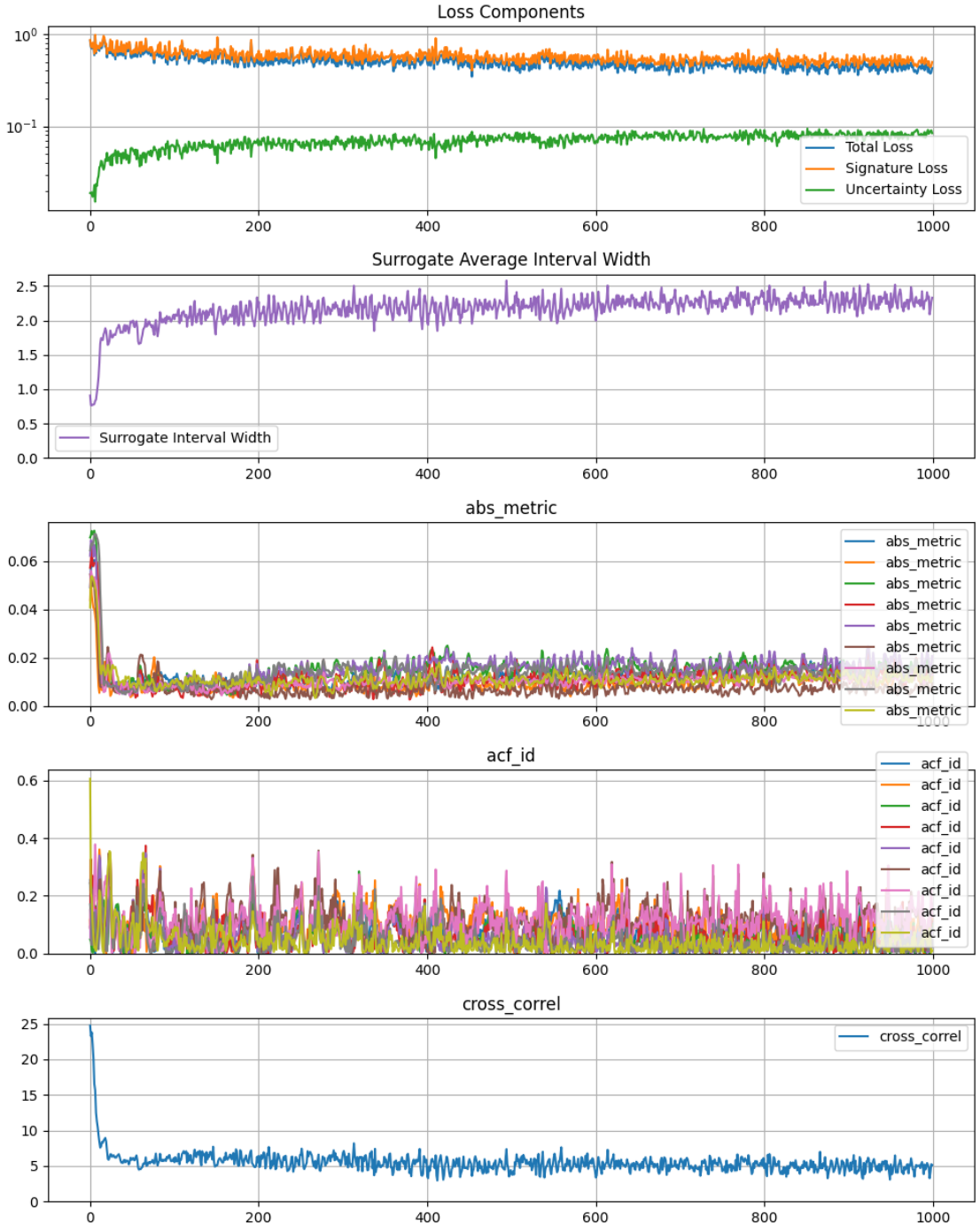

Figure A.4: Training diagnostics for $\lambda = 0.03$, $\alpha = 0.05$. Panels show (top to bottom): loss components (total, signature, uncertainty), forecaster APIW, absolute histogram distance (`abs_metric`), lag-1 autocorrelation (`acf_id`), and cross-correlation (`cross_correl`) over training iterations.

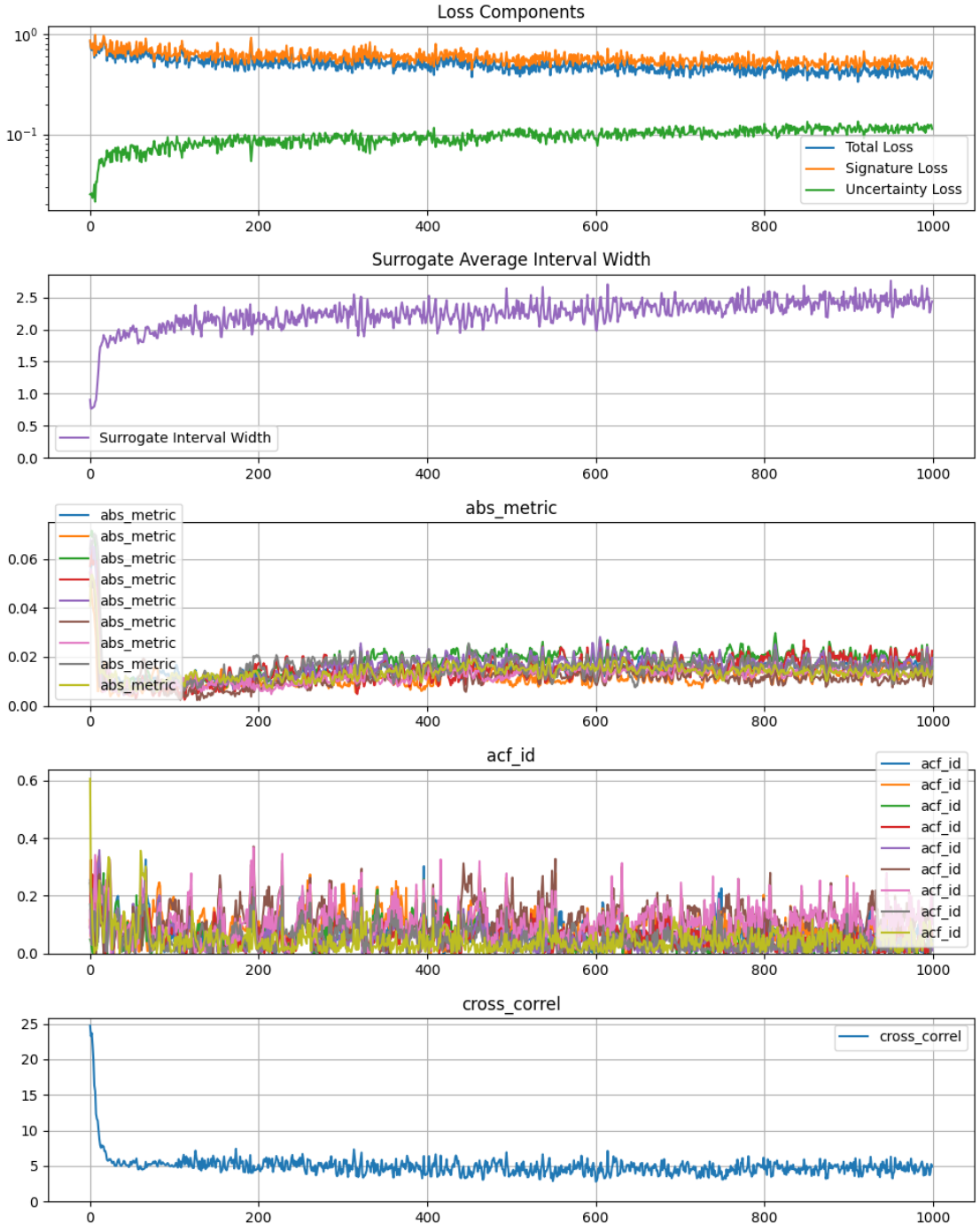

Figure A.5: Training diagnostics for $\lambda = 0.04$, $\alpha = 0.05$. Panels show (top to bottom): loss components (total, signature, uncertainty), forecaster APIW, absolute histogram distance (`abs_metric`), lag-1 autocorrelation (`acf_id`), and cross-correlation (`cross_correl`) over training iterations.

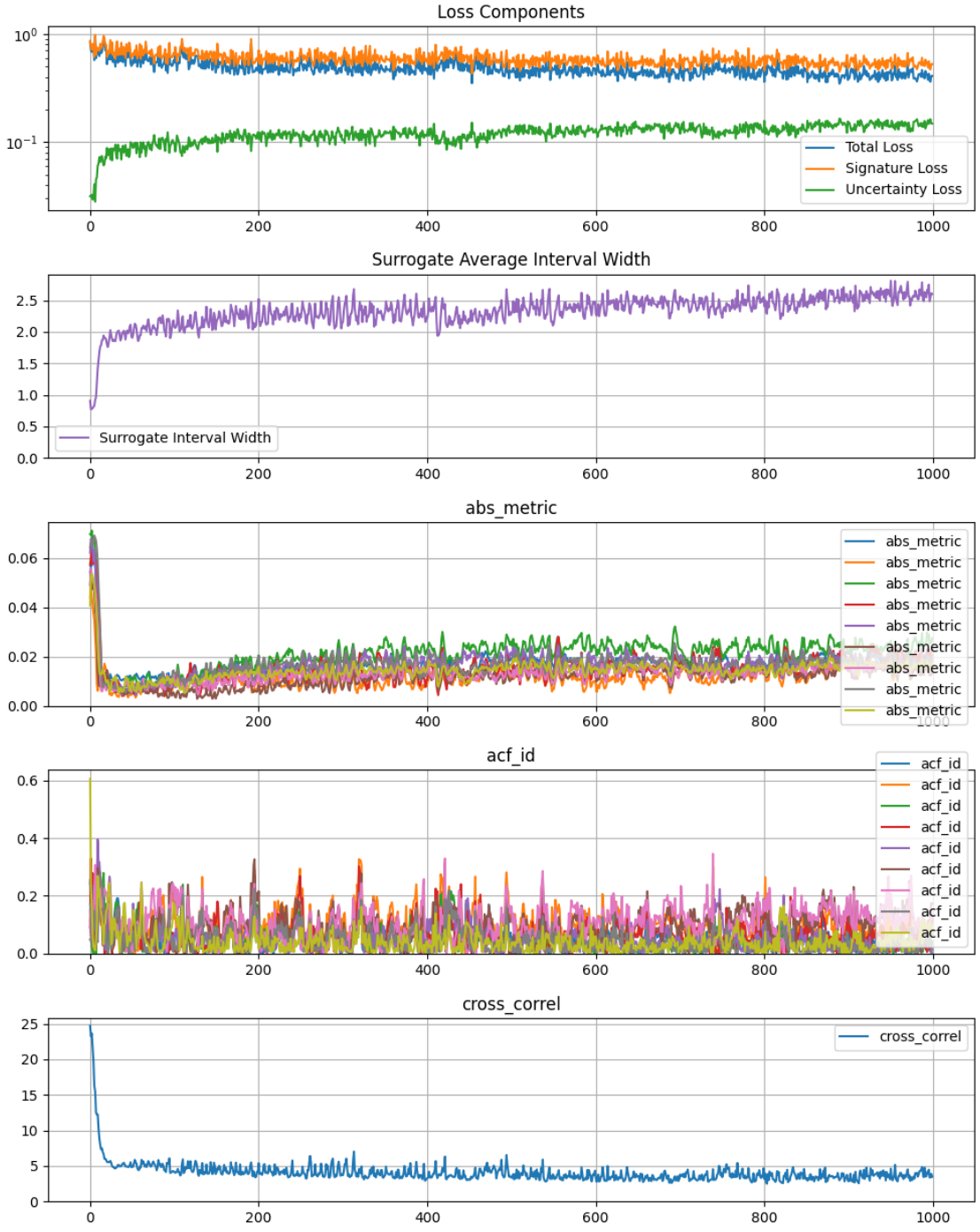

Figure A.6: Training diagnostics for $\lambda = 0.05$, $\alpha = 0.05$. Panels show (top to bottom): loss components (total, signature, uncertainty), forecaster APIW, absolute histogram distance (`abs_metric`), lag-1 autocorrelation (`acf_id`), and cross-correlation (`cross_correl`) over training iterations.

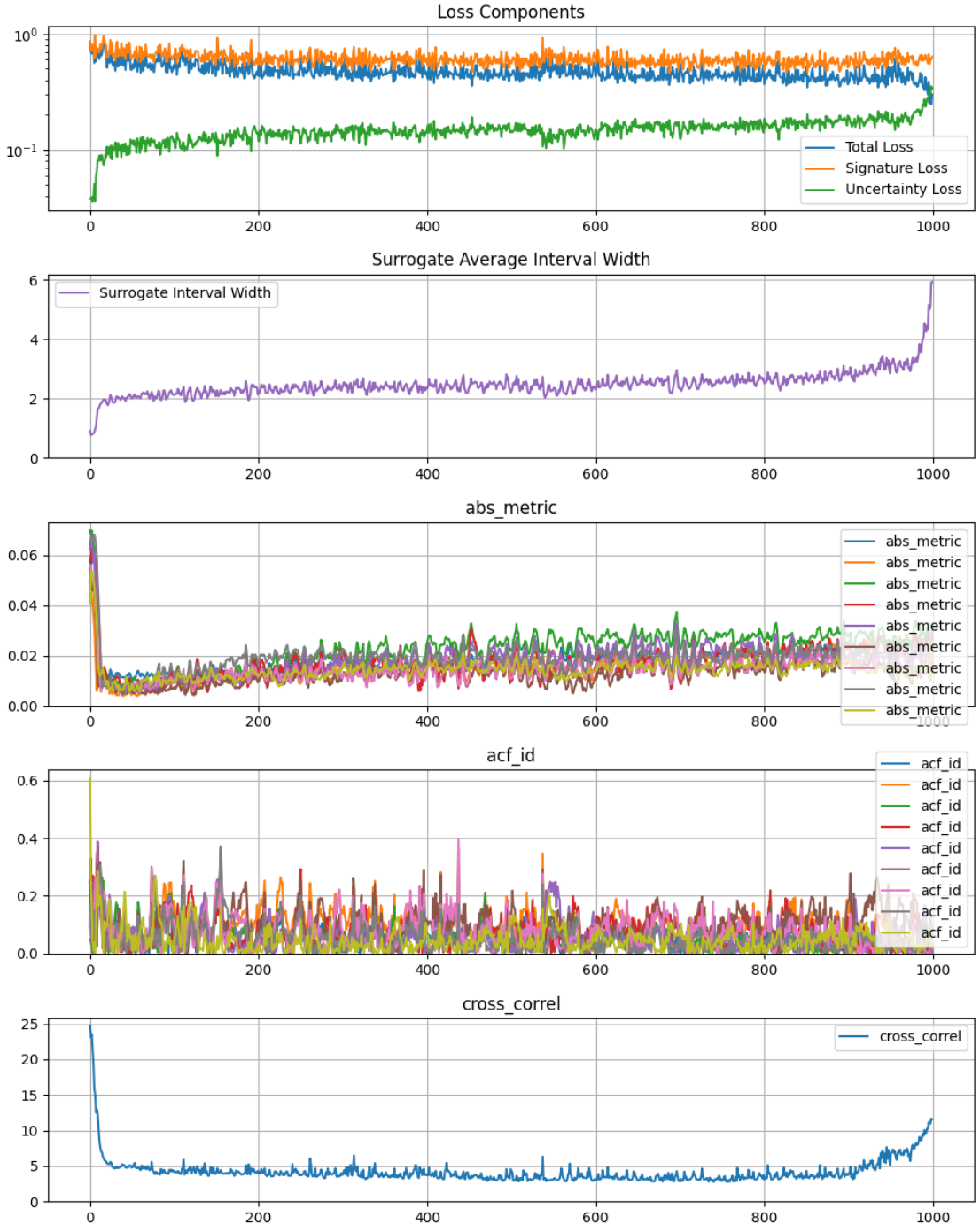

Figure A.7: Training diagnostics for $\lambda = 0.06$, $\alpha = 0.05$. Panels show (top to bottom): loss components (total, signature, uncertainty), forecaster APIW, absolute histogram distance (`abs_metric`), lag-1 autocorrelation (`acf_id`), and cross-correlation (`cross_correl`) over training iterations.

A.3 Distributional and Temporal Comparisons

The following figures compare the statistical properties of historical (blue) and synthetic (orange) data across all assets and experiments. Each row corresponds to one asset, and columns show: (i) the probability density function of returns, (ii) the log-scaled distribution, and (iii) the lag-1 autocorrelation function. The panels report empirical skewness (s) and kurtosis (k) values for historical and generated data, allowing visual assessment of the generator's ability to reproduce key distributional moments and temporal dependence. For illustration purposes, results are presented for $\alpha = 0.05$ across multiple values of the uncertainty regularization parameter λ, as this configuration is representative of the broader trends observed across confidence levels.

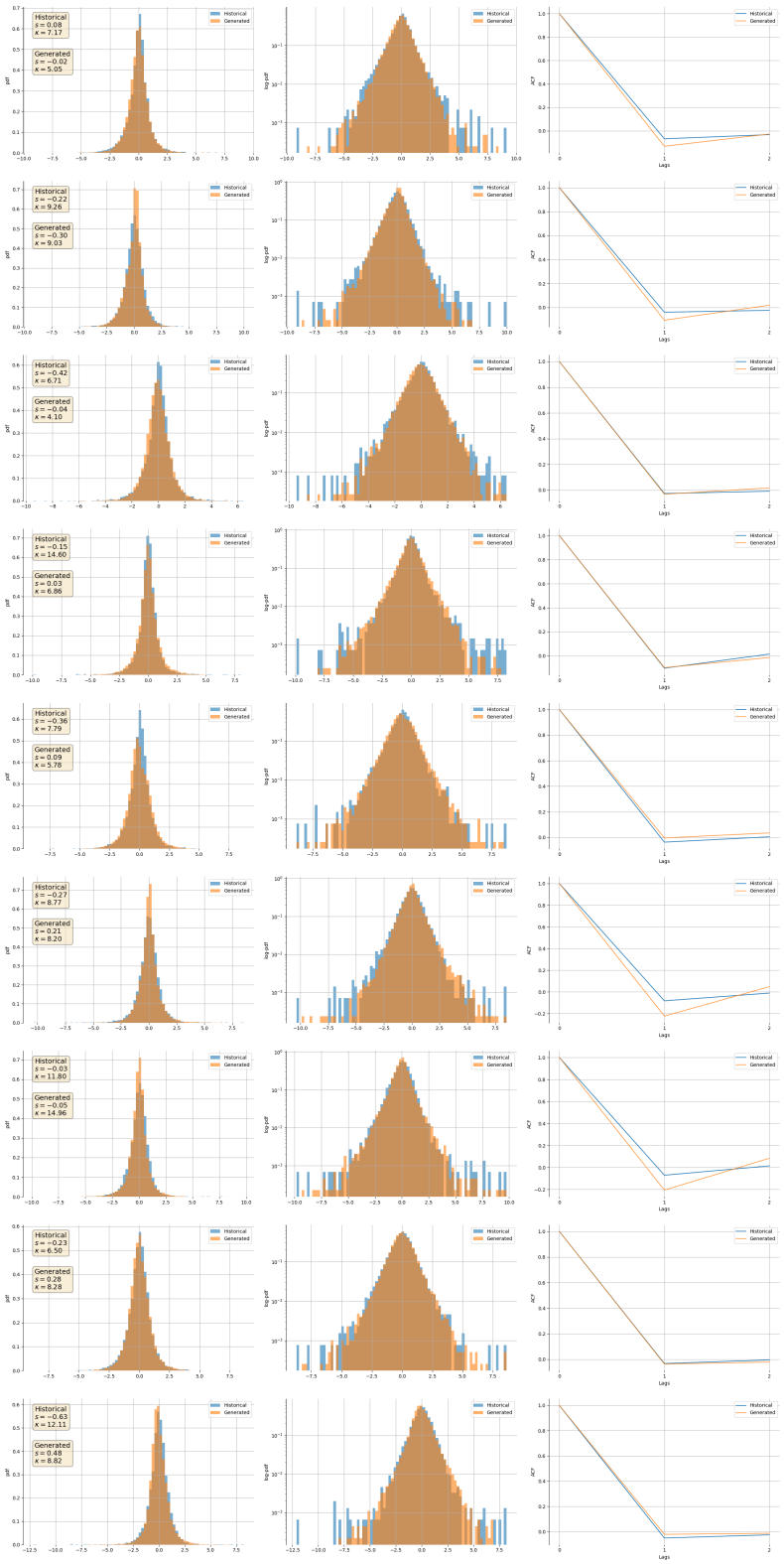

Figure A.8: Distributional and temporal structure comparison for $\lambda = 0$, $\alpha = 0.05$. Each row corresponds to a different time series, showing histogram alignment, log-scale distribution, and lag-1 autocorrelation between historical (blue) and generated (orange) returns.

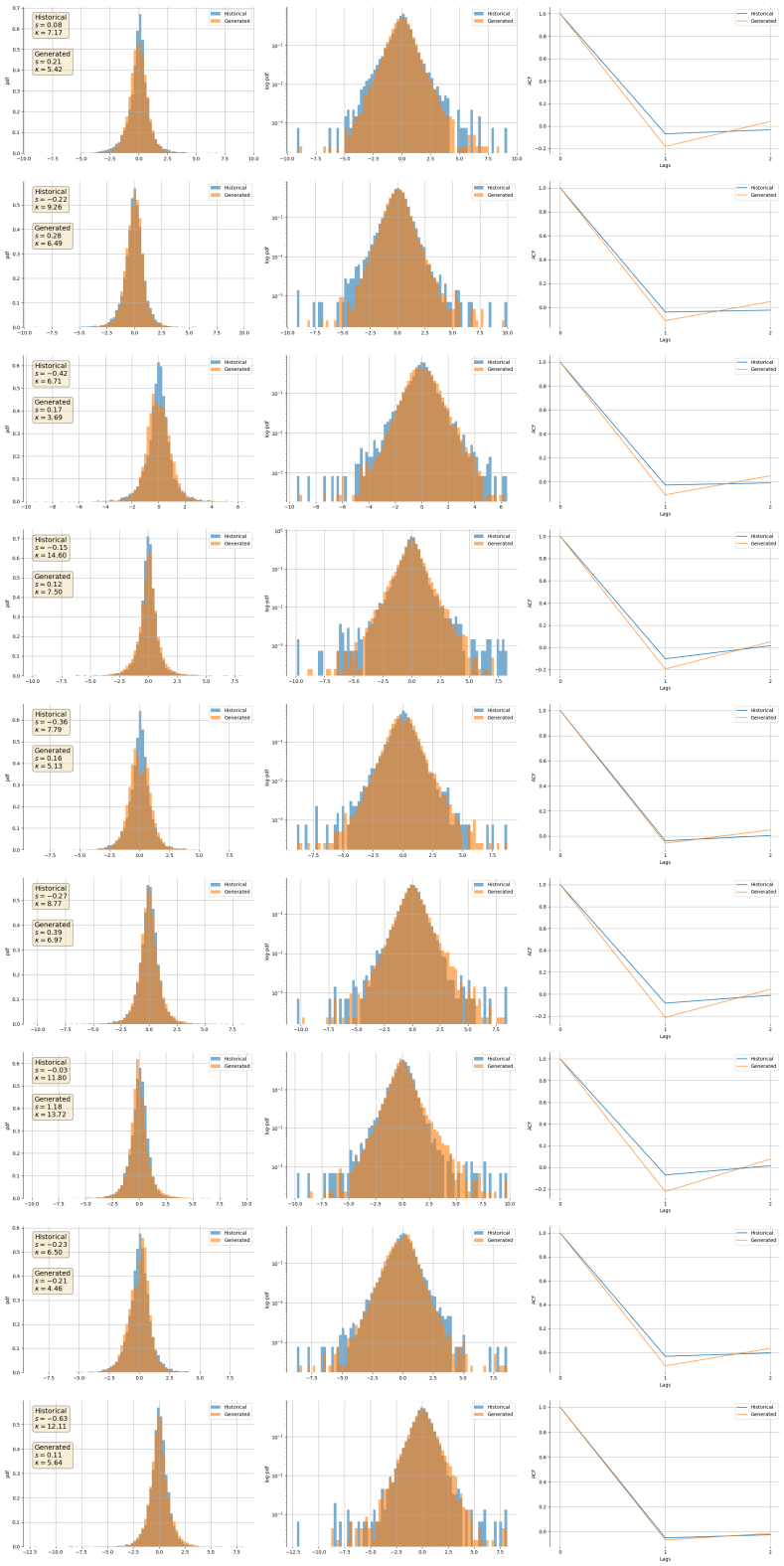

Figure A.9: Distributional and temporal structure comparison for $\lambda = 0.01$, $\alpha = 0.05$. Each row corresponds to a different time series, showing histogram alignment, log-scale distribution, and lag-1 autocorrelation between historical (blue) and generated (orange) returns.

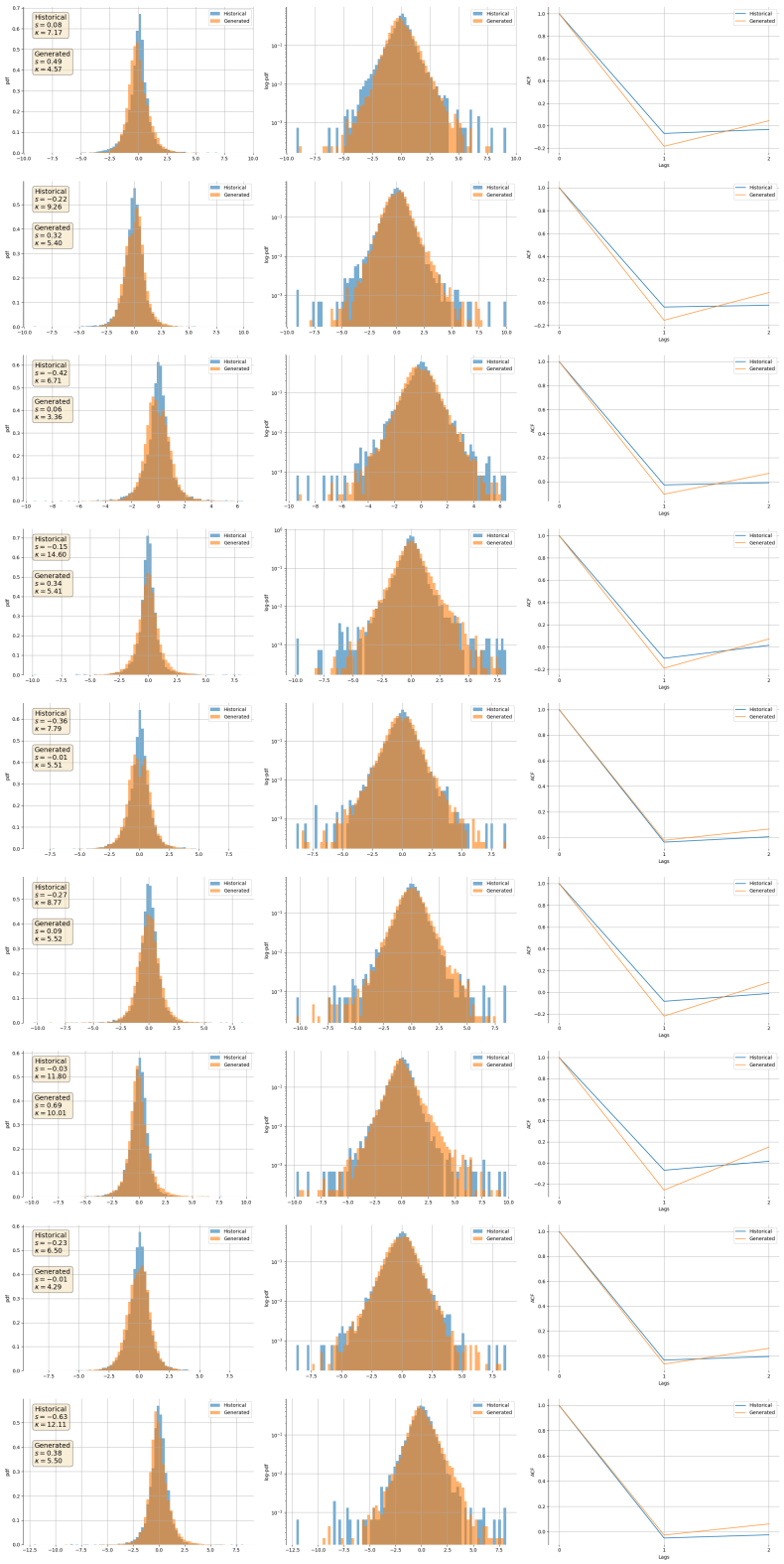

Figure A.10: Distributional and temporal structure comparison for $\lambda = 0.02$, $\alpha = 0.05$. Each row corresponds to a different time series, showing histogram alignment, log-scale distribution, and lag-1 autocorrelation between historical (blue) and generated (orange) returns.

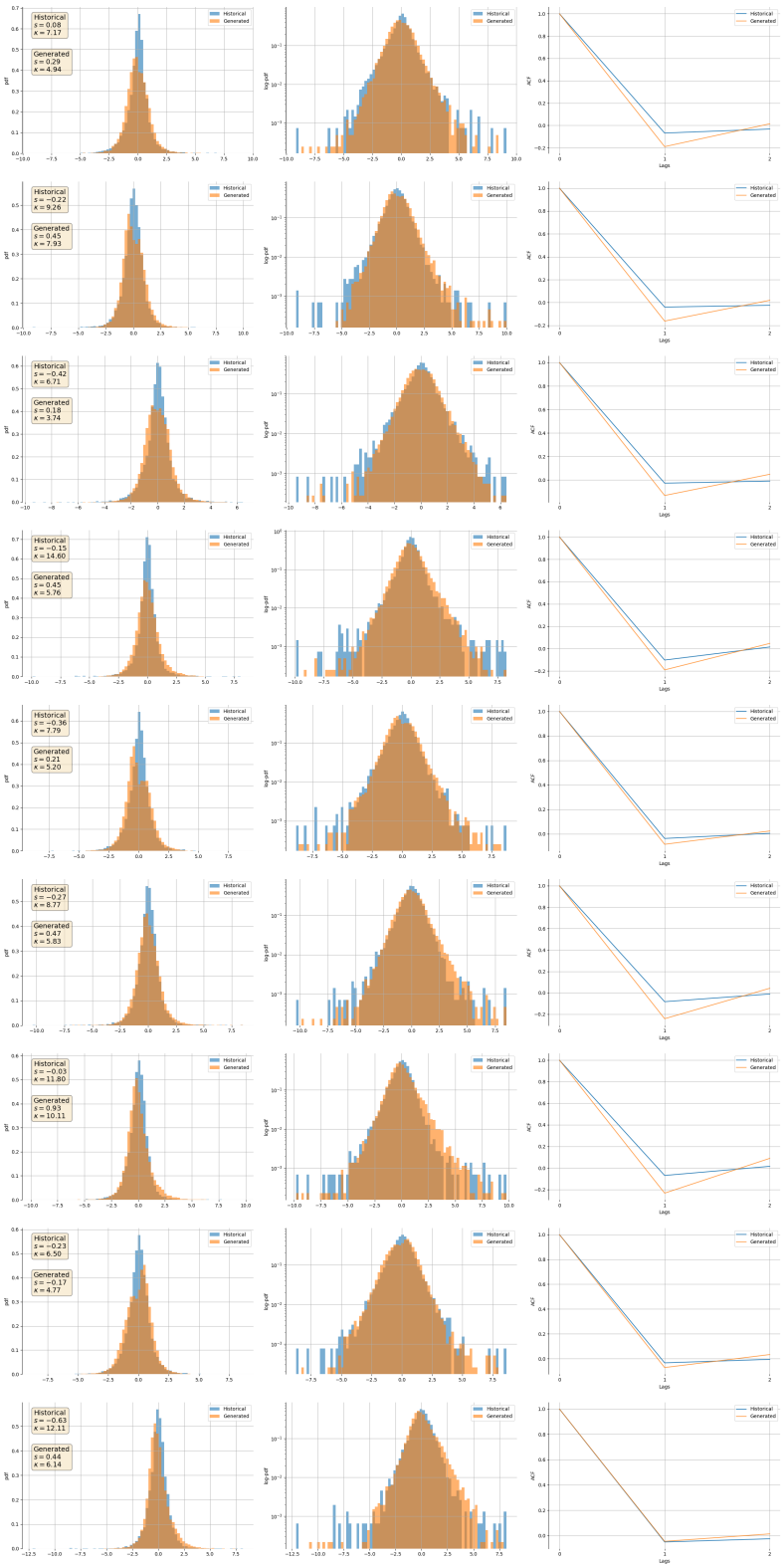

Figure A.11: Distributional and temporal structure comparison for $\lambda = 0.03$, $\alpha = 0.05$. Each row corresponds to a different time series, showing histogram alignment, log-scale distribution, and lag-1 autocorrelation between historical (blue) and generated (orange) returns.

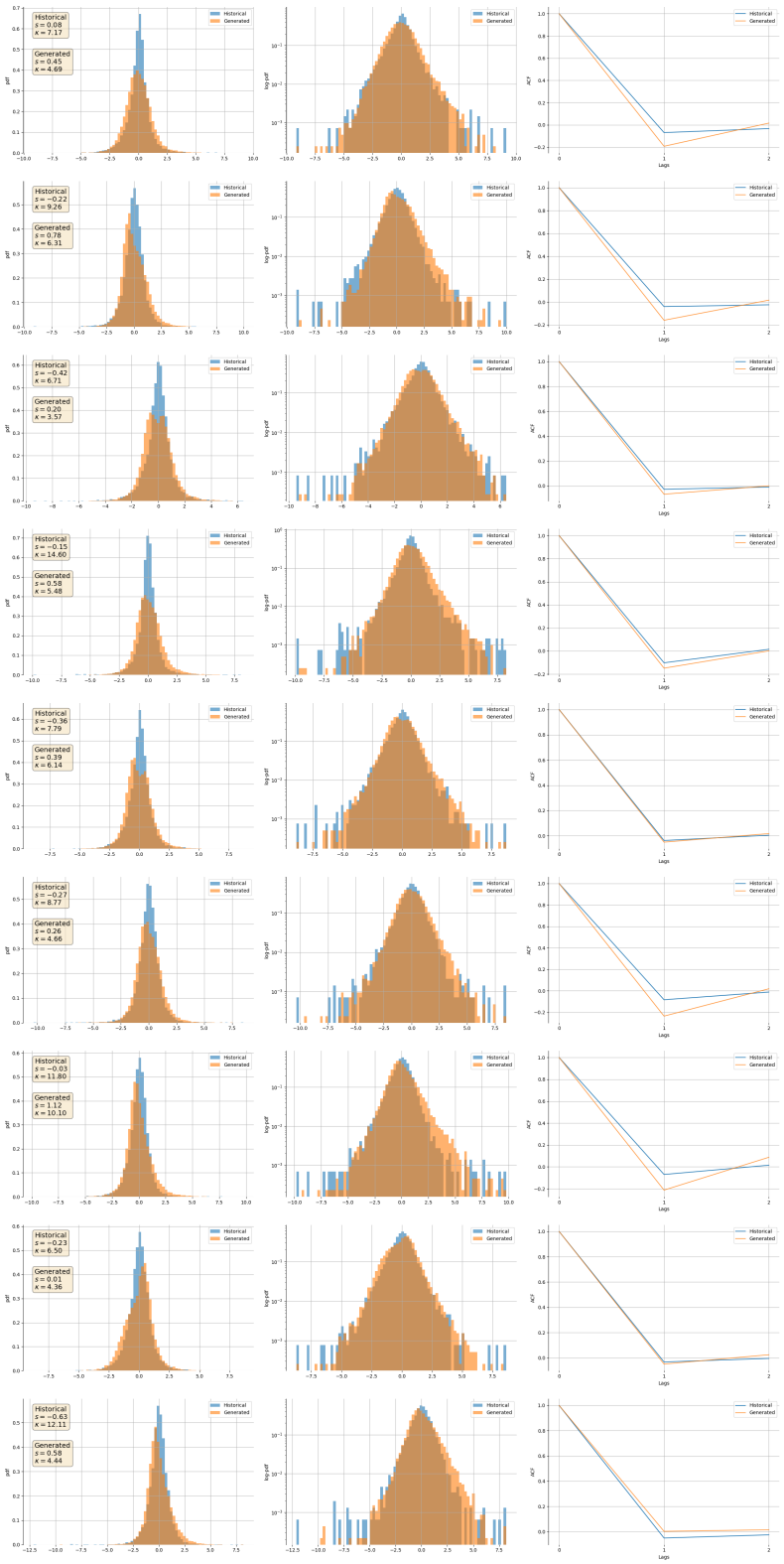

Figure A.12: Distributional and temporal structure comparison for $\lambda = 0.04$, $\alpha = 0.05$. Each row corresponds to a different time series, showing histogram alignment, log-scale distribution, and lag-1 autocorrelation between historical (blue) and generated (orange) returns.

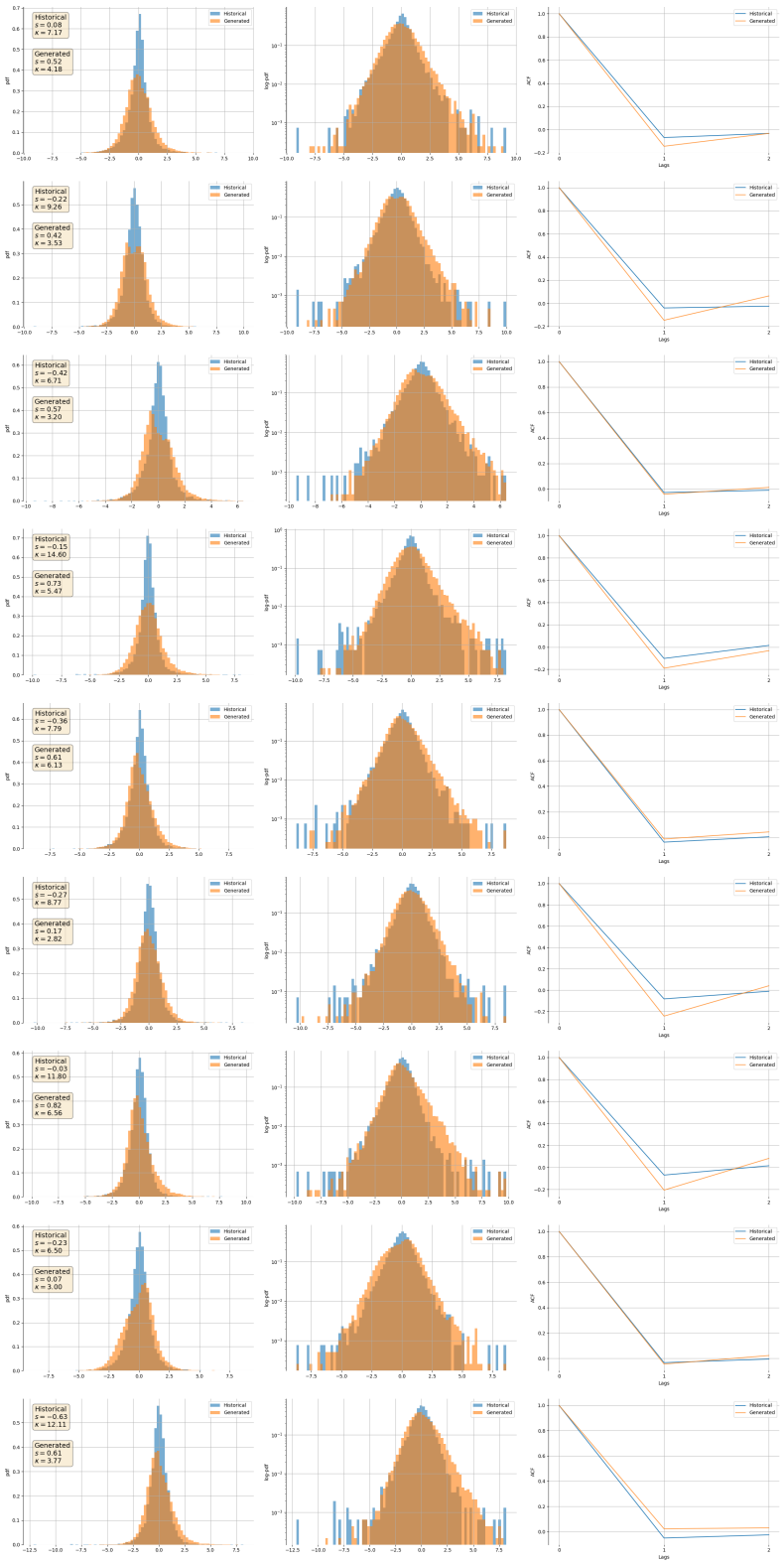

Figure A.13: Distributional and temporal structure comparison for $\lambda = 0.05$, $\alpha = 0.05$. Each row corresponds to a different time series, showing histogram alignment, log-scale distribution, and lag-1 autocorrelation between historical (blue) and generated (orange) returns.

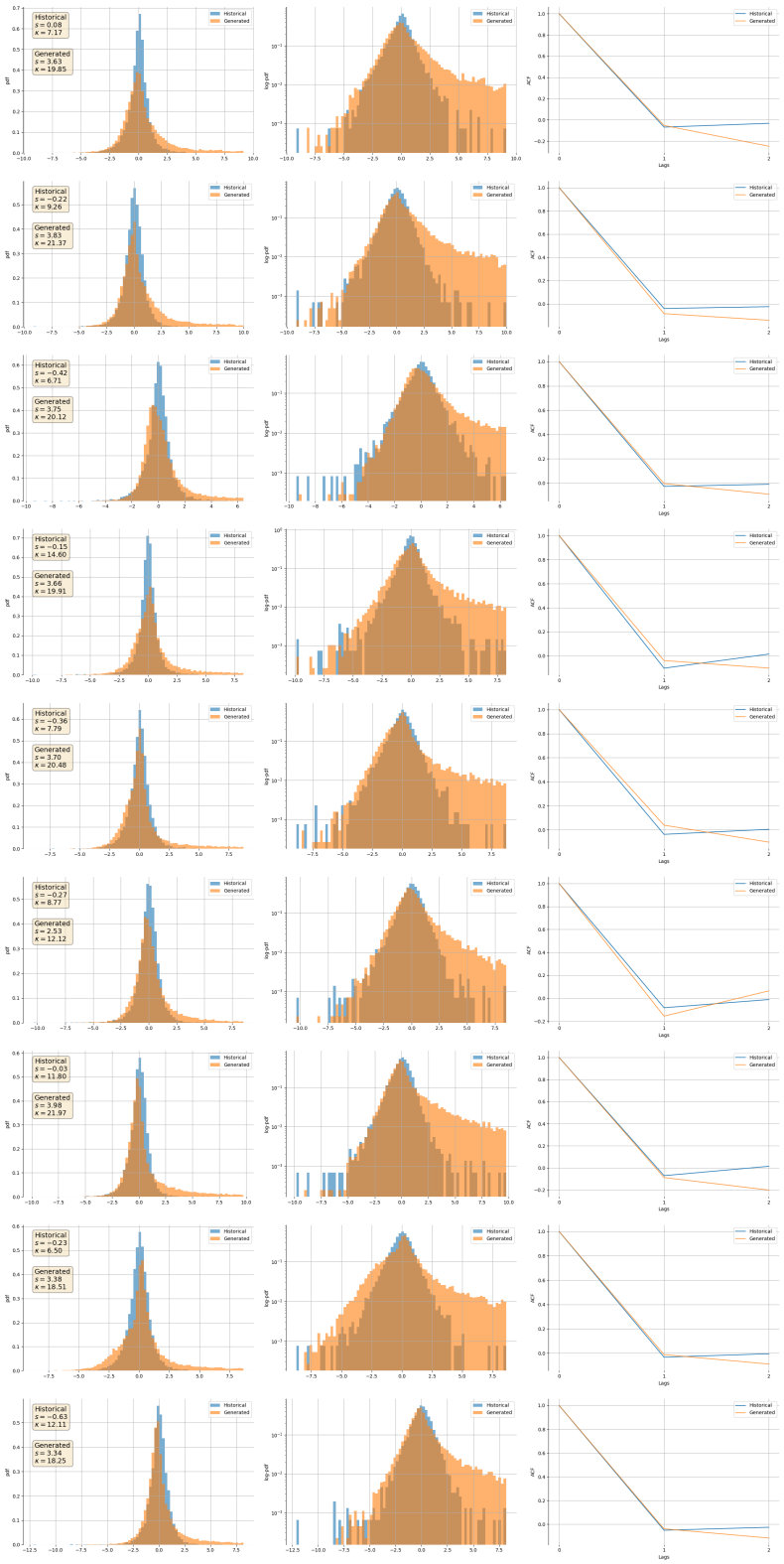

Figure A.14: Distributional and temporal structure comparison for $\lambda = 0.06$, $\alpha = 0.05$. Each row corresponds to a different time series, showing histogram alignment, log-scale distribution, and lag-1 autocorrelation between historical (blue) and generated (orange) returns.

A.3.1 Cross-Asset Correlation Structure

To complement the univariate and temporal comparisons, this subsection presents the cross-asset correlation matrices computed for both historical and synthetic datasets. These matrices capture the dependence structure between sectoral or asset-specific returns, providing a multivariate perspective on the fidelity of the generated data. Each figure displays side-by-side comparisons of empirical and synthetic correlation matrices for different uncertainty regularization strengths λ, with the same color scale to facilitate visual interpretation. For illustration purposes, results are shown for $\alpha = 0.05$, as they are representative of the correlation preservation patterns observed across confidence levels.

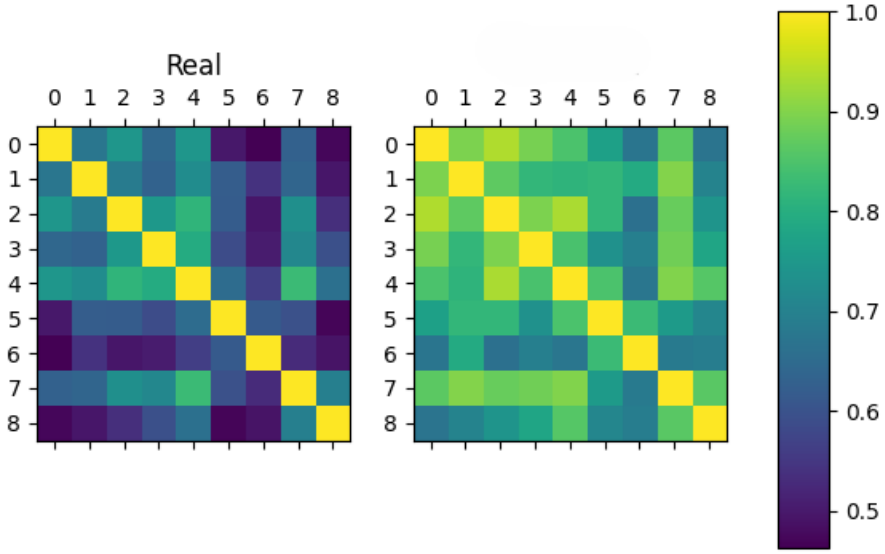

Figure A.15: Cross-asset correlation matrices for real (left) and synthetic (right) data generated with $\lambda = 0$, $\alpha = 0.05$.

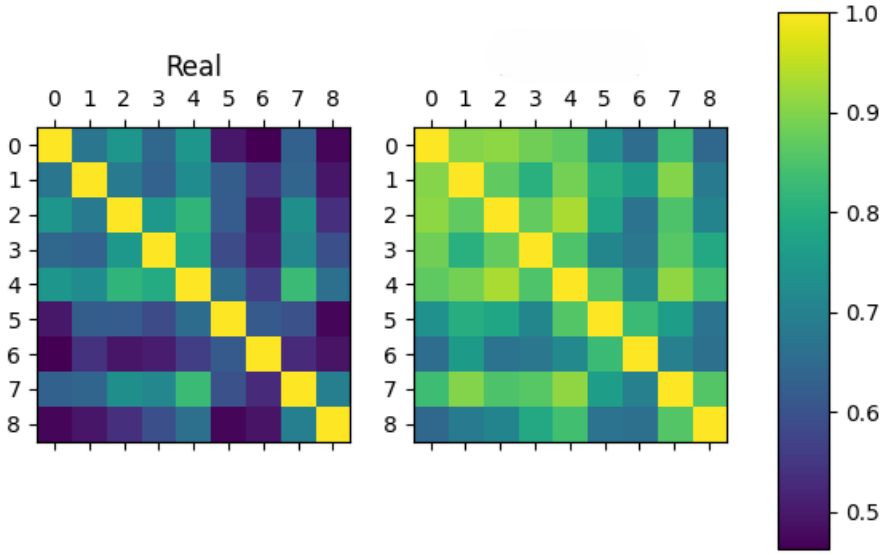

Figure A.16: Cross-asset correlation matrices for real (left) and synthetic (right) data generated with $\lambda = 0.01$, $\alpha = 0.05$.

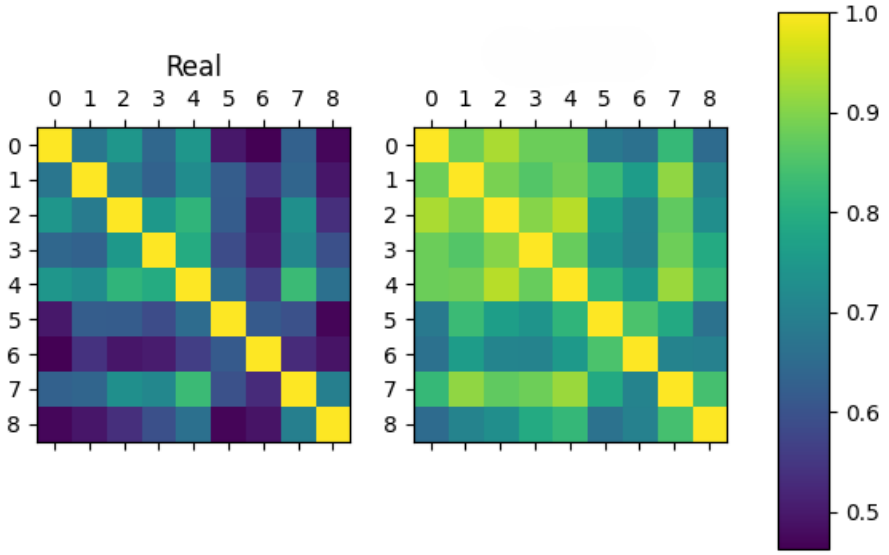

Figure A.17: Cross-asset correlation matrices for real (left) and synthetic (right) data generated with $\lambda = 0.02$, $\alpha = 0.05$.

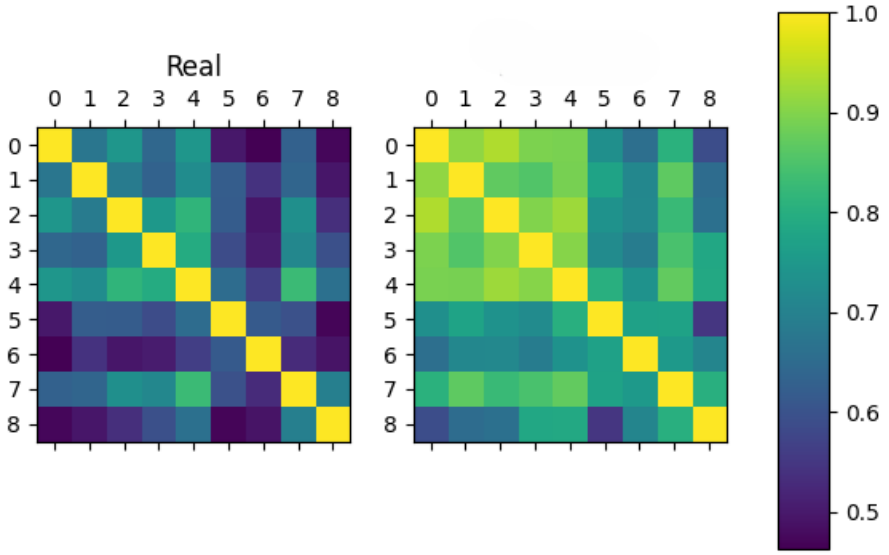

Figure A.18: Cross-asset correlation matrices for real (left) and synthetic (right) data generated with $\lambda = 0.03$, $\alpha = 0.05$.

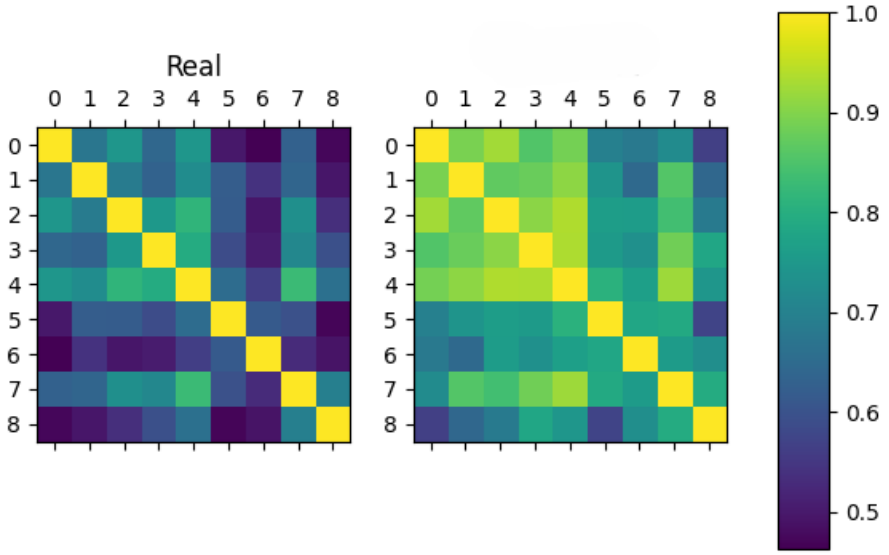

Figure A.19: Cross-asset correlation matrices for real (left) and synthetic (right) data generated with $\lambda = 0.04$, $\alpha = 0.05$.

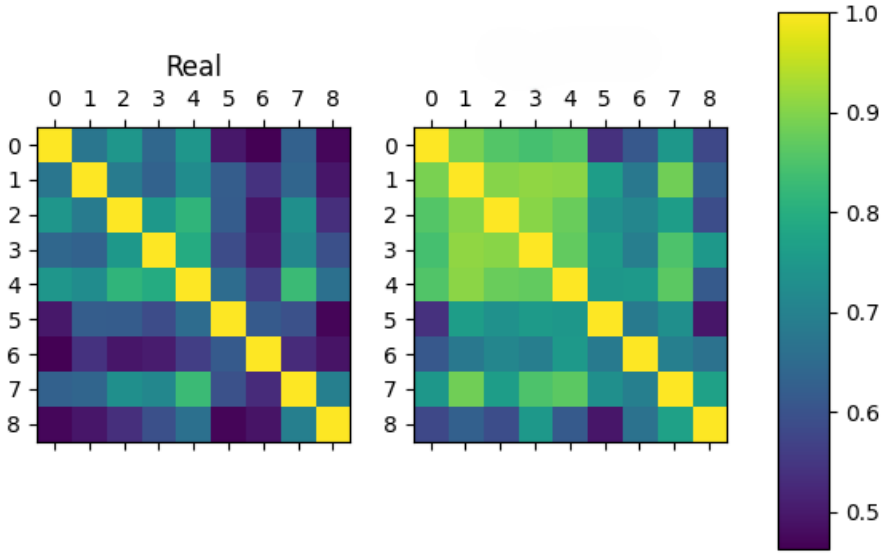

Figure A.20: Cross-asset correlation matrices for real (left) and synthetic (right) data generated with $\lambda = 0.05$, $\alpha = 0.05$.

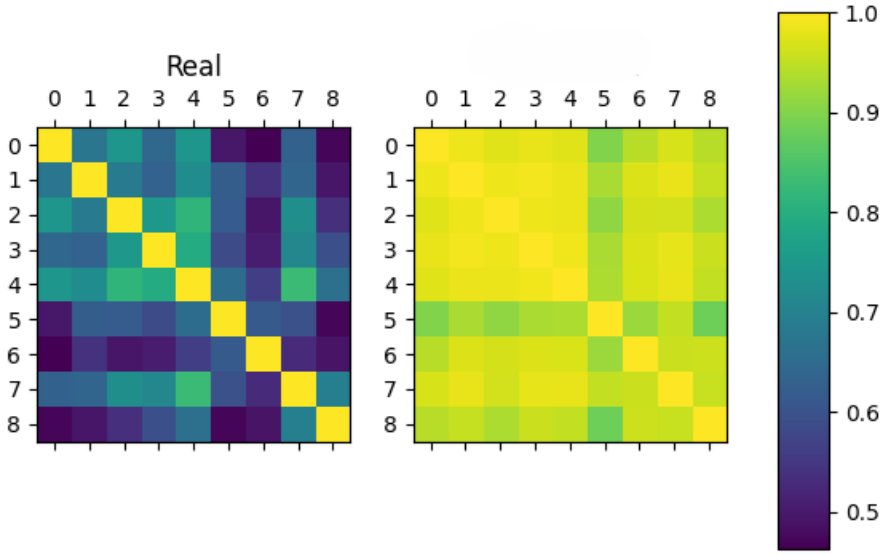

Figure A.21: Cross-asset correlation matrices for real (left) and synthetic (right) data generated with $\lambda = 0.06$, $\alpha = 0.05$.

References

[1] Decision making under uncertainty: Theory and application. 2015. URL https://api.semanticscholar.org/CorpusID:64210316.

[2] Beatrice Acciaio, Stephan Eckstein, and Songyan Hou. Time-causal vae: Robust financial time series generator. *arXiv preprint arXiv:2411.02947*, 2024.

[3] Ratnadip Adhikari and R. K. Agrawal. An introductory study on time series modeling and forecasting, 2013. URL https://arxiv.org/abs/1302.6613.

[4] Nesreen Ahmed, Amir F. Atiya, Neamat El Gayar, and Hisham El-Shishiny. An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29:594 – 621, 2010. URL https://api.semanticscholar.org/CorpusID:15917396.

[5] Rami Al-Hmouz, Witold Pedrycz, and Abdullah Saeed Balamash. Description and prediction of time series: A general framework of granular computing. *Expert Syst. Appl.*, 42:4830–4839, 2015. URL https://api.semanticscholar.org/CorpusID:27652119.

[6] R. Cont and. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 1(2):223–236, 2001. doi: 10.1080/713665670. URL https://doi.org/10.1080/713665670.

[7] Adebiyi Ariyo Ariyo, Aderemi Oluyinka Adewumi, and Charles K. Ayo. Stock price prediction using the arima model. *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, pages 106–112, 2014. URL https://api.semanticscholar.org/CorpusID:6767688.

[8] Martín Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *ArXiv*, abs/1701.04862, 2017. URL https://doi.org/10.48550/arXiv.1701.04862.

[9] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, 2017. URL https://doi.org/10.48550/arXiv.1701.07875.

[10] Matthew Arnold, Rachel K. E. Bellamy, Michael Hind, Stephanie Houde, Sameep Mehta, Aleksandra Mojsilovic, Ravindranath Gopinathan Nair, Karthikeyan Natesan Ramamurthy, Alexandra Olteanu, David Piorkowski, Darrell Reimer, John T. Richards, Jason Tsay, and Kush R. Varshney. Factsheets: Increasing trust in ai services through supplier's declarations of conformity. *IBM J. Res. Dev.*, 63:6:1–6:13, 2019. URL https://api.semanticscholar.org/CorpusID:263502962.

[11] Yahui Bai, Yuhe Gao, Runzhe Wan, Sheng Zhang, and Rui Song. A review of reinforcement learning in financial applications. *Annual Review of Statistics and Its Application*, 12(1): 209–232, 2025.

[12] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2019.12.012. URL https://www.sciencedirect.com/science/article/pii/S1566253519308103.

[13] Horatio Boedihardjo and Xi Geng. The uniqueness of signature problem in the non-markov setting. *Stochastic Processes and their Applications*, 125(12):4674–4701, 2015.

[14] Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31:307–327, 1986. URL https://api.semanticscholar.org/CorpusID:8797625.

[15] Ali Borji. Pros and cons of gan evaluation measures. *Computer vision and image understanding*, 179:41–65, 2019.

[16] Ali Borji. Pros and cons of gan evaluation measures: New developments. *Computer Vision and Image Understanding*, 215:103329, 2022.

[17] Oscar Cartagena, Sebastián Parra, Diego Muñoz-Carpintero, Luis Gabriel Marín, and Doris Śaez. Review on fuzzy and neural prediction interval modelling for nonlinear dynamical systems. *IEEE Access*, 9:23357–23384, 2021. URL https://api.semanticscholar.org/CorpusID:231914190.

[18] Anirban Chakraborti, Ioane Muni Toke, Marco Patriarca, and Frédéric Abergel. Econophysics: empirical facts. *Quantitative Finance*, 2011. URL https://api.semanticscholar.org/CorpusID:261025294.

[19] Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza, Max Mergenthaler-Canseco, and Artur Dubrawski. N-hits: Neural hierarchical interpolation for time series forecasting, 2022. URL https://arxiv.org/abs/2201.12886.

[20] James Ming Chen and Charalampos Agiropoulos. Hints of earlier and other creation: Unsupervised machine learning in financial time-series analysis. *Engineering Proceedings*, 39 (1):42, 2023.

[21] Victor Chernozhukov, Kaspar Wüthrich, and Yinchu Zhu. Distributional conformal prediction. *Proceedings of the National Academy of Sciences*, 118, 2019. URL https://api.semanticscholar.org/CorpusID:202583745.

[22] Ilya Chevyrev and Andrey Kormilitzin. A primer on the signature method in machine learning. *arXiv preprint arXiv:1603.03788*, 2016.

[23] Ilya Chevyrev and Harald Oberhauser. Signature moments to characterize laws of stochastic processes. *Journal of Machine Learning Research*, 23(176):1–42, 2022.

[24] Rama Cont. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 1:223 – 236, 2001. URL https://api.semanticscholar.org/CorpusID:5625400.

[25] Rama Cont. Volatility clustering in financial markets: Empirical facts and agent-based models. *Capital Markets: Market Microstructure*, 2005. URL https://api.semanticscholar.org/CorpusID:16345695.

[26] Rama Cont, Marc Potters, and Jean-Philippe Bouchaud. Scaling in stock market data: stable laws and beyond. In *Scale Invariance and Beyond: Les Houches Workshop, March 10–14, 1997*, pages 75–85. Springer, 1997.

[27] Luís Cunha, Carlos Soares, André Restivo, and Luís F. Teixeira. Gasten: Generative adversarial stress test networks. In *International Symposium on Intelligent Data Analysis*, 2023. URL https://api.semanticscholar.org/CorpusID:258440686.

[28] Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *Journal of Machine Learning Research*, 23(226):1–61, 2022.

[29] Fernando de Meer Pardo. Enriching financial datasets with generative adversarial networks. 2019. URL https://api.semanticscholar.org/CorpusID:199010183.

[30] Fernando de Meer Pardo, Peter Schwendner, and Marcus Wunsch. Tackling the exponential scaling of signature-based generative adversarial networks for high-dimensional financial time-series generation. *Journal of Financial Data Science*, 4(4), 2022.

[31] Akash TTU Deep. Advanced financial market forecasting: integrating monte carlo simulations with ensemble machine learning models. 2024.

[32] Stefan Depeweg. Modeling epistemic and aleatoric uncertainty with bayesian neural networks and latent variables. 2019. URL https://api.semanticscholar.org/CorpusID:208224498.

[33] Nicolas Dewolf, Bernard De Baets, and Willem Waegeman. Valid prediction intervals for regression problems. *Artificial Intelligence Review*, 56:577–613, 2021. URL https://api.semanticscholar.org/CorpusID:245124696.

[34] Mihai Dogariu, Liviu-Daniel Ştefan, Bogdan Andrei Boteanu, Claudiu Lamba, Bomi Kim, and Bogdan Ionescu. Generation of realistic synthetic financial time-series. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 18(4):1–27, 2022.

[35] Georgios Douzas and Fernando Bação. Effective data generation for imbalanced learning using conditional generative adversarial networks. *Expert Syst. Appl.*, 91:464–471, 2018. URL https://api.semanticscholar.org/CorpusID:27011880.

[36] Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning, 2020. *arXiv preprint arXiv:2011.03395*, 2011.

[37] Arul Earnest, Mark I. C. Chen, Donald Ng, and Leo Yee Sin. Using autoregressive integrated moving average (arima) models to predict and monitor the number of beds occupied during a sars outbreak in a tertiary hospital in singapore. *BMC Health Services Research*, 5: 36 – 36, 2005. URL https://api.semanticscholar.org/CorpusID:2733128.

[38] Florian Eckerli and Joerg Osterrieder. Generative adversarial networks in finance: an overview. *arXiv preprint arXiv:2106.06364*, 2021.

[39] Peter Eickelberg. Volatility, correlation, and diversification in a multi-factor world. *Cfa Digest*, 43, 2013. URL https://api.semanticscholar.org/CorpusID:155978781.

[40] Robert F. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50:987–1007, 1982. URL https://api.semanticscholar.org/CorpusID:18673159.

[41] Robert F. Engle and Clive William John Granger. Co-integration and error correction: representation, estimation and testing. *Econometrica*, 55:251–276, 1987. URL https://api.semanticscholar.org/CorpusID:16616066.

[42] Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional gans. *ArXiv*, abs/1706.02633, 2017. URL https://doi.org/10.48550/arXiv.1706.02633.

[43] Ugo Fiore, Alfredo De Santis, Francesca Perla, Paolo Zanetti, and Francesco Palmieri. Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. *Inf. Sci.*, 479:448–455, 2017. URL https://api.semanticscholar.org/CorpusID:59528925.

[44] Thomas G. Fischer and Christopher Krauss. Deep learning with long short-term memory networks for financial market predictions. *Eur. J. Oper. Res.*, 270:654–669, 2017. URL https://api.semanticscholar.org/CorpusID:207640363.

[45] Sebastian Flennerhag, Jane X. Wang, Pablo Sprechmann, Francesco Visin, Alexandre Galashov, Steven Kapturowski, Diana Borsa, Nicolas Manfred Otto Heess, André Barreto, and Razvan Pascanu. Temporal difference uncertainties as a signal for exploration. *ArXiv*, abs/2010.02255, 2020. URL https://api.semanticscholar.org/CorpusID:222140936.

[46] Javier Franco-Pedroso, Joaquín González-Rodríguez, Jorge Cubero, Maria Planas, Rafael Cobo, and Fernando Pablos. Generating virtual scenarios of multivariate financial data for quantitative trading applications. In *The Journal of Financial Data Science*, 2018. URL https://api.semanticscholar.org/CorpusID:36261311.

[47] Rao Fu, Jie Chen, Shutian Zeng, Yiping Zhuang, and A. Sudjianto. Time series simulation by conditional generative adversarial net. *ERN: Time-Series Models (Single) (Topic)*, 2019. URL https://api.semanticscholar.org/CorpusID:131778163.

[48] Yarin Gal. Uncertainty in deep learning. 2016. URL https://api.semanticscholar.org/CorpusID:86522127.

[49] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna M. Wallach, Hal Daumé, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64:86 – 92, 2018. URL https://api.semanticscholar.org/CorpusID:4421027.

[50] Chuanxing Geng, Sheng-Jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:3614–3631, 2018. URL https://api.semanticscholar.org/CorpusID:53734567.

[51] Isaac Gibbs and Emmanuel J. Candès. Adaptive conformal inference under distribution shift. In *Neural Information Processing Systems*, 2021. URL https://api.semanticscholar.org/CorpusID:235266057.

[52] Ethan Goan and Clinton Fookes. Bayesian neural networks: An introduction and survey. *ArXiv*, abs/2006.12024, 2020. URL https://api.semanticscholar.org/CorpusID:219873953.

[53] Rakshitha Godahewa, Kasun Bandara, Geoffrey I. Webb, Slawek Smyl, and Christoph Bergmeir. Ensembles of localised models for time series forecasting. *Knowledge-Based Systems*, 233:107518, December 2021. ISSN 0950-7051. doi: 10.1016/j.knosys.2021.107518. URL http://dx.doi.org/10.1016/j.knosys.2021.107518.

[54] Inês Gomes, Luís F Teixeira, Jan N Van Rijn, Carlos Soares, André Restivo, Luís Cunha, and Moisés Santos. Finding patterns in ambiguity: Interpretable stress testing in the decision boundary. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8316–8321, 2024.

[55] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Neural Information Processing Systems*, 2014. URL https://doi.org/10.48550/arXiv.1406.2661.

[56] Jan G. De Gooijer and Rob J Hyndman. 25 years of time series forecasting. *International Journal of Forecasting*, 22:443–473, 2006. URL https://api.semanticscholar.org/CorpusID:14996235.

[57] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. In *Neural Information Processing Systems*, 2017. URL https://doi.org/10.48550/arXiv.1704.00028.

[58] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017. URL https://api.semanticscholar.org/CorpusID:28671436.

[59] Erkam Güresen, Gulgun Kayakutlu, and Tugrul Unsal Daim. Using artificial neural network models in stock market index prediction. *Expert Syst. Appl.*, 38:10389–10397, 2011. URL https://api.semanticscholar.org/CorpusID:16016785.

[60] Swaminathan Gurumurthy, Ravi Kiran Sarvadevabhatla, and R. Venkatesh Babu. Deligan: Generative adversarial networks for diverse and limited data. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4941–4949, 2017. URL https://api.semanticscholar.org/CorpusID:24527328.

[61] Ben Hambly and Terry Lyons. Uniqueness for the signature of a path of bounded variation and the reduced path group. *Annals of Mathematics*, pages 109–167, 2010.

[62] Kay Gregor Hartmann, Robin Tibor Schirrmeister, and Tonio Ball. Eeg-gan: Generative adversarial networks for electroencephalograhic (eeg) brain signals. *ArXiv*, abs/1806.01875, 2018. URL https://api.semanticscholar.org/CorpusID:46945788.

[63] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 41–50, 2018. URL https://api.semanticscholar.org/CorpusID:55700923.

[64] José Miguel Hernández-Lobato, Matthew W. Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. *ArXiv*, abs/1406.2541, 2014. URL https://api.semanticscholar.org/CorpusID:1776111.

[65] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Neural Information Processing Systems*, 2017. URL https://api.semanticscholar.org/CorpusID:326772.

[66] David Hirshleifer and Siew Hong Teoh. Herd behaviour and cascading in capital markets: A review and synthesis. *Capital Markets: Market Efficiency eJournal*, 2003. URL https://api.semanticscholar.org/CorpusID:28553134.

[67] John Hunter, Simon P Burke, and Alessandra Canepa. *Multivariate modelling of non-stationary economic time series*. Springer, 2017.

[68] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[69] Guillermo Iglesias, Edgar Talavera, Ángel González-Prieto, Alberto Mozo, and Sandra Gómez-Canaval. Data augmentation techniques in time series domain: a survey and taxonomy. *Neural Computing and Applications*, 35(14):10123–10145, 2023.

[70] Brian Kenji Iwana and Seiichi Uchida. An empirical survey of data augmentation for time series classification with neural networks. *PLoS ONE*, 16, 2020. URL https://api.semanticscholar.org/CorpusID:220920067.

[71] Tim Januschowski, Jan Gasthaus, Yuyang Wang, David Salinas, Valentin Flunkert, Michael Bohlke-Schneider, and Laurent Callot. Criteria for classifying forecasting methods, 2022. URL https://arxiv.org/abs/2212.03523.

[72] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiao Long Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-llm: Time series forecasting by reprogramming large language models. *ArXiv*, abs/2310.01728, 2023. URL https://api.semanticscholar.org/CorpusID:263609325.

[73] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[74] Christopher Kath and Florian Ziel. Conformal prediction interval estimation and applications to day-ahead and intraday power markets. *International Journal of Forecasting*, 1, 2019. URL https://api.semanticscholar.org/CorpusID:159042040.

[75] Taranjot Kaur, Gurmeet Kaur, et al. Forecasting financial market using machine learning techniques. 2024.

[76] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *ArXiv*, abs/1703.04977, 2017. URL https://api.semanticscholar.org/CorpusID:71134.

[77] Mehdi Khashei and Mehdi Bijari. A novel hybridization of artificial neural networks and arima models for time series forecasting. *Appl. Soft Comput.*, 11:2664–2675, 2011. URL https://api.semanticscholar.org/CorpusID:6186690.

[78] Patrick Kidger and Terry Lyons. Signatory: differentiable computations of the signature and logsignature transforms, on both CPU and GPU. In *International Conference on Learning Representations*, 2021. https://github.com/patrick-kidger/signatory.

[79] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International conference on learning representations*, 2021.

[80] Armen Der Kiureghian and O. Ditlevsen. Aleatory or epistemic? does it matter? *Structural Safety*, 31:105–112, 2009. URL https://api.semanticscholar.org/CorpusID:55548189.

[81] Roger W. Koenker and Kevin F. Hallock. Quantile regression. 2001. URL https://api.semanticscholar.org/CorpusID:155050906.

[82] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, 2017. URL https://api.semanticscholar.org/CorpusID:13193974.

[83] Adriano Koshiyama, Nick Firoozye, and Philip Treleaven. Generative adversarial networks for financial trading strategies fine-tuning and combination. *Quantitative Finance*, 21(5): 797–813, 2021.

[84] Adriano Soares Koshiyama, Nikan B. Firoozye, and Philip C. Treleaven. Generative adversarial networks for financial trading strategies fine-tuning and combination. *Quantitative Finance*, 21:797 – 813, 2019. URL https://api.semanticscholar.org/CorpusID:57573848.

[85] Sang Il Lee and Seong Joon Yoo. A deep efficient frontier method for optimal investments. 2017. URL https://api.semanticscholar.org/CorpusID:158179537.

[86] Pedro Lencastre, Marit Gjersdal, Leonardo Rydin Gorjão, Anis Yazidi, and Pedro G Lind. Modern ai versus century-old mathematical models: How far can we go with generative adversarial networks to reproduce stochastic processes? *Physica D: Nonlinear Phenomena*, 453:133831, 2023.

[87] Daniel Levin, Terry Lyons, and Hao Ni. Learning from the past, predicting the statistics for the future, learning an evolving system. *arXiv preprint arXiv:1309.0260*, 2013.

[88] Junyi Li, Xintong Wang, Yaoyang Lin, Arunesh Sinha, and Michael P. Wellman. Generating realistic stock market order streams. In *AAAI Conference on Artificial Intelligence*, 2020. URL https://api.semanticscholar.org/CorpusID:53369150.

[89] Xiaomin Li, Vangelis Metsis, Huan Wang, and Anne H. H. Ngu. Tts-gan: A transformer-based time-series generative adversarial network. In *Conference on Artificial Intelligence in Medicine in Europe*, 2022. URL https://api.semanticscholar.org/CorpusID:246634793.

[90] Shujian Liao, Hao Ni, Lukasz Szpruch, Magnus Wiese, Marc Sabate-Vidales, and Baoren Xiao. Conditional sig-wasserstein gans for time series generation. *arXiv preprint arXiv:2006.05421*, 2020.

[91] Bingqing Liu, Ivan Kiskin, and Stephen Roberts. An overview of gaussian process regression for volatility forecasting. *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 681–686, 2020. URL https://api.semanticscholar.org/CorpusID:215817495.

[92] Jeremiah Zhe Liu, John William Paisley, Marianthi-Anna Kioumourtzoglou, and B. Coull. Accurate uncertainty estimation and decomposition in ensemble learning. *ArXiv*, abs/1911.04061, 2019. URL https://api.semanticscholar.org/CorpusID:202785571.

[93] Ana Lucic, Madhulika Srikumar, Umang Bhatt, Alice Xiang, Ankur Taly, Q. Vera Liao, and Maarten de Rijke. A multistakeholder approach towards evaluating ai transparency mechanisms. *ArXiv*, abs/2103.14976, 2021. URL https://api.semanticscholar.org/CorpusID:232403994.

[94] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. In *Neural Information Processing Systems*, 2017. URL https://api.semanticscholar.org/CorpusID:4053393.

[95] Spyros Makridakis and Nikolas Bakas. Forecasting and uncertainty: A survey. *Risk and Decision Analysis*, 6(1):37–64, 2016.

[96] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The m4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34(4):802–808, 2018. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2018.06.001. URL https://www.sciencedirect.com/science/article/pii/S0169207018300785.

[97] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, 38(4):1346–1364, 2022. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2021.11.013. URL https://www.sciencedirect.com/science/article/pii/S0169207021001874. Special Issue: M5 competition.

[98] Andrey Malinin and Mark John Francis Gales. Predictive uncertainty estimation via prior networks. In *Neural Information Processing Systems*, 2018. URL https://api.semanticscholar.org/CorpusID:3580844.

[99] Hans Malmsten and Timo Teräsvirta. Stylized facts of financial time series and three popular models of volatility. *European Journal of Pure and Applied Mathematics*, 3:443–477, 2004. URL https://api.semanticscholar.org/CorpusID:8281446.

[100] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2813–2821, 2016. URL https://api.semanticscholar.org/CorpusID:206771128.

[101] Giovanni Mariani, Yada Zhu, Jianbo Li, Florian Scheidegger, Roxana Istrate, Costas Bekas, and A Cristiano I Malossi. Pagan: Portfolio analysis with generative adversarial networks. *arXiv preprint arXiv:1909.10578*, 2019.

[102] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *ArXiv*, abs/1411.1784, 2014. URL https://api.semanticscholar.org/CorpusID:12803511.

[103] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2018. URL https://api.semanticscholar.org/CorpusID:52946140.

[104] Olof Mogren. C-rnn-gan: Continuous recurrent neural networks with adversarial training. *ArXiv*, abs/1611.09904, 2016. URL https://doi.org/10.48550/arXiv.1611.09904.

[105] Falak Naaz, Aniruddh Herle, Janamejaya Channegowda, Aditya Raj, and Meenakshi Lakshminarayanan. A generative adversarial network-based synthetic data augmentation technique for battery condition evaluation. *International Journal of Energy Research*, 45: 19120 – 19135, 2021. URL https://api.semanticscholar.org/CorpusID:237830133.

[106] Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, 2015:2901–2907, 2015. URL https://api.semanticscholar.org/CorpusID:6292807.

[107] Roberto I. Oliveira, Paulo Orenstein, Thiago Rodrigo Ramos, and João Vitor Romano. Split conformal prediction for dependent data. 2022. URL https://api.semanticscholar.org/CorpusID:247794086.

[108] Eyas Gaffar A Osman and Faisal A Otaibi. Integrating deep learning and econometrics for stock price prediction: A comprehensive comparison of lstm, transformers, and traditional time series models. *Machine Learning with Applications*, page 100730, 2025.

[109] Georg Ostrovski, Will Dabney, and Rémi Munos. Autoregressive quantile networks for generative modeling. *ArXiv*, abs/1806.05575, 2018. URL https://api.semanticscholar.org/CorpusID:49212449.

[110] Yaniv Ovadia, Emily Fertig, Jie Jessie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's

uncertainty? evaluating predictive uncertainty under dataset shift. In *Neural Information Processing Systems*, 2019. URL https://api.semanticscholar.org/CorpusID:174803437.

[111] Efstathios Paparoditis and Dimitris N Politis. Resampling and subsampling for financial time series. In *Handbook of financial time series*, pages 983–999. Springer, 2009.

[112] Santiago Pascual, Antonio Bonafonte, and Joan Serrà. Segan: Speech enhancement generative adversarial network. In *Interspeech*, 2017. URL https://doi.org/10.48550/arXiv.1703.09452.

[113] Peter L. Pedroni. The econometric modelling of financial time series. *Journal of the American Statistical Association*, 96:339 – 355, 2001. URL https://api.semanticscholar.org/CorpusID:120871353.

[114] D. S. Poskitt and A. R. Tremayne. The selection and use of linear and bilinear time series models. *International Journal of Forecasting*, 2:101–114, 1986. URL https://api.semanticscholar.org/CorpusID:122623370.

[115] Inioluwa Deborah Raji and Jingyi Yang. About ml: Annotation and benchmarking on understanding and transparency of machine learning lifecycles. *ArXiv*, abs/1912.06166, 2019. URL https://api.semanticscholar.org/CorpusID:209370835.

[116] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021.

[117] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016. URL https://api.semanticscholar.org/CorpusID:13029170.

[118] Yaniv Romano, Evan Patterson, and Emmanuel J. Candès. Conformalized quantile regression. In *Neural Information Processing Systems*, 2019. URL https://api.semanticscholar.org/CorpusID:147703930.

[119] Kevin Roth, Aurélien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. In *Neural Information Processing Systems*, 2017. URL https://api.semanticscholar.org/CorpusID:23519254.

[120] Ayari Salah and Gatfaoui Hayette. A meta-analysis of supervised and unsupervised machine learning algorithms and their application to active portfolio management. *Expert Systems with Applications*, 271:126611, 2025.

[121] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *ArXiv*, abs/1606.03498, 2016. URL https://api.semanticscholar.org/CorpusID:1687220.

[122] Bayram Veli Salur. Investor sentiment in the stock market. 2013. URL https://api.semanticscholar.org/CorpusID:167117180.

[123] Wim Schoutens, Erwin Simons, and Jurgen Tistaert. A perfect calibration! now what? *The best of Wilmott*, page 281, 2003.

[124] David A. Schum, Gheorghe Tecuci, Dorin Marcu, and Mihai Boicu. Toward cognitive assistants for complex decision making under uncertainty. *Intell. Decis. Technol.*, 8:231–250, 2014. URL https://api.semanticscholar.org/CorpusID:10698006.

[125] Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. Financial time series forecasting with deep learning : A systematic literature review: 2005-2019. *ArXiv*, abs/1911.13288, 2019. URL https://api.semanticscholar.org/CorpusID:208513396.

[126] Maxim V Shcherbakov, Andrei Brebels, Natalia L Shcherbakova, Alexander P Tyukov, Tomas Janovsky, and Victor A Kamaev. A survey of forecast error measures. *World Applied Sciences Journal*, 24(24):171–176, 2013.

[127] Murtuza N Shergadwala, Himabindu Lakkaraju, and Krishnaram Kenthapadi. A human-centric perspective on model monitoring. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 10, pages 173–183, 2022.

[128] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. A comparison of arima and lstm in forecasting time series. *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1394–1401, 2018. URL https://api.semanticscholar.org/CorpusID:58671842.

[129] Brandon Da Silva and Sylvie Shang Shi. Towards improved generalization in financial markets with synthetic data generation. *ArXiv*, abs/1906.03232, 2019. URL https://api.semanticscholar.org/CorpusID:174801817.

[130] Justin A. Sirignano and Rama Cont. Universal features of price formation in financial markets: perspectives from deep learning. *Quantitative Finance*, 19:1449 – 1459, 2018. URL https://api.semanticscholar.org/CorpusID:54170930.

[131] Kate Smith-Miles and Simon Bowly. Generating new test instances by evolving in instance space. *Computers Operations Research*, 63:102–113, 2015. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2015.04.022. URL https://www.sciencedirect.com/science/article/pii/S0305054815001136.

[132] Kate Smith-Miles and Mario Andrés Muñoz. Instance space analysis for algorithm testing: Methodology and software tools. *ACM Computing Surveys*, 55:1–31, 03 2023. doi: 10.1145/3572895.

[133] Slawek Smyl. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, 36(1):75–85, 2020. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2019.03.017. URL https://www.sciencedirect.com/science/article/pii/S0169207019301153. M4 Competition.

[134] Antti Sorjamaa, Jin Hao, Nima Reyhani, Yongnan Ji, and Amaury Lendasse. Methodology for long-term prediction of time series. *Neurocomputing*, 70:2861–2869, 2007. URL https://api.semanticscholar.org/CorpusID:1340178.

[135] Shuntaro Takahashi, Yu Chen, and Kumiko Tanaka-Ishii. Modeling financial time-series with generative adversarial networks. *Physica A: Statistical Mechanics and its Applications*, 2019. URL https://api.semanticscholar.org/CorpusID:149615019.

[136] Shuntaro Takahashi, Yu Chen, and Kumiko Tanaka-Ishii. Modeling financial time-series with generative adversarial networks. *Physica A: Statistical Mechanics and its Applications*, 527:121261, 2019.

[137] Tomonori Takahashi and Takayuki Mizuno. Generation of synthetic financial time series by diffusion models. *Quantitative Finance*, 25(10):1507–1516, 2025.

[138] Stephen L Taylor. Modelling financial time series. 1987. URL https://api.semanticscholar.org/CorpusID:154975283.

[139] Martin Tegner and Stephen J. Roberts. A bayesian take on option pricing with gaussian processes. 2021. URL https://api.semanticscholar.org/CorpusID:53140367.

[140] Cátia Teixeira, Inês Gomes, Luís Cunha, Carlos Soares, and Jan N van Rijn. Gastenv2: Generative adversarial stress testing networks with gaussian loss. In *EPIA Conference on Artificial Intelligence*, pages 261–272. Springer, 2024.

[141] Mattias Teye, Hossein Azizpour, and Kevin Smith. Bayesian uncertainty estimation for batch normalized deep networks. In *International Conference on Machine Learning*, 2018. URL https://api.semanticscholar.org/CorpusID:51867681.

[142] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *CoRR*, abs/1511.01844, 2015. URL https://api.semanticscholar.org/CorpusID:2187805.

[143] George C. Tiao and Ruey S. Tsay. Some advances in non-linear and adaptive modelling in time-series. *Journal of Forecasting*, 13:109–131, 1994. URL https://api.semanticscholar.org/CorpusID:62653659.

[144] Richard J. Tomsett, Alun David Preece, Dave Braines, Federico Cerutti, Supriyo Chakraborty, Mani B. Srivastava, Gavin Pearson, and Lance M. Kaplan. Rapid trust calibration through interpretable and uncertainty-aware ai. *Patterns*, 1, 2020. URL https://api.semanticscholar.org/CorpusID:225588887.

[145] Howell Tong. Non-linear time series. a dynamical system approach. 1990. URL https://api.semanticscholar.org/CorpusID:40257847.

[146] Howell Tong and K. S. Lim. Threshold autoregression, limit cycles and cyclical data. *Journal of the royal statistical society series b-methodological*, 42:245–268, 1980. URL https://api.semanticscholar.org/CorpusID:38634480.

[147] United Nations. Transforming our world: The 2030 agenda for sustainable development. https://sdgs.un.org/2030agenda, 2015. [Online; accessed 21-June-2025].

[148] Joost R. van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning*, 2020. URL https://api.semanticscholar.org/CorpusID:263889552.

[149] Anne Marthe van der Bles, Sander van der Linden, Alexandra L. J. Freeman, James Mitchell, Ana Beatriz Galvão, Lisa Zaval, and David J. Spiegelhalter. Communicating uncertainty about facts, numbers and science. *Royal Society Open Science*, 6, 2019. URL https://api.semanticscholar.org/CorpusID:164948189.

[150] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. Algorithmic learning in a random world. 2005. URL https://api.semanticscholar.org/CorpusID:118783209.

[151] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, P. Zhang, and Dell Zhang. Irgan: A minimax game for unifying generative and discriminative information retrieval models. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017. URL https://doi.org/10.48550/arXiv.1705.10513.

[152] Shuo Wang, Carsten Rudolph, Surya Nepal, Marthie Grobler, and Shangyu Chen. Partgan: Privacy-preserving time-series sharing. In *International Conference on Artificial Neural Networks*, 2020. URL https://api.semanticscholar.org/CorpusID:224804844.

[153] A.S. Weigend, Taylor & Francis Group, and N.A. Gershenfeld. *Time Series Prediction: Forecasting The Future And Understanding The Past*. Proceedings Volume, Santa Fe Institute Studies in the Sciences of Complexity. Taylor & Francis Group, 2019. ISBN 9780367095055. URL https://books.google.pt/books?id=Wl7-xwEACAAJ.

[154] Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*, 2020.

[155] Magnus Wiese, Robert Knobloch, Ralf Korn, and Peter Kretschmer. Quant gans: deep generation of financial time series. *Quantitative Finance*, 20:1419 – 1440, 2019. URL https://api.semanticscholar.org/CorpusID:196831870.

[156] Magnus Wiese, Robert Knobloch, Ralf Korn, and Peter Kretschmer. Quant gans: deep generation of financial time series. *Quantitative Finance*, 20(9):1419–1440, 2020.

[157] Christopher K. I. Williams and Carl E. Rasmussen. Gaussian processes for regression. In *Neural Information Processing Systems*, 1995. URL https://api.semanticscholar.org/CorpusID:265038751.

[158] Wojciech Wisniewski, David Lindsay, and Siân Lindsay. Application of conformal prediction interval estimations to market makers' net positions. In *International Symposium on Conformal and Probabilistic Prediction with Applications*, 2020. URL https://api.semanticscholar.org/CorpusID:221706190.

[159] Tianlin Xu, Li Kevin Wenliang, Michael Munn, and Beatrice Acciaio. Cot-gan: Generating sequential data via causal optimal transport. *ArXiv*, abs/2006.08571, 2020. URL https://doi.org/10.48550/arXiv.2006.08571.

[160] Yahoo Finance. Yahoo finance - stock market live, quotes, business & finance news. https://finance.yahoo.com. Accessed: 2024-05-10.

[161] Nanyang Ye and Zhanxing Zhu. Bayesian adversarial learning. In *Neural Information Processing Systems*, 2018. URL https://api.semanticscholar.org/CorpusID:54041958.

[162] Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. In *Neural Information Processing Systems*, 2019. URL https://api.semanticscholar.org/CorpusID:202774781.

[163] Peter C. Young and Stephen Shellswell. Time series analysis, forecasting and control. *IEEE Transactions on Automatic Control*, 17:281–283, 1972. URL https://api.semanticscholar.org/CorpusID:51664364.

[164] Haozhe Zhang, Joshua Zimmerman, Dan Nettleton, and Daniel J. Nordman. Random forest prediction intervals. *The American Statistician*, 74:392 – 406, 2020. URL https://api.semanticscholar.org/CorpusID:146036890.

[165] Yunfeng Zhang, Qingzi Vera Liao, and Rachel K. E. Bellamy. Effect of confidence and explanation on accuracy and trust calibration in ai-assisted decision making. *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020. URL https://api.semanticscholar.org/CorpusID:210023849.

[166] Junbo Jake Zhao, Michaël Mathieu, and Yann LeCun. Energy-based generative adversarial network. *ArXiv*, abs/1609.03126, 2016. URL https://api.semanticscholar.org/CorpusID:15876696.

[167] Xingyu Zhou, Zhisong Pan, Guyu Hu, Siqi Tang, and Cheng Zhao. Stock market prediction on high-frequency data using generative adversarial nets. *Mathematical Problems in Engineering*, 2018(1):4907423, 2018.

[168] Eric R. Ziegel. Analysis of financial time series. *Technometrics*, 44:408 – 408, 2002. URL https://api.semanticscholar.org/CorpusID:40735270.