Master Thesis

Probabilistic LTSF: Investigating a DMS-IMS Trade-off

Kai Reffert
(1980476)

July 14, 2025

Submitted to

Data and Web Science Group
Prof. Dr. Rainer Gemulla
University of Mannheim

Abstract

This thesis addresses the underexplored intersection between point-based long-term time series forecasting (LTSF) and probabilistic forecasting. While most state-of-the-art LTSF models rely on direct multi-step (DMS) decoding for improved speed, stability and accuracy, they often ignore temporal dependencies across predicted time steps, limiting their ability to generate coherent probabilistic forecasts. We extend several leading point LTSF models with probabilistic forecasting techniques, including distributional forecasting, quantile regression and implicit quantile networks, and evaluate their performance across standard LTSF tasks. A key contribution is the introduction of *multi-world* scenarios, which illustrate how the same historical context can plausibly lead to multiple divergent futures. In these settings, we demonstrate that while DMS-based probabilistic models yield accurate univariate distributions, they struggle to produce consistent sample trajectories. In contrast, iterative multi-step (IMS) models better preserve temporal coherence due to their conditional dependency structure. Our findings highlight a trade-off between the robustness and realism of uncertainty quantification in probabilistic LTSF and advocate for a re-evaluation of decoding strategies. All models and experiments are integrated into the open-source BasicTS+ framework to promote reproducibility and future research.

Contents

1. Introduction

Time series analysis encompasses a variety of downstream tasks such as forecasting, classification, and anomaly detection. Among these, time series forecasting (TSF), i.e. forecasting the future of a time series given its historical values, is one of the most important tasks with wide applicability across critical domains like epidemiology (Wu et al. 2018; Tapak et al. 2019), traffic modeling (Jayanthi and Jothilakshmi 2021; Raeesi et al. 2014), energy demand prediction (Wang et al. 2018), wind forecasting (Xie et al. 2023), and stock price prediction (Mehtab and Sen 2020). For instance, in epidemiology, TSF of infectious diseases enables public health authorities to anticipate outbreaks, allocate resources efficiently and implement timely interventions to mitigate disease spread (Tapak et al. 2019). Accordingly, TSF is essential across these fields because it directly improves decision-making, risk mitigation and efficient resource allocation (Gneiting and Katzfuss 2014). Furthermore, similar to other domains, TSF research encountered a paradigm shift from classical statistical models to deep learning architectures (Benidis et al. 2022; Lim and Zohren 2021). Moreover, two major branches, point-based long-term TSF (LTSF) and probabilistic short-term TSF (STSF), have emerged in this rapidly evolving field (Zhang et al. 2024).

Until Zhou et al. (2021) proposed the LTSF task, i.e. learning from large amounts of data to forecast multiple hundred steps into the future, TSF research primarily considered shorter forecasting horizons, e.g. only predicting dozens of time steps (Cirstea et al. 2022). However, short-term forecasts often fail to provide important insights needed for long-term planning, whereas long-range forecasting helps to capture future trends over extended periods, supporting more informed long-term decision-making (Jia et al. 2024). Increasing recognition for LTSF has driven the development of various specialized models, including DLinear (Zeng et al. 2023), PatchTST (Nie et al. 2022), iTransformer (Liu et al. 2023), FITS (Xu et al. 2023) and PatchMixer (Gong et al. 2024) among many more (see Table 3.1). A growing trend among these models is the adoption of a direct multistep (DMS) (or non-autoregressive) decoding strategy, in which the whole horizon is predicted at once. A primary reason for the dominance of DMS approaches is that iterative multistep (IMS) (or autoregressive) models, which predict the whole horizon by recursively producing one-step predictions, tend to run into error accumulation problems, where previous errors influence the next predictions as well. Nevertheless, a major limitation of current LTSF methods is that they have been predominantly evaluated based on their point forecasting performance while largely overlooking probabilistic forecasting, i.e. estimating the joint distribution over future values of a time series (Rasul et al. 2023; Zhang et al. 2024).

Probabilistic forecasts quantify the uncertainty of forecasts, making them an essential ingredient for robust decision-making and risk management (Gneiting and Katzfuss 2014). Consequently, many research efforts have focused on developing probabilistic TSF methods, for instance DeepAR (Salinas et al. 2020), C2FAR (Bergsma et al. 2022), DSDP-GP (Biloš et al. 2023) or RATD (Liu et al. 2024) including many more (see Table 3.2). However, the primary concern of these works is to capture complex data distributions in STSF scenarios. Accordingly, both IMS and DMS strategies are adopted equally, since error accumulation is less pronounced when forecasting over shorter horizons. Altogether, integrating point LTSF and probabilistic TSF approaches is still an open challenge, as noted by Zhang et al. (2024):

> "Notably, while recent probabilistic forecasting approaches have shown proficiency in short-term distribution estimation, we find that long-term distributional forecasting remains a significant challenge." (Zhang et al. 2024)

Therefore, one dimension of this thesis aims to bridge the gap between these two parallel lines of research. To this end, we extend several state-of-the-art point LTSF models with probabilistic forecasting techniques, including distributional forecasting, quantile regression and implicit quantile networks (IQN) (Dabney et al. 2018; Gouttes et al. 2021).

However, while transitioning from point LTSF models to probabilistic LTSF models requires only minimal architectural modifications, largely preserving the underlying backbone, the adoption of the predominant DMS strategy produces conditionally independent predictions (Taieb and Hyndman 2012). This design limits their ability to capture probabilistic temporal dependencies across forecasted time steps, resulting in less coherent and less realistic uncertainty quantification in the generated forecasts. In contrast, IMS models explicitly capture sequential dependencies by conditioning each prediction on the preceding predicted value, thereby producing predictions that are conditionally dependent across time steps. To better understand this underexplored limitation, we introduce the notion of *multi-world* scenarios, in which a given historical prefix can lead to multiple divergent future trajectories that are indistinguishable from the past alone. In such settings, DMS models fail to generate coherent sample forecasts due to their inability to model joint temporal dependencies. Although one potential solution is to use multivariate distributions to explicitly capture these dependencies, this approach is often computationally intensive and prone to instability (Salinas et al. 2019; Wu et al. 2013). By contrast, we demonstrate that IMS models naturally generate sample trajectories that more faithfully reflect the underlying stochastic process, resulting in more coherent and realistic forecasts. In summary, the key contributions of this work are:

- A comprehensive literature study and taxonomy of point LTSF, probabilistic TSF and associated probabilistic evaluation metrics.
- Probabilistic implementations of state-of-the-art point LTSF models and their probabilistic evaluation on common LTSF tasks.
- A detailed empirical study via *single-* and *multi-world* simulations, highlighting the limitations of DMS models in certain probabilistic settings.

- Integration of all models and experiments into the public benchmark framework BasicTS+[1] (Shao et al. 2025), ensuring reproducibility and facilitating future research. Our code is available at `https://github.com/Kai-Ref/Probabilistic_LTSF/`.

Chapter 2 introduces the mathematical formulation of the TSF problem and distinguishes between modeling paradigms, such as global versus local models and direct versus iterative strategies. Chapter 3 presents a structured review of existing work on LTSF, probabilistic TSF, and the evaluation of probabilistic forecasts. In Chapter 4, we detail our proposed methodology, including model adaptations and extensions for probabilistic LTSF. Chapter 5 covers the experimental design, empirical results, and discussions. Finally, Chapter 6 summarizes the findings and suggests directions for future work.

[1] `https://github.com/GestaltCogTeam/BasicTS/`

2. Preliminaries: TSF Problem Formulation

This chapter formalizes the task of TSF by introducing the mathematical framework and clarifying important modeling distinctions. Hereby, we closely follow the notation and taxonomy proposed by Benidis et al. (2022). To avoid confusion with multivariate probability distributions, we refer to multivariate time series as multi-channel time series. Each multi-channel time series consists of values $x_t^{(i)} \in \mathcal{X}$, where $i \in 1, 2, \ldots, N$ indexes the individual channels, i.e. univariate series[1], and t denotes the time index. Moreover, the domain $\mathcal{X}$ of time series values is typically either $\mathbb{N}$, $\mathbb{Z}$, $[0, 1]$ or $\mathbb{R}$. The vector of a single time step will be denoted as $x_t \in \mathcal{X}^N$. In the multi-channel TSF setup, the input consists of a group of such vectors with look-back window length L, i.e. x_1, $\ldots$, $x_L = x_{1:L} \in \mathcal{X}^{N \times L}$, and the aim is to forecast the future values x_{L+1}, $\ldots$, $x_{L+H} = x_{L+1:L+H} \in \mathcal{X}^{N \times H}$ over a forecasting horizon H. Subsequently, the general objective is to learn the following conditional distribution

$$p(x_{L+1:L+H} | x_{1:L},\ z_{1:L+H};\ \theta) \tag{2.1}$$

where $z_{1:L+H}$ represents known covariates available throughout the entire input and forecast window and θ denotes the parameters of a model (Benidis et al. 2022). According to Benidis et al. (2022), Equation 2.1 can be expressed through three principal modeling strategies. First, are local single-channel, i.e. univariate, models, which treat each of the N single-channel time series independently and forecast a single channel. That is, a separate model is trained *locally* for each series without parameter sharing, leading to the decomposition:

$$p(x_{L+1:L+H}^{(i)} | x_{1:L}^{(i)},\ z_{1:L+H}^{(i)};\ \theta^{(i)}), \quad \theta^{(i)} = \Psi(x_{1:t}^{(i)},\ z_{1:L+H}^{(i)}) \tag{2.2}$$

Here, Ψ is a function that determines the parameters $\theta^{(i)}$ of the probabilistic model, which are specific to series i. Since models are fit independently, information is not shared across series, limiting generalization and making this approach unsuited to cold-start scenarios, i.e. forecasting an unseen series without its historical information (Benidis et al. 2022; Montero-Manso and Hyndman 2021). This framework is primarily prevalent in classical statistical methods, e.g. exponential smoothing (Hyndman et al. 2002) or ARIMA (Hyndman and Khandakar 2008), and early neural networks (Zhang et al. 1998). Second, global single-channel models (Januschowski et al. 2020) share parameters *globally* across all series. A single model is trained using all available time series but makes forecasts for each individual channel series separately:

$$p(x_{L+1:L+H}^{(i)} | x_{1:L},\ z_{1:L+H};\ \theta^{(i)}), \quad \theta^{(i)} = \Psi(x_{1:t}^{(i)},\ z_{1:L+H}^{(i)},\ \Phi) \tag{2.3}$$

[1]The term channel is sometimes used interchangeably with time series, where one channel corresponds to a single univariate time series.

where Φ represents the global learned parameters shared across all series, enabling cross-learning (Semenoglou et al. 2021). The function Ψ, often a neural network, maps individual series histories and covariates to predictive parameters. Although the model learns Φ based on all series, during inference each time series is predicted independently, i.e. without looking at other time series. Lastly, global multi-channel models go a step further by additionally forecasting multiple channels jointly with a single model as well. They consider dependencies between time series during both training and inference:

$$p(x_{L+1:L+H}|x_{1:L},\ z_{1:L+H};\ \theta), \quad \theta = \Psi(x_{1:t},\ z_{1:L+H},\ \Phi) \tag{2.4}$$

While there are also a few examples of local multi-channel models, e.g. VARMA (Lütkepohl 2005), they are certainly not as widespread as the other modeling strategies (Benidis et al. 2022).

Thus far, we have described probabilistic forecasters that predict full distributions. In some applications, however, only point forecasts, e.g. the mean, median, or a specific quantile, are desired. These point forecasts can be derived from the learned distribution or trained directly via different metrics (Kolassa 2020). For instance, a point-forecast global single-channel model can learn to forecast the mean as $\hat{x}^{(i)}_{L+1:L+H} = \Psi(x^{(i)}_{1:L},\ z^{(i)}_{1:L+H},\ \Phi)$ by using the mean squared error between the predictions $\hat{x}^{(i)}_{L+1:L+H}$ and the ground truth $x^{(i)}_{L+1:L+H}$. Furthermore, in the discussion so far, our formulations assume direct prediction of the entire horizon for $H > 1$, i.e. a DMS strategy (Chevillon 2007). Nevertheless, IMS forecasters (Taieb and Hyndman 2012), which recursively generate single-step forecasts one step at a time, are seamlessly applicable to the presented multi-step prediction scenario by producing iterating forecasts for the predictive horizon H (Benidis et al. 2022). Following Graves (2014), for IMS methods, the conditional distribution of the overall goal in Equation 2.1 can be expressed as:

$$p(x_{L+1:L+H}|x_{1:L},\ z_{1:L+H};\ \theta) = \prod_{t=L+1}^{L+H} p(x_t|x_{1:t-1},\ z_{1:L+H};\ \theta_t) \tag{2.5}$$

Note that, Equations 2.2, 2.3, 2.4 can be decomposed equivalently. Here, during training, teacher forcing (Williams and Zipser 1989) is usually used to condition the model on ground-truth data $x_{1:t-1}$. Whereas, during inference, the value of a previous time step x_{t-1} is replaced by a model-generated sample $\hat{x}_{t-1}$ if $t \geq L+2$ (Salinas et al. 2020). Additionally, the model parameters θ_t are often time-dependent and θ collectively denotes the parameters for all time steps t. While IMS predictions have smaller variance than DMS counterparts, they suffer from error accumulation effects, which are especially influential for larger H values (Zeng et al. 2023). On the contrary, DMS models are initialized with a fixed prediction horizon length H *a priori*, whereas IMS models can provide forecasts of arbitrary length (Bergsma et al. 2023).

In recent times, another dimension to classify TSF models gained increasing research attention: the distinction between channel-independent (CI) and channel-dependent (CD)

models (Nie et al. 2022; Zeng et al. 2023; Han et al. 2024). CI models treat each channel separately, modeling them in isolation without explicitly capturing inter-channel dependencies (Han et al. 2024). By contrast, CD models explicitly incorporate information from multiple other target channels, enabling them to learn cross-series dependencies in a direct way. In theory, all previously discussed model categories could adopt either strategy (CI or CD), as the distinction between them solely relates to whether cross-channel dependencies are explicitly modeled. Hence, their division is independent of whether a model is trained globally or locally, or whether it forecasts single or multiple variables. For example, a global single-channel CD model can explicitly incorporate inter-series dependencies while still producing forecasts for a single channel. Conversely, a multi-channel CI model could, in principle, generate forecasts for multiple series simultaneously without directly modeling any interactions between them. In practice, global single-channel models are the most common implementation of the CI strategy (Nie et al. 2022; Lin et al. 2024; Challu et al. 2023), since they are able to share some statistical power and learned patterns across series without explicit modeling through parameter sharing (Rangapuram et al. 2018). Although any of the earlier model types can be implemented in a CD manner as well, this strategy is predominantly adopted by global single- and multi-channel models (Zhou et al. 2021; Han et al. 2024; Huang et al. 2024). In the next chapter, we will look into related work.

3. Literature Review

This chapter reviews key works that have influenced our research. We begin by covering broader developments in TSF, followed by recent advances in long-term forecasting. We then examine related probabilistic TSF approaches. Finally, we discuss the probabilistic evaluation metrics used to train probabilistic TSF models and to assess their forecast quality.

3.1. Related Work on (L)TSF

TSF has a long and extensive literature history. However, this work will primarily focus on recent developments in (long-term) TSF. For a comprehensive overview of earlier research and traditional TSF methods, readers are referred to existing surveys such as Box (2013); Box et al. (2015); De Gooijer and Hyndman (2006); Mahalakshmi et al. (2016); Hamilton (1994). Some traditional statistical time series forecasting models, such as ARIMA (Box and Pierce 1970) or Prophet (Taylor and Letham 2018), are still popular to this day (Long et al. 2023; Ning et al. 2022; Albahli 2025). However, they are often fit separately to each time series, come with many prior assumptions and their performances may deteriorate for long-range forecasting, making them unsuitable for large scale TSF tasks (Qin et al. 2017; Li et al. 2019). Therefore, similar to other domains, TSF research showed an increasingly large interest towards deep learning based approaches (Benidis et al. 2022; Hewamalage et al. 2021; Lara-Benítez et al. 2021).

At first, primarily recurrent neural networks (RNNs), which are specifically designed to work with sequential data, were adopted in the form of sequence-to-sequence architectures (Sutskever et al. 2014). Furthermore, many SOTA performances across TSF tasks with shorter forecasting horizons stem from models of this architecture, e.g. TimeGrad (Rasul et al. 2021), DA-RNN (Qin et al. 2017) or DeepAR (Salinas et al. 2020). In contrast, Convolutional neural networks (CNNs), which are designed for tasks where the input data has a known sequential or spatial structure, such as images or audio signals (Dosovitskiy et al. 2021; Van den Oord et al. 2016) but also time series (Benidis et al. 2022; Goodfellow et al. 2016), began to demonstrate superior performance over RNNs in various sequence modeling tasks, e.g. audio generation or machine translation (Van den Oord et al. 2016; Kalchbrenner et al. 2017). Motivated by these results, Bai et al. (2018) conducted a comprehensive comparison between CNNs and RNNs across a diverse set of sequential learning benchmarks. Their findings showed that a simple convolutional architecture, the Temporal Convolutional Network (TCN), consistently outperformed RNN-based models while also benefiting from longer effective memory. These promising results spurred increased interest in applying CNNs to time series forecasting as well. For

instance, Borovykh et al. (2018) adapted the autoregressive WaveNet CNN architecture (Van den Oord et al. 2016), originally developed for raw audio synthesis, to the TSF domain and demonstrated superior performance over LSTM-based models. DeepGLO (Sen et al. 2019) combines a matrix factorization model with a TCN and outperforms traditional and RNN-based methods. However, despite their promising empirical performance, CNNs have not emerged as a definitive replacement for RNNs. Instead, the two architectures were generally viewed as complementary, with approximation theory (Jiang et al. 2021) supporting the idea that each bring distinct strengths to time series modeling. Therefore, hybrid models like LSTNet (Lai et al. 2018) or DCRNN (Li et al. 2018) gained popularity by combining CNNs and RNNs, effectively capturing both short-term dependencies and inter-series correlations through CNNs, while leveraging RNNs for modeling longer-term temporal trends. Nevertheless, both RNNs and CNNs exhibit inherent limitations when it comes to longer forecasting horizons. The main limitation of RNNs are their large information propagation paths, which directly lead to numerous issues. In particular, RNNs have performance problems in capturing long-term dependencies with poor efficiency in sequential calculations (Jia et al. 2024). Furthermore, although RNN-cells, such as LSTM (Hochreiter and Schmidhuber 1997) or GRU (Cho et al. 2014), were designed to tackle vanishing and exploding gradients (Bengio et al. 1994), those problems often could not be mitigated sufficiently for longer input sequences leading to an unstable training process (Zhou et al. 2021). On the other hand, CNNs are limited by their local receptive fields; while some argue that they offer better long-term memory than RNNs (Bai et al. 2018), their 1D convolutions can only model variations in adjacent time steps (Wu et al. 2022). Therefore, compared to models with global receptive fields, e.g. Transformers (Vaswani et al. 2017) or MLP-based architectures (Zeng et al. 2023), CNNs often fall short in handling the complexity of long-term temporal dependencies (Donghao and Xue 2023). Altogether, these limitations are critical in TSF tasks, which often require models to capture both short- and long-term repeating patterns (Lai et al. 2018). In the context of long-term TSF, the importance of modeling long-range dependencies becomes even more pronounced, as they tend to be more dispersed and harder to learn (Li et al. 2019).

In response to these challenges, Transformer-based models (Vaswani et al. 2017) were proposed as a promising alternative (Zhou et al. 2021; Li et al. 2019), offering a self-attention mechanism, which allows the model to access the entire input sequence at once, facilitating parallel processing and enabling global context understanding. Furthermore, Transformers have displayed state-of-the-art performances in capturing long-range dependency structures (Wen et al. 2023) and are SOTA across various domains, e.g. natural language processing (Brown et al. 2020), speech (Kim et al. 2022) and computer vision (Dosovitskiy et al. 2021). LogSparse, proposed by Li et al. (2019), was among the first Transformer-based methods applied to TSF. It demonstrated superior performance in modeling long-term dependencies compared to DeepAR and statistical models. Although Li et al. (2019) extended the forecasting horizon relative to earlier work, the input and output sequences they considered were still shorter compared to modern LTSF tasks. A breakthrough came when Zhou et al. (2021) introduced Informer and formalized the

modern LTSF problem setting by substantially extending input and prediction horizons. Informer managed to outperform prior SOTA models including LogSparse, DeepAR, other RNN-based and statistical baselines in LTSF. A key innovation of Informer came with its switch to a DMS strategy (Zeng et al. 2023), which contrasts the IMS approach used in earlier methods. Moreover, models that follow an IMS strategy are prone to slow inference and error accumulation, issues that become particularly problematic with longer forecast lengths (Zhou et al. 2021).

In succession, the DMS strategy was successfully adopted by most SOTA LTSF models, see Table 3.1. However, DMS forecasting is not novel. In fact, the first occurance of a DMS prediction model can be dated back to Cox (1961). Over the years several theoretical and empirical studies have shown that the direct strategy performs better when models are misspecified, i.e. the model class does not contain the true model, while the recursive approach tends to be superior for well-specified models (Weiss 1991; Tiao and Tsay 1994; Ing 2007; Chevillon and Hendry 2005). In summary, Chevillon (2007) showed that DMS is less biased, more stable, more efficient and more robust to model misspecification. Later on, Taieb and Atiya (2016) investigated different multi-step strategies with NNs in TSF and concluded that IMS is preferable for short-term forecasts when the model is likely well-specified, whereas DMS is better suited for long time series or situations where minimizing bias is crucial. Despite these findings the IMS strategy was still more popular around that time, part of the reason is that it is highly similar to well-studied autoregressive and Markovian modeling assumptions while benefiting from shorter forecasting horizons as well (Wen et al. 2018). Moreover, DMS was regarded as costly, since, without cross-learning, it required training separate models for each horizon step (Bontempi et al. 2013). However, this drawback became negligible with newer architectures efficiently sharing parameters across time steps, for example only requiring small changes in the prediction head while enabling faster prediction speeds (Zhou et al. 2021). Prior to Informer, other deep learning models also adopted DMS strategies. For instance, MQ-RNN and MQ-CNN (Wen et al. 2018) use shared-parameter decoders at each time step to produce forecasts. Building on MQ-CNN, Wen and Torkkola (2019) added a generative quantile copula improving the forecast quality. NBeats (Oreshkin et al. 2019) is built on a deep residual stack of MLPs, whereas DeepTCN (Chen et al. 2019) is a CNN-based DMS approach. Nonetheless, the DMS strategy has important drawbacks: it treats the forecasted points as independent, overlooking their mutual dependencies (Kline 2004; Bontempi et al. 2013) and it must be retrained whenever the forecast horizon is extended.

The breaktrough of Informer led to a rising adoption of LTSF models, specifically Transformer models. However, despite their advantages, the memory and time complexity of self-attention in Transformers grows quadratically $O(L^2)$ with the input length L, becoming a large bottleneck for long input sequences present in LTSF (Zhou et al. 2021). Hence, many of the first Transformer-based models for LTSF focused on improving the efficiency of the attention module, in which Wen et al. (2023) classify the approaches into two branches. On the one hand, models such as LogSparse (Li et al. 2019) or Pyraformer

(Liu et al. 2021) tried enforcing a sparsity bias into the attention module. On the other hand, Informer (Zhou et al. 2021) or FEDformer (Zhou et al. 2022) analyzed low-rank properties of the self-attention matrix. Furthermore, in their respective LTSF studies each model manages to outperform previous traditional and RNN-based SOTA methods, such as ARIMA, Prophet or DeepAR on a variety of LTSF data sets (Zhou et al. 2021; Wu et al. 2021; Liu et al. 2021; Li et al. 2019; Zhou et al. 2022). Despite their performances, Zeng et al. (2023) point out that they were evaluated solely against IMS approaches and suggest that the observed improvement is primarily due to the adoption of the DMS strategy rather than the Transformer architecture itself. To investigate this, Zeng et al. (2023) introduce DLinear and NLinear, two simple linear MLP DMS models, which were able to outperform the Transformer-based methods on multiple different benchmarks. Thus, challenging the effectiveness of Transformers on LTSF tasks. An important aspect of DLinear and NLinear is that they are CI methods, therefore they mitigate from modeling potentially misleading cross-channel dependencies (Nie et al. 2022). In contrast, many previous methods (Zhou et al. 2021; Wu et al. 2021; Zhou et al. 2022) tried to incorporate information from all channels via a CD strategy, but this approach appeared to be ineffective in comparison. Building on the success of Zeng et al. (2023) with the CI strategy, many LTSF models adopted it successfully, see Table 3.1. Furthermore, Han et al. (2024) investigate the relation between CI and CD methods more in-depth. By comparing a linear CI model to its CD counterpart, they propose that the CI approach exhibits less distribution shift, because the sum of correlation differences between train and test data has lower variation than the correlation differences of individual channels. Subsequently, Han et al. (2024) propose that CD methods have high capacity and low robustness, whereas CI approaches have low capacity and high robustness. Lastly, coming to the conclusion that robustness is often more important in real-world non-stationary time series with distribution shifts; therefore, CI methods often perform better.

Since the work of Zeng et al. (2023) challenged the effectiveness of Transformers in LTSF, this opened the door for other architectures to gain back some ground. Furthermore, in what follows some important recent contributions of LTSF models are briefly described, for a detailed categorization of these models see Table 3.1.

MLP architectures. The success of DLinear (Zeng et al. 2023) revived interest in pure MLP architectures for LTSF. At the same time, the computer vision community saw the rise of MLP-Mixer models (Tolstikhin et al. 2021; Liu et al. 2021; Touvron et al. 2023), which use simple MLPs to mix information within and across image input patches, achieving competitive results without relying on convolutions or self-attention. Building on this, TSMixer (Ekambaram et al. 2023) adapts the Mixer architecture for LTSF, leveraging its well-suited compatibility with sequential data due to the preservation of input order. TSMixer uses a patch-based MLP backbone enhanced with online reconciliation heads that capture hierarchical structure and cross-channel dependencies. Following TSMixer, several studies extended the idea to address specific challenges in time series

modeling. TimeMixer (Wang et al. 2023) leverages multiscale-mixing, differentiating finer seasonal patterns and coarser trends through novel mixing blocks. U-Mixer (Ma et al. 2024) tackles the issue of non-stationarity by arranging MLP encoder-decoder blocks in a U-Net structure (Ronneberger et al. 2015) while also introducing a stationarity correction mechanism. Furthermore, HDMixer (Huang et al. 2024) improves fixed-sized patching via length-extendable patching while also modeling hierarchical short- and long-range dynamics. Beyond Mixer-based architectures, a range of MLP-centric models have emerged that take alternative approaches to enhancing time series forecasting performance. NHITS (Challu et al. 2023) extends NBEATS (Oreshkin et al. 2019) by introducing hierarchical interpolation and multi-rate sampling to sequentially assemble forecasts across multiple temporal resolutions. FreTS (Yi et al. 2023) operates entirely in the frequency domain, using MLPs to learn real and imaginary components of transformed series. CycleNet (Lin et al. 2024) leverages residual cycle forecasting to explicitly model periodic components. SOFTS (Han et al. 2024) proposes a centralized STAR module to model inter-channel relationships more efficiently than attention mechanisms. Finally, TiDE (Das et al. 2023) employs a simple MLP-based encoder-decoder framework that combines the speed of linear models with the ability to capture nonlinear dependencies.

Transformers. Despite the success of MLP-based approaches, Transformers remained the popular choice for LTSF tasks, see Table 3.1. One reason for this was the introduction of PatchTST (Nie et al. 2022), which marked a turning point for Transformer-based models in time series forecasting. It adopts the CI strategy of DLinear (Zeng et al. 2023) while also introducing patching to TSF. Patching, inspired by Vision Transformers (Dosovitskiy et al. 2021), segments a time series into subseries-level patches. It allows the model to capture local semantic patterns, reduce attention complexity, and extend its receptive field, significantly boosting long-term forecasting accuracy (Nie et al. 2022). As a result, patching has since become a standard practice in time series Transformers, widely adopted in models like Crossformer (Zhang and Yan 2022), MCFormer (Han et al. 2024) and Pathformer (Chen et al. 2023). In addition to its success in Transformer-based models, patching has been adopted across other architectural families, including MLPs (Chen et al. 2023), CNNs (Gong et al. 2024), and RNNs (Lin et al. 2023). However, the dominance of classic fixed-length patching has recently been challenged. The MLP-based HDMixer (Huang et al. 2024) critiques the inflexibility of fixed-length patches, which can lead to information loss at the patch boundaries. It proposes length-extendable patches to better preserve local structure. In addition, DeformableTST (Luo and Wang 2024) highlights that modern Transformers have become overly reliant on patching to achieve strong performance, which limits their applicability in scenarios with short input sequences or tasks unsuited to patching. To address this, DeformableTST introduces deformable attention, a data-driven sparse attention mechanism capable of focusing on important time points without explicit patching, allowing the model to generalize across a broader range of forecasting tasks. Lastly, several works have sought alternatives to patching through other input transformations. Fredformer (Piao et al. 2024) applies

a Discrete Fourier Transform to overcome frequency bias in attention, enabling more balanced learning across frequency bands. iTransformer (Liu et al. 2023) takes a different route by inverting the input dimensions, treating time points as tokens and leveraging attention to capture multivariate correlations, improving scalability and performance without altering the Transformer's core components.
Similar to patching, the standard Transformer encoder (Vaswani et al. 2017) has become a standard modeling choice for Transformer-based time series models. In many cases, the decoder is simply replaced with a basic flatten and linear head, e.g. MCFormer (Han et al. 2024), PatchTST (Nie et al. 2022), iTransformer (Liu et al. 2023) and Fredformer (Piao et al. 2024). On top of that, many models make targeted replacements to the vanilla Transformer encoder, where it is common to make changes to the attention mechanism: Triformer (Cirstea et al. 2022) reduces complexity via triangular patch attention, SDformer (Zhou et al. 2024) enhances expressiveness with spectral filtering and dynamic directional attention, SCAT (Zhou et al. 2024) introduces alternating attention using spectral clustering centers and CARD (Wang et al. 2023) aligns attention across channels to better model inter-channel dependencies. Similarly, CATS (Lu et al. 2024) removes self-attention altogether, opting for a cross-attention-only framework. To better capture long-range dependencies, Kang et al. (2024) introduce spectral attention, a frequency-based mechanism that preserves temporal patterns and improves gradient flow. Outside of encoder-only Transformer models, a few different architectures have been implemented as well. SMARTformer (Li et al. 2023) adopts a full encoder-decoder Transformer architecture, but deviates from the standard non-autoregressive decoder commonly used in time series models. Crossformer (Zhang and Yan 2022) also uses an encoder-decoder architecture, but places special emphasis on modeling cross-dimension dependencies. To this end, it proposes a Two-Stage Attention mechanism within a hierarchical encoder-decoder structure that separately captures temporal and inter-variable correlations. In contrast, FPPformer (Shen et al. 2024) also retains the encoder-decoder setup but focuses on redesigning the decoder. It introduces a top-down decoder architecture, inspired by feature pyramid networks in computer vision (Lin et al. 2017), and enhances it with a combination of elementwise and patchwise attention to improve multiscale sequence reconstruction.

CNNs. While Transformer- and MLP-based models have rapidly gained traction and became dominant in time series analysis, convolutional approaches have been falling out of favor (Donghao and Xue 2023). Nevertheless, several recent studies have achieved SOTA performance in LTSF using CNN-based models, renewing interest in convolutional methods. MICN (Wang et al. 2022) introduces a multi-scale convolutional architecture that captures both local features and global correlations, enabling separate modeling of trend and seasonality in time series forecasting. TimesNet (Wu et al. 2022) leverages the Fast Fourier Transform (FFT) to identify periodic patterns in time series data, which it then restructures into 2D tensors. Its core component, the TimesBlock, is built based on a convolutional inception block (Szegedy et al. 2015), enabling it to effectively model both inter-period and intra-period variations. PatchMixer (Gong et al. 2024) and

ModernTCN (Donghao and Xue 2023) process time series in patches (Nie et al. 2022) and then utilize depthwise separable convolutions to achieve SOTA performance with faster training and inference speeds. Moreover, ModernTCN extends a convolution block better suited for time series, resulting in larger effective receptive fields.

RNNs. Despite their limitations and general subpar performance in LTSF, RNNs occasionally resurfaced in LTSF research. Lin et al. (2023) identify the large number of recurrent iterations as a primary drawback of traditional RNNs. To address this, they propose SegRNN, which adopts a patching mechanism to reduce the number of recurrent steps when processing input time series. In addition, they employ a DMS strategy for prediction. This involves incorporating positional embeddings, as in Vaswani et al. (2017), which are combined with the last hidden state and then passed into a GRU cell with shared parameters. Jia et al. (2023) introduced WITRAN, which operates on rearranged 2D time series, i.e. a matrix of patches inspired by Wu et al. (2022). Then, they propose a novel RNN cell alongside the recurrent acceleration network, which processes the data points of the matrix vertically and horizontally, enabling parallel computation. Lastly, they decode the processed information with a MLP in a DMS fashion. Similarly, Jia et al. (2024) introduce TPGN, a dual-branch model that also uses a 2D representation to capture long- and short-term patterns. At its core is the Parallel Gated Network, which replaces the sequential structure of RNNs with a layer that aggregates information from previous time steps in parallel, reducing the propagation path to $O(1)$.

Other model types. Beyond common model archetypes, LTSF has recently seen novel architectures inspired by other domains. LLM-based models like LeRet (Huang et al. 2024), AutoTimes (Liu et al. 2024) or Time-LLM (Jin et al. 2023) leverage pre-trained language models by aligning time series with token-based representations, enabling few-shot and in-context forecasting. Graph-based models such as Ada-MSHyper (Shang et al. 2024), CrossGNN (Huang et al. 2023) and MSGNet (Cai et al. 2024) introduce graph structures to better capture multi-scale or inter-series correlations. Lastly, dynamical system-based approaches like Koopa (Liu et al. 2023) and Attraos (Hu et al. 2024) leverage Koopman embeddings to linearize complex dynamics or draw on chaos theory, respectively.

In addition to exploring different backbone NN architectures, studies have also examined the impact of other design choices.

Sparse models. Despite the success of Zeng et al. (2023) with simple linear models, many of the previously discussed methods rely on significantly larger approaches with a high amount of parameters. To counter the trend toward increasingly large models, some methods focused on more efficient sparser models that often only implement one or a few linear layers. For instance, FITS (Xu et al. 2023), LightTS (Zhang et al. 2022), SSCNN (Deng et al. 2024), Attraos (Hu et al. 2024) and SparseTSF (Lin et al. 2024) achieve performances comparable to SOTA methods while being several magnitudes

smaller, resulting in faster training and inference speeds as well as a smaller memory footprint. These models first simplify the forecasting task by downsampling (Lin et al. 2024; Zhang et al. 2022), by decomposition (Deng et al. 2024) or by operating in the frequency domain via FFT (Xu et al. 2023) or via phase space reconstruction (Hu et al. 2024). Then, they process the condensed representation with a smaller model, often containing only a single (non-) linear layer.

Channel dependence. The success of DLinear (Zeng et al. 2023) and PatchTST (Nie et al. 2022) with the CI strategy led to many subsequent CI models such as Pathformer (Chen et al. 2023), CATS (Kim et al. 2024) and DeformableTST (Luo and Wang 2024). However, growing interest in leveraging inter-series correlations developed into a resurgence of CD methods, which can be broadly categorized by their mechanism of capturing cross-channel interactions. A large subset utilizes cross-channel attention, with models like Crossformer (Zhang and Yan 2022), CARD (Wang et al. 2023), Client (Gao et al. 2023) and MCformer (Han et al. 2024) incorporating attention modules to jointly model temporal and inter-channel dependencies. Another line of work applies spectral or frequency-based modeling, such as SDformer (Chen et al. 2024), Fredformer (Piao et al. 2024) and FreTS (Yi et al. 2023), which leverage frequency-domain representations to capture global dependencies and improve channel interaction modeling. Meanwhile, MLP-Mixer-based architectures offer an alternative to attention-heavy designs. For instance, TSMixer (Ekambaram et al. 2023) introduces hybrid channel modeling, while SOFTS (Han et al. 2024) similarly proposes a centralized STAR module to fuse global and intra-channel representations. On another note, models like TimeMixer (Wang et al. 2023), ModernTCN (Donghao and Xue 2023) and MICN (Wang et al. 2022) explore multi-scale decomposition and convolutional modeling to disentangle and aggregate information across variables and temporal resolutions. Lastly, CrossGNN (Huang et al. 2023) applies graph-based modules to model cross-variable structure.

DMS dominance. Although Li et al. (2019) were among the first to apply Transformers to LTSF in an IMS setting, nearly all major recent models for LTSF adopt a DMS forecasting strategy, see Table 3.1. This trend can be traced back to Informer (Zhou et al. 2021), which popularized the use of non-autoregressive decoding to mitigate error accumulation in long-range predictions of IMS methods, as mathematically shown by Sun and Boning (2022). Even recurrent architectures, which are closely related to IMS forecasting, have adopted a DMS strategy for LTSF (Lin et al. 2023; Jia et al. 2023). Two recent works stand out as rare exceptions that reintroduce autoregressive principles into LTSF. SMARTformer (Li et al. 2023) proposes a semi-autoregressive (SAR) decoding approach, consisting of two key components: a segment autoregressive layer that generates the forecast iteratively in segments, and a non-autoregressive refining layer that globally refines the output in a DMS manner. This hybrid structure captures both local and global temporal patterns. Empirical results show that SMARTformer achieves consistent improvements in both univariate and multivariate forecasting tasks while an ablation study highlights that other SOTA LTSF methods also benefit from

a SAR decoder. On the other hand, AutoTimes (Liu et al. 2024) leverages the autoregressive nature of LLMs to forecast time series through token-wise next-step prediction. However, its main novelty lies in repurposing decoder-only LLMs for time series.

Table 3.1.: Summary of point forecasting models for LTSF. Each model is categorized by venue, decoding strategy (IMS/DMS), backbone architecture (including Transformer variations: Encoder-only (E), Decoder-only (D), Encoder-Decoder (E-D)) and CI/CD strategy. LLMs and hybrid models are noted where applicable.

Model	**Venue**	**IMS/DMS**	**Backbone**	**CI/CD**
LogSparse (Li et al. 2019)	NeurIPS'19	IMS	Transformer (D)	CD
Autoformer (Wu et al. 2021)	NeurIPS'21	DMS	Transformer (E-D)	CD
Informer (Zhou et al. 2021)	AAAI'21	DMS	Transformer (E-D)	CD
Triformer (Cirstea et al. 2022)	IJCAI'22	DMS	Transformer (E)	CD
LightTS (Zhang et al. 2022)	-	DMS	MLP	CD
Koopa (Liu et al. 2023)	NeurIPS'23	DMS	Koopman Theory (Koopman 1931)	CD
CrossGNN (Huang et al. 2023)	NeurIPS'23	DMS	GNN	CD
WITRAN (Jia et al. 2023)	NeurIPS'23	DMS	RNN	CI
FreTS (Yi et al. 2023)	NeurIPS'23	DMS	MLP	CD
MICN (Wang et al. 2022)	ICLR'23	DMS	CNN	CD
TimesNet (Wu et al. 2022)	ICLR'23	DMS	CNN	CD
Crossformer (Zhang and Yan 2022)	ICLR'23	DMS	Transformer (E-D)	CD
PatchTST (Nie et al. 2022)	ICLR'23	DMS	Transformer (E)	CI
DLinear (Zeng et al. 2023)	AAAI'23	DMS	MLP	CI
NHITS (Challu et al. 2023)	AAAI'23	DMS	MLP	CD
SMARTformer (Li et al. 2023)	IJCAI'23	IMS & DMS	Transformer (E-D)	CD
TSMixer (Ekambaram et al. 2023)	KDD'23	DMS	MLP	CI/CD
TiDE (Das et al. 2023)	TMLR'23	DMS	MLP	CI
SegRNN (Lin et al. 2023)	-	DMS	RNN	CI
Client (Gao et al. 2023)	-	DMS	Transformer (E)	CD
Attraos (Hu et al. 2024)	NeurIPS'24	DMS	Chaos Theory (Devaney 2018)	CI
Ada-MSHyper (Shang et al. 2024)	NeurIPS'24	DMS	HGNN (Feng et al. 2019)	CI
SSCNN (Deng et al. 2024)	NeurIPS'24	DMS	CNN & Decomposition	CI
SOFTS (Han et al. 2024)	NeurIPS'24	DMS	MLP	CD
CycleNet (Deng et al. 2024)	NeurIPS'24	DMS	MLP	CI
CATS (Kim et al. 2024)	NeurIPS'24	DMS	Transformer (E)	CI
DeformableTST (Luo and Wang 2024)	NeurIPS'24	DMS	Transformer (E)	CI
TPGN (Liu et al. 2024)	NeurIPS'24	DMS	RNN	CI
AutoTimes (Liu et al. 2024)	NeurIPS'24	IMS	LLM (D)	CI
SparseTSF (Lin et al. 2024)	ICML'24	DMS	MLP	CI
SAMformer (Ilbert et al. 2024)	ICML'24	DMS	Transformer (E)	CD
TimeMixer (Chen et al. 2023)	ICLR'24	DMS	MLP	CD
Pathformer (Chen et al. 2023)	ICLR'24	DMS	Transformer (E)	CI
Time-LLM (Jin et al. 2023)	ICLR'24	DMS	LLM	CI

Continued on next page

Model	Venue	IMS/DMS	Backbone	CI/CD
iTransformer (Liu et al. 2023)	ICLR'24	DMS	Transformer (E)	CD
FITS (Xu et al. 2023)	ICLR'24	DMS	MLP	CI
CARD (Wang et al. 2023)	ICLR'24	DMS	Transformer (E)	CD
ModernTCN (Donghao and Xue 2023)	ICLR'24	DMS	CNN	CD
MSGNet (Cai et al. 2024)	AAAI'24	DMS	GNN	CD
UMixer (Ma et al. 2024)	AAAI'24	DMS	MLP	CD
HDMixer (Huang et al. 2024)	AAAI'24	DMS	MLP	CD
LeRet (Huang et al. 2024)	IJCAI'24	DMS	LLM + Retentive Net	CI
PatchMixer (Gong et al. 2024)	IJCAI'24	DMS	CNN	CI
SDformer (Zhou et al. 2024)	IJCAI'24	DMS	Transformer (E)	CD
SCAT (Zhou et al. 2024)	IJCAI'24	DMS	Transformer (E)	CI
Fredformer (Piao et al. 2024)	KDD'24	DMS	Transformer (E)	CD
MCformer (Han et al. 2024)	IoT-J'24	DMS	Transformer (E)	CD

In summary, the literature on point LTSF is extensive, with numerous methods achieving strong performances. Hence, determining the definitive state-of-the-art in point LTSF is challenging due to the vast and rapidly evolving literature. However, certain models, DLinear, PatchTST and iTransformer (Zeng et al. 2023; Nie et al. 2022; Liu et al. 2023), have emerged as de facto standards for comparison, frequently adopted as baseline or comparative methods in a wide range of recent works (Jia et al. 2024, 2023; Lu et al. 2024; Lin et al. 2023; Han et al. 2024; Lin et al. 2024; Luo and Wang 2024; Hu et al. 2024; Shang et al. 2024). Consequently, we consider them representative of the current state-of-the-art in point LTSF. Nonetheless, the distinction between IMS and DMS strategies has been largely overlooked, with DMS decoding being often adopted by default. Moreover, DMS forecasting can underperform in certain settings, which has not been sufficiently investigated in prior work. To address this gap, we empirically examine scenarios when and why DMS may fall short, using *multi-world* examples to highlight the conditions under which IMS offers advantages. Furthermore, while SOTA point LTSF models are highly effective at predicting the conditional mean (Li et al. 2023), many real-world scenarios require a more nuanced understanding of uncertainty, making probabilistic forecasts preferable. Hence, the next section reviews existing probabilistic models proposed for time series forecasting.

3.2. Related Work on Probabilistic Forecasting

Traditionally, point or single-valued forecasting methods have been among the most common forecasting techniques due to their simplicity (Benidis et al. 2022). However, these methods lack information about uncertainties of their predictions, which can be a major disadvantage when the forecasts are used in decision-making. Hence, Gneiting and Katzfuss (2014) state that forecasts should take on a probabilistic form, as this enables the modelling of uncertainties in forecasts. As formalized in Chapter 2, the general goal is to produce a probabilistic estimate for $p(x_{L+1:L+H}|x_{1:L})$, see Equation 2.1. Moreover, this distribution can be represented equivalently by its probability den-

sity function (PDF), the cumulative density function (CDF) or its inverse, the quantile function (Benidis et al. 2022). In the following, we present various methods for generating probabilistic forecasts, precise formulations of adopted approaches are deferred to Section 4.2.

Parametric Distributional Forecasting. A common approach to produce an estimate for $p(x_{L+1:L+H}|x_{1:L})$ is via parametric distributional forecasting, where models typically output location and spread parameters of a pre-chosen probability distribution, which can be maximized via the log-likelihood with respect to the ground-truth $x_{L+1:L+H}$ (Bergsma et al. 2022). For instance, an early NN-based example is the work of Nix and Weigend (1994), where neural networks were trained to output the mean $\hat{\mu}$ and variance $\hat{\sigma}^2$ of a Gaussian distribution for regression tasks. While Gaussian likelihoods are common, alternative distributions such as Student-t (Alexandrov et al. 2020), negative binomial (Salinas et al. 2020) or Gaussian mixture distributions (Mukherjee et al. 2018) have been used depending on the statistical properties of the data. DeepAR (Salinas et al. 2020), an IMS CI RNN-based approach, adopts parametric distributional forecasting with a negative binomial distribution to model demand data. Building on this, VQ-AR (Rasul et al. 2022) combines the DeepAR backbone with a Vector Quantized-Variational Autoencoder (VQ-VAE) architecture (Van den Oord et al. 2017), introducing a discrete latent bottleneck that captures recurring temporal patterns in probabilistic forecasting. Extending this idea to Transformer-based architectures, VQ-TR (Rasul et al. 2023) integrates VQ-VAE into the attention mechanism of Transformers. Lastly, CNN-based models like BiTCN (Sprangers et al. 2023) and DeepTCN (Chen et al. 2019) also leverage parametric distributional forecasting but with a DMS CD strategy. A disadvantage of these approaches is that they require an *a priori* choice of the distributional form and are limited to the parametric assumptions, which may not capture complex data dynamics. To address the limitations of fixed-form parametric models, non-parametric approaches have gained interest in the community.

Flexible Density Estimation. Among non-parametric approaches, normalizing flows (Tabak and Turner 2013; Papamakarios et al. 2021) have emerged as a powerful tool for flexible density estimation. Normalizing flows, such as Real NVP (Dinh et al. 2017) and Masked Autoregressive Flow (MAF) (Papamakarios et al. 2017), transform a simple base distribution, e.g. isotropic Gaussian, into a complex target distribution through a series of invertible and differentiable mappings. Invertibility ensures the preservation of probability mass and enables the evaluation of the corresponding density function at all points (Benidis et al. 2022). A unified description of normalizing flows and their core principles is provided by Papamakarios et al. (2021). For time series forecasting, Rasul et al. (2020) combine IMS backbones, e.g. RNN and Transformer, with conditioned normalizing flows to capture multivariate temporal dependencies without restrictive parametric assumptions. Furthermore, MANF (Feng et al. 2024) combines conditioned normalizing flows with multi-scale attention and relative positional encoding to model multivariate dependencies efficiently in a DMS fashion. TACTiS and TACTiS-2 (Drouin et al. 2022; Ashok

et al. 2023) introduce an IMS Transformer-based Copula model using Deep Sigmoidal Flows (Huang et al. 2018) to estimate marginal CDFs. Although flow-based models retain tractable likelihood computation via the change of variables formula, they have problems with discrete data distributions, often present in TSF applications (e.g., sales data) (Rasul et al. 2020). Moreover, learning highly flexible continuous distributions for discrete data may encourage learning distributions with spiking densities at each possible discrete value (Uria et al. 2013). Altogether, this hinders training the models to maximize the likelihood (Bergsma et al. 2022). To address this, dequantizing (Rasul et al. 2020), e.g. adding `Uniform[0,1]` noise, may be applied to bound the log-likelihood (Theis et al. 2016). However, this assumes that the discrete nature of the series is known in advance and that the potential loss in precision is justified relative to the advantages offered by continuous models (Bergsma et al. 2022). In contrast to flow-based models, C2FAR (Bergsma et al. 2022) represents time series variables through a hierarchical sequence of categorical distributions, built on top of the DeepAR framework. Instead of relying on fixed parametric forms, C2FAR generates increasingly finer intervals of support in an autoregressive manner, where each step is conditioned on coarser previous intervals. This hierarchical discretization allows it to better capture multi-modal behaviors and extreme values compared to flat binning or standard parametric approaches. Building on this, SutraNets (Bergsma et al. 2023) extend the idea to long-term time series problems by dividing inputs into frequency-based sub-series, which is different from the regular patching of adjacent time steps as done in models like PatchTST (Nie et al. 2022) or SegRNN (Lin et al. 2023). Each sub-series in SutraNets is modeled by its own C2FAR-LSTM.

Generative Diffusion Models. While normalizing flows offer tractable likelihoods and exact sampling through invertible transformations, their structural constraint, particularly invertibility and the need for tractable Jacobians, can limit their expressiveness in modeling complex, high-dimensional, and multimodal distributions (Benidis et al. 2022). In contrast, energy-based models (EBMs) relax these constraints by modeling unnormalized log-probabilities, especially important in high-dimensional spaces (LeCun et al. 2006). However, EBMs are notoriously difficult to train due to challenges in sampling and normalizing (Du and Mordatch 2019). Diffusion probabilistic models (Ho et al. 2020; Sohl-Dickstein et al. 2015; Graikos et al. 2022), such as the well-known denoising diffusion probabilistic model (DDPM) (Ho et al. 2020), can be viewed as a practical compromise: they implicitly learn energy gradients via score-based training (Hyvärinen 2005; Song et al. 2021) and enable stable sampling using Langevin-like denoising processes (Neal 2011; Welling and Teh 2011). On a high level, diffusion probabilistic models operate by first applying a forward process that gradually corrupts data into noise, followed by a reverse process that reconstructs the original data from the noise (Ho et al. 2020). Over the last few years, diffusion-based generative models have emerged as strong generative tools, achieving SOTA performances in text generation (Li et al. 2022), audio (Kong et al. 2020) and image synthesis (Dhariwal and Nichol 2021). In the context of TSF, diffusion-based models often guide generation by conditioning on

partial observations, reference samples, decompositions or architectural priors such as RNNs, transformers or state space models (Rasul et al. 2021; Tashiro et al. 2021; Alcaraz and Strodthoff 2022; Shen and Kwok 2023; Liu et al. 2024; Shen et al. 2023). For instance, TimeGrad (Rasul et al. 2021) uses RNN hidden states, while TimeDiff (Shen and Kwok 2023) incorporates task-specific conditioning like future mixup and autoregressive initialization. On the other hand, mr-Diff (Shen et al. 2023) leverages the multi-scale structure of time series by conditioning the denoising process on progressively refined trends, starting from coarse to fine levels. Similarly, TMDM (Li et al. 2023) and RATD (Liu et al. 2024) integrate transformer-based and retrieval-augmented conditioning, respectively. Going one step further, D^3M (Yan et al. 2024) introduces a decomposable denoising diffusion framework that unifies continuous flow models and diffusion models, achieving high-speed generation with fewer diffusion steps. Alternatively, TSDiff (Kollovieh et al. 2023) mitigates explicit conditioning during training by adopting an unconditional framework, using a self-guidance mechanism at inference time to adapt to tasks like forecasting and refinement without auxiliary networks. Contrary to previous approaches, DSPD-GP (Biloš et al. 2023) treats the time series as continuous functions rather than discrete measurements, defining diffusion not over discrete vectors but over functions, enabling direct handling of irregularly-sampled time series. Moreover, their Stochastic Process Diffusion framework applies diffusion in function space, using correlated noise from Gaussian processes to preserve temporal continuity and handle irregular sampling. On the other hand, D^3VAE (Li et al. 2022) proposes a bidirectional VAE augmented with a coupled diffusion process, which simultaneously diffuses input and target series to reduce uncertainty. It further integrates denoising score matching and disentangled latent variables to improve interpretability and robustness, demonstrating strong performance on short and noisy time series.

Latent Generative methods. While diffusion-based models have gained traction in generative modeling for their flexibility and ability to capture complex, multimodal distributions, they often suffer from high computational cost and slow sampling, limitations that can be especially problematic in time-sensitive or resource-constrained forecasting scenarios (Yeğin and Amasyalı 2024; Yang et al. 2024). In contrast, latent-variable approaches such as Variational Autoencoders and (probabilistic) State Space Models offer a compelling alternative by trading off expressivity for faster inference and improved interpretability (Tong et al. 2022; de Bézenac et al. 2020).
Variational Autoencoders (VAEs) (Kingma and Welling 2014; Rezende et al. 2014) simplify generative modeling by learning to represent complex data distributions in a lower-dimensional latent space using variational inference. In variational inference, the main idea is to approximate the true distribution with a simpler distribution, e.g. Gaussian, and minimize the Kullback-Leibler (KL) divergence, shown in the upcoming Section 3.3 in Equation 3.10, between the approximate and true distribution, also known as evidence lower bound optimization (Yeğin and Amasyalı 2024). Broadly speaking, VAEs first encode the data into a lower-dimensional latent space where a simpler probabilistic model can be imposed, then forecasts are generated by decoding samples drawn from this

latent distribution back into the observation space. Compared to other likelihood-based models like normalizing flows or energy-based models, VAEs offer efficient, tractable sampling and readily accessible inference via encoder networks (Vahdat and Kautz 2020). Recent advances in other generative tasks, e.g. image generation (Vahdat and Kautz 2020; Luo et al. 2025), have motivated an adoption in time series forecasting tasks. For instance, the Temporal Latent AutoEncoder (TLAE) (Nguyen and Quanz 2021) introduces a nonlinear factorization framework for multivariate time series, enabling end-to-end learning of complex latent dynamics while preserving scalability. Similarly, VSMHN (Li et al. 2021), based on the conditional VAE (Sohn et al. 2015), handles asynchronous event-driven data with aligned time encodings to jointly forecast across heterogeneous temporal sources. To better address long-range dependencies and structural interpretability, PDTrans (Tong et al. 2022) fuses Transformer-based temporal modeling with a VAE-based latent decomposition, providing interpretable forecasts through trend and seasonality disentanglement. The Latent Diffusion Transformer (LDT) (Feng et al. 2024) compresses high-dimensional multivariate series into latent representations using a statistics-aware autoencoder and generates forecasts via a diffusion-based generator with self-conditioning. Likewise, the D^3VAE (Li et al. 2022) model fuses the Nouveau VAE (Vahdat and Kautz 2020) with a coupled diffusion process and multiscale denoising to enhance robustness under limited or noisy data.

State Space Models (SSMs)(Hyndman et al. 2002; Seeger et al. 2016; Durbin and Koopman 2012) provide a framework for modeling sequential data via latent variables that evolve over time according to structured transition dynamics and generate observations through stochastic emission processes. Commonly implemented deterministic time series forecasting methods such as Exponential Smoothing (Hyndman et al. 2002, 2008), ARIMA (Box and Pierce 1970), and LSTMs (Hochreiter and Schmidhuber 1997) can all be interpreted as special cases or deterministic approximations of SSMs (Durbin and Koopman 2012). A classic probabilistic instance is the linear-Gaussian SSM (Roweis and Ghahramani 1999), which offers closed-form solutions for filtering, smoothing, and likelihood computation, enabling efficient handling of missing data, and tractable multi-step forecasting with full uncertainty quantification (avoiding error accumulation) (de Bézenac et al. 2020). Furthermore, in their probabilistic form, SSMs define joint distributions over latent states and observations, such as in Gaussian Process SSMs (Ko and Fox 2011; Deisenroth et al. 2012). This enables uncertainty quantification through inference techniques like Kalman filtering (Kalman 1960), treating the hidden state as a distribution that is recursively updated and propagated to generate predictive distributions over future observations. In probabilistic TSF, recent works have explored hybrid models that combine the structural benefits of SSMs with the expressivity of deep learning, where one prominent direction is the use of deep neural networks to parameterize SSM components. For example, DeepState (Rangapuram et al. 2018) employs an RNN to learn a shared global mapping from covariates to the parameters of a linear-Gaussian SSM, in which each individual time series has an associated linear Gaussian SSM whose parameters are dynamically generated by the RNN. Similarly, DNLSSM (Du et al. 2023) integrates LSTM networks with the unscented Kalman filter (Julier and Uhlmann 2004) to better model non-linear dynamics. Extending these ideas to attention-based tempo-

ral dependencies, the Probabilistic Transformer (ProTran) (Tang and Matteson 2021) replaces recurrent components with a Transformer architecture, enabling the model to learn non-Markovian dependencies in the latent space via self-attention. From a different standpoint, PR-SSM (Doerr et al. 2018) substitute the parametric transition dynamics with Gaussian Processes, where training relies on doubly stochastic variational inference. To address the restrictive assumption of Gaussianity, flow-based SSMs have been proposed. For instance, the Normalizing Kalman Filter (NKF) (de Bézenac et al. 2020) augments a linear-Gaussian SSM by integrating normalizing flows into the observation model. This allows the latent state dynamics to remain analytically tractable via Kalman filtering, while enabling the observation distribution to flexibly model complex, non-Gaussian behavior. Similarly, EMSSM (Sun et al. 2022) implements an external memory mechanism and conditional normalizing flows, enhancing the model's capacity to capture long-range dependencies and adapt to distributional shifts.

Generative Adversarial Paradigms. Generative Adversarial Networks (GANs) (Goodfellow et al. 2014), on the other hand, bypass likelihood estimation altogether, directly learning to generate realistic samples through adversarial training. Furthermore, GANs typically consist of two NNs, a generator and a discriminator. In their min-max adversarial training game, the generator creates synthetic samples while the discriminator is tasked with distinguishing between artificial samples from the generator and real samples (Wu et al. 2020). GAN-based models have demonstrated impressive results in various generative tasks, such as image generation (Jiang et al. 2021) and audio synthesis (Donahue et al. 2018). Similarly, GANs primary focus in time series applications also lies in time series synthesis and generation, e.g. see timeGAN (Yoon et al. 2019), GT-GAN (Jeon et al. 2022) or GAN-based approaches in the Time Series Generative Modeling benchmark (Nikitin et al. 2024). Nevertheless, GAN-based probabilistic time series forecasting approaches were implemented as well. For instance, the Adversarial Sparse Transformer (AST) (Wu et al. 2020) combines a sparse Transformer with adversarial training to improve time series forecasting. AST employs a generator-discriminator framework, where the generator learns sparse attention patterns for forecasting, while the discriminator ensures sequence-level fidelity. Furthermore, Koochali et al. (2021) introduce ProbCast, a probabilistic forecasting model based on conditional GAN (Mirza and Osindero 2014). On a final note, despite their potential, GANs remain notoriously difficult to train, often facing challenges such as hyperparameter sensitivity, training instability, and mode collapse (Yeğin and Amasyalı 2024).

Quantile. In practice, specifying an exact (parametric) distribution is often unnecessary. Instead, estimating a few key quantiles is sufficient for making optimal decisions, e.g. in epidemiologic forecasting (Bracher et al. 2021; Ray et al. 2020), wind power forecasting (Wan et al. 2017) or the classical newsvendor problem (Gneiting et al. 2023; Tarima and Zenkova 2020; Harsha et al. 2021), by helping quantify uncertainty and minimize losses (Bracher et al. 2021; Wen et al. 2018; Gneiting et al. 2023). As a result, a common non-parametric approach to modeling $p(x_{L+1:L+H}|x_{1:L})$ is via the quantile

function, where the models are typically trained using the quantile loss (see Equation 3.15 of Section 3.3). Quantile regression methods (Koenker and Bassett 1978; Koenker 2005) are frequently applied in forecasting, either by predicting a single quantile (for point forecasts) or multiple quantiles at once (Wen et al. 2018). Effectively, this method approximates the quantile function by targeting specific quantile levels (Benidis et al. 2022). Two notable approaches of this kind are the Multi-horizon Quantile Recurrent Forecaster (MQ-RNN) (Wen et al. 2018) and the Temporal Fusion Transformer (TFT) (Lim et al. 2021). MQ-RNN combines sequence-to-sequence NNs with quantile regression and a DMS strategy. On the other hand, TFT leverages attention mechanisms for interpretable DMS forecasting, using specialized components like gating mechanisms and variable selection networks to provide insights into temporal dynamics. Nevertheless, these approaches require the specification of a set of quantile levels *a priori*, necessitating retraining when querying new quantile levels (Park et al. 2022). As another option, Implicit Quantile Networks (IQNs) (Dabney et al. 2018) learn to map samples from a uniform distribution (`Uniform[0,1]`) to corresponding quantile values of the target distribution, removing the need for predefined quantile levels. This allows for continuous and arbitrarily fine-grained quantile estimation. Building on this, several models adapt IQNs for time series forecasting. For example, IQN-RNN (Gouttes et al. 2021) combines the IQN framework with RNNs, enabling autoregressive probabilistic forecasts without assuming a specific distributional form. GQFormer (Jawed and Schmidt-Thieme 2022) introduces a novel multi-task loss that encourages both sharpness and diversity in quantile estimates, allowing the model to capture multiple modes in complex multivariate joint distributions. Finally, GMQ-forecaster (Wen and Torkkola 2019) integrates IQNs with multivariate copula modeling, using IQNs for marginals and a learned copula to represent their dependencies, resulting in a fully generative joint distribution. Despite achieving SOTA performance in probabilistic TSF, the aforementioned quantile-based models remain susceptible to the quantile crossing problem, a violation of the inherent monotonicity of quantile functions (Park et al. 2022). Specifically, when different quantile levels (e.g., $\alpha_1 < \alpha_2$) are modeled independently, for instance via separate functions (r_{α_1} and r_{α_2}), it is possible to encounter inputs x for which $r_{\alpha_1}(x) > r_{\alpha_2}(x)$, thereby breaking the non-decreasing property of quantile functions (Gasthaus et al. 2019). Therefore, as an alternative to traditional quantile regression, one can directly model the entire quantile function by imposing a parametric structure instead, which requires the function to be defined over the unit interval [0,1] and to be monotonically increasing (Benidis et al. 2022). A practical way to satisfy these constraints is through the use of linear splines, as demonstrated in the Spline Quantile Function (SQF)-RNN framework (Gasthaus et al. 2019), where an RNN is trained to produce the parameters of a monotonic spline. By construction, this avoids quantile crossing and allows the full quantile function to be recovered from a compact parameterization. However, the SQF-RNN model has reduced flexibility in tail regions (Park et al. 2022). Hence, Park et al. (2022) propose Incremental (Spline) Quantile Functions, which extend SQF by adding tail extrapolation strategies, such as exponential Pareto distributions. Finally, the Multivariate Quantile Function Forecaster (MQF^2) (Kan et al. 2022) models the multivariate quantile function as the gradient of a convex function, parametrized via partially input convex neural networks

(Amos et al. 2017; Huang et al. 2020).

Multivariate dependencies. The probabilistic models considered so far involve an additional layer of complexity: the presence of multivariate dependencies in modeling the conditional distribution $p(\mathbf{x}_{L+1:L+H} \mid \mathbf{x}_{1:L})$. These dependencies can be broadly categorized along two primary axes.

- **Cross-series (multi-channel) dependencies**, for instance involving the modeling of the multivariate distribution $p(\mathbf{x}_t)$ across all N series at time step t, with $\mathbf{x}_t \in \mathbb{R}^N$. Methods that model these dependencies are often simply classified as multivariate probabilistic forecasting methods (Benidis et al. 2022).
- **Temporal dependencies**, for example referring to the joint modeling of the future horizon for a single series, $p(\mathbf{x}^{(i)}_{L+1:L+H})$, where $\mathbf{x}^{(i)}_{L+1:L+H} \in \mathbb{R}^H$.

Models may capture neither, one, or both forms of dependencies. Nevertheless, in the following, our analysis focuses primarily on temporal dependencies. This emphasis is motivated by the observation that DMS models, albeit dominant in point forecasting (see Table 3.1), generate all future steps in a single forward pass conditioned only on the input context. Moreover, this means that DMS models cannot naturally model how future points depend on each other, instead assuming independence between future steps (Taieb et al. 2012), unless such dependencies are explicitly modeled. In contrast, probabilistic IMS methods typically factorize the joint temporal distribution $p(\mathbf{x}_{L+1:L+H} \mid \mathbf{x}_{1:L})$ into a product of conditional one-step distributions, as shown in Equation 2.5 of Chapter 2. Therefore, they naturally model temporal dependencies as sequential step-by-step predictions. In addition to this, many point LTSF models adopt a CI design, explicitly excluding cross-series dependencies. As such, cross-series interactions are often not represented in the underlying model architecture, making it difficult to include the probabilistic multi-channel dependencies during the transition from point LTSF to probabilistic LTSF. Irrespective of the type of multivariate dependencies being modeled (cross-series, temporal, or both) various modeling strategies can be employed to approximate multivariate distributions. Two key challenges in modeling multivariate distributions are the positive-definiteness constraint and the quadratic complexity of estimating full covariance matrices, which requires $O(N^2)$ parameters, where $N = H$ in the case of temporal dependencies (Pourahmadi 2011). To address this, one approach is to impose structural constraints on the covariance matrix by using a low-rank plus diagonal decomposition, which ensures positive-definiteness and reduces the number of parameters (Wu et al. 2020; Horn and Johnson 2012). An alternative strategy is to bypass direct estimation of the covariance structure and instead use copulas, which provide a way to decouple the modeling of marginal distributions from their joint dependence structure (Wilson and Ghahramani 2010; Größer and Okhrin 2022). According to Sklar's theorem (Sklar 1959), any multivariate distribution can be expressed in terms of its marginals and a copula function that captures the dependencies among variables. This decomposition is particularly useful in time series settings, where marginals can exhibit diverse

characteristics (e.g., skewness, seasonality) and dependencies may be nonlinear and time-varying (e.g., temporal autocorrelation or inter-series interactions) (Salinas et al. 2019). GPVar (Salinas et al. 2019) is an example of a parametric copula approach, in which the marginals are estimated independently and then transformed into a latent Gaussian space, where dependencies are modeled using a Gaussian copula with a low-rank plus diagonal covariance structure. Although this reduces the parameter complexity from $O(N^2)$ to $O(N)$, parametric approaches make strong assumptions about the underlying data distribution (Ashok et al. 2023). To address this, nonparametric copula estimators such as TACTiS and TACTiS-2 (Drouin et al. 2022; Ashok et al. 2023) offer greater flexibility by learning the copula function directly from the data without assuming a fixed parametric form. TACTiS (Drouin et al. 2022) models the copula density autoregressively in the copula-transformed space using a transformer-based architecture, where each variable is conditioned on previously modeled ones. To ensure that the learned copula is valid i.e., invariant to the ordering of variables, TACTiS averages over multiple variable permutations during training. However, achieving full permutation invariance would require averaging over all possible orderings, which results in factorial complexity $O(N!)$ in the number of variables (with $N = H$ for temporal dependencies) (Ashok et al. 2023). TACTiS-2 (Ashok et al. 2023) mitigates this by implementing a two-stage training protocol to learn marginals and the copula. It first fits normalizing flows to estimate marginal CDFs, then uses a fixed-order attention decoder with causal masking to model the joint copula density. This reduces complexity from factorial $O(!N)$ to linear $O(N)$. Beyond the copula framework, a different direction involves directly learning the multivariate quantile function, the inverse CDF, as done by MQF^2 (Kan et al. 2022). Furthermore, MQF^2 represents the joint quantile function as the gradient of a convex function parameterized by an input convex neural network (ICNN) (Amos et al. 2017), a NN architecture designed to be convex with respect to a subset of its inputs through specific structural constraints. MQF^2 avoids quantile crossing but introduces additional complexity during both training and inference, as evaluating the quantile function involves solving an optimization problem. Lastly, several generative approaches naturally extend to jointly model either or both of cross-series and temporal dependencies by learning full multivariate distributions. For instance, this includes diffusion (TMDM (Li et al. 2023)), VAE (D^{3}VAE (Li et al. 2022)) or GAN (AST (Wu et al. 2020)) models.

LTSF and probabilistic forecasting. So far, we have seen two primary branches of time series forecasting: long-term point forecasting methods and probabilistic forecasting approaches. However, integrating both approaches into long-term distributional forecasting is still a significant open challenge (Zhang et al. 2024). Furthermore, in their analysis Zhang et al. (2024) find that point LTSF methods tend to focus on a DMS strategy, due to error accumulation effects of IMS methods on longer forecasting horizons. Additionally, their analysis shows that probabilistic TSF methods adopt either IMS or DMS strategies without a clear preference, which they account to shorter forecasting horizons. Looking at Table 3.1 and Table 3.2, we come to the same conclusion. Furthermore, although probabilistic methods consider mostly the same datasets as point LTSF methods

(Kollovieh et al. 2023; Rasul et al. 2023; Drouin et al. 2022; Tong et al. 2022), there are only a few probabilistic methods listed in Table 3.2, where the forecasting horizon H is larger than 300. In contrast, the longest forecasting horizon in the default LTSF setup is 720 (Zhou et al. 2021). Nevertheless, recent studies have begun to explore modeling longer-term dependencies in probabilistic TSF. For instance, SSSD (Alcaraz and Strodthoff 2022) includes a long-term forecasting experiment inspired by the point-based LTSF framework of Zhou et al. (2021). However, their evaluation in this setting focuses solely on point predictions, neglecting the probabilistic forecasting performance. More recently, SutraNets (Bergsma et al. 2023) and RATD (Liu et al. 2024) have investigated probabilistic forecasting over extended horizons. While SutraNets consider long horizons (e.g., $H > 300$), those are limited to the MNIST dataset (LeCun et al. 1998), which is not a standard benchmark in time series forecasting. In contrast, RATD (Liu et al. 2024) is evaluated under a more classical LTSF setting, comparing performance against established point forecasting models, such as iTransformer, Informer, PatchTST, TimesNet, and DLinear (Liu et al. 2023; Zhou et al. 2021; Nie et al. 2022; Wu et al. 2022; Zeng et al. 2023), as well as probabilistic models like CSDI and D^3VAE (Tashiro et al. 2021; Li et al. 2022). However, point LTSF models in this evaluation are not adapted nor interpreted within a probabilistic framework. From a different standpoint, several model-agnostic approaches have been proposed to derive probabilistic forecasts from point forecasting methods. VQ-TR (Rasul et al. 2023), for example, is evaluated in a probabilistic setting alongside transformer-based methods that were originally proposed for point LTSF, e.g. Informer, Autoformer and PatchTST (Zhou et al. 2021; Wu et al. 2021; Nie et al. 2022). To enable probabilistic outputs, Rasul et al. (2023) modify deterministic TSF architectures to either implement a parametric distributional forecasting approach with a negative binomial or Student-t distribution or implement an IQN head. However, the forecasting horizons considered are relatively short and not directly comparable to common LTSF settings. TMDM (Li et al. 2023) presents a plug-and-play framework compatible with arbitrary point forecasting models. Its core idea is to leverage the strength of LTSF models in estimating the conditional mean, using this to guide the generation of full predictive distributions via diffusion. TMDM integrates architectures such as Informer, Autoformer or NSformer (Zhou et al. 2021; Wu et al. 2021; Liu et al. 2022) as backbones, but also limits forecasting to horizons no greater than 196 steps. Finally, ProbCast (Koochali et al. 2021) introduces a GAN-based framework for transforming point forecasting models into probabilistic ones. However, their evaluation also does not address extended forecasting horizons and is restricted to a simple RNN-based backbone.

To summarize, in the domain of probabilistic TSF, the diversity of forecasting setups, e.g. with respect to prediction horizons (see Table 3.2), makes it challenging to establish consistent benchmarks or perform definitive model comparisons across studies. Furthermore, we do not define a clear state-of-the-art for probabilistic forecasting as we have for point LTSF, since the studies tend to be more isolated and less directly comparable. In addition to this, as discussed, only a limited number of studies have explored probabilistic methods within the LTSF setting (Alcaraz and Strodthoff 2022; Bergsma et al. 2023; Liu et al. 2024). Likewise, only a few works have attempted extending point

LTSF architectures to support probabilistic outputs (Rasul et al. 2023; Li et al. 2023; Koochali et al. 2021). Hence, in this work we aim to bridge this gap by evaluating point LTSF methods in a probabilistic setting, focusing on longer forecasting horizons than prior studies and assessing them using probabilistic metrics. In doing so, we also analyze the behavior of IMS and DMS strategies more in depth, identifying cases where DMS methods fail to yield reliable probabilistic forecasts. To properly train and evaluate probabilistic methods, suitable evaluation metrics are necessary, which is the subject of the following and final section of the related work.

Table 3.2.: Overview of key probabilistic TSF models. Each entry includes the model name, publication venue, decoding strategy (IMS/DMS), model backbone (e.g., CNN, SSM, RNN, Encoder-only (E), Decoder-only (D) or Encoder-Decoder (E-D) Transformer), whether it uses a channel-independent or dependent strategy (CI/CD), maximum forecasting horizon (H) and the core probabilistic method employed (e.g., diffusion models, quantile methods or variational autoencoders).

Model	**Venue**	**IMS/DMS**	**Backbone**	**CI/CD**	**H**	**Prob. Method**
MQ-RNN/MQ-CNN (Wen et al. 2018)	NeurIPS'17	DMS	CNN/ RNN	CI	52	Quantiles
DeepState (Rangapuram et al. 2018)	NeurIPS'18	DMS	SSM/ RNN	CI	168	Prob. Latent Variable
GP-Copula (Salinas et al. 2019)	NeurIPS'19	IMS	RNN	CD	30	Distributional & Copula
GMQ (Wen and Torkkola 2019)	ICML'19	IMS	CNN/ RNN	CD	30	IQN & Copula
SQF-RNN (Gasthaus et al. 2019)	AISTATS'19	IMS	RNN	CI	60	Quantile Function
DeepTCN (Chen et al. 2019)	MileTS'19	DMS	CNN	CD	31	Distributional/ Quantiles
AST (Wu et al. 2020)	NeurIPS'20	IMS/DMS	Transformer(E-D)	CI	168	GAN
DeepAR (Salinas et al. 2020)	IJF'20	IMS	RNN	CI	52	Distributional
ProTran (Tang and Matteson 2021)	NeurIPS'21	DMS	SSM/Transformer(E)	CD	30	Prob. Latent Variable
CSDI (Tashiro et al. 2021)	NeurIPS'21	DMS	Transformer(E)	CD	24	Diffusion
TimeGrad (Rasul et al. 2021)	ICML'21	IMS	RNN	CD	30	Diffusion
IQN-RNN (Gouttes et al. 2021)	ICML'21	IMS	RNN	CI	30	IQN
LSTM-Real-NVP/ Transformer-MAF (Rasul et al. 2020)	ICLR'21	IMS	RNN/ Transformer(E-D)	CD	30	Flow
VSMHN (Li et al. 2021)	AAAI'21	IMS	RNN	CD	24	VAE

Continued on next page

Model	Venue	IMS/DMS	Backbone	CI/CD	H	Prob. Method
TLAE (Nguyen and Quanz 2021)	AAAI'21	DMS	AE/RNN	CD	24	VAE
ProbCast (Koochali et al. 2021)	Eng. Proc.'21	DMS	RNN	CD	24	GAN
TFT (Lim et al. 2021)	IJF'21	DMS	RNN	CI	30	Quantiles
D^3VAE (Li et al. 2022)	NeurIPS'22	DMS	MLP	CI	64	Diffusion & VAE
C2FAR (Bergsma et al. 2022)	NeurIPS'22	IMS	RNN	CD	48	Distributional
TACTiS (Drouin et al. 2022)	ICML'22	IMS	Transformer(E)	CI	72	Flow & Copula
EMSSM (Sun et al. 2022)	IJCAI'22	IMS	SSM & RNN	CI	30	Flow & Prob. Latent Variables
MQF^2 (Kan et al. 2022)	AISTATS'22	DMS	RNN & PICNN (Amos et al. 2017)	CD	24	Quantile Function
ISQF (Park et al. 2022)	AISTATS'22	DMS	CNN	CI	30	Quantile Function
DNLSSM (Du et al. 2023)	CAAI'22	IMS	RNN & SSM	CI	196	Prob. Latent Variable
GQFormer (Jawed and Schmidt-Thieme 2022)	IEEE Big Data'22	DMS	Transformer(E)	CD	168	IQN
VQ-AR (Rasul et al. 2022)	-	IMS	RNN	CI	30	Distributional
TSDiff (Kollovieh et al. 2023)	NeurIPS'23	DMS	SSM	CI	48	Diffusion
SutraNets (Bergsma et al. 2023)	NeurIPS'23	IMS	RNN	CI	392	Distributional
DSDP-GP (Biloš et al. 2023)	ICML'23	DMS	RNN	CI	30	Diffusion
mr-diff (Shen et al. 2023)	ICLR'24	DMS	MLP	CD	24	Diffusion
SSSD (Alcaraz and Strodthoff 2022)	TMLR'23	DMS	SSM	CD	672	Diffusion
pTSE (Zhou et al. 2023)	IJCAI'23	DMS	Ensemble	CD	24	HMM
BiTCN (Sprangers et al. 2023)	IJF'23	DMS	CNN	CD	30	Distributional
PDTrans (Tong et al. 2022)	SDM'23	IMS & DMS	Transformer(E-D)	CI	168	Prob. Latent Variable
RATD (Liu et al. 2024)	NeurIPS'24	DMS	Transformer(E)/RAG/ DiffWave	CD	336	Diffusion
D^3M (Yan et al. 2024)	ICML'24	DMS	SSM & WaveNet (CNN)	CD	30	Diffusion & Flow
TACTiS-2 (Ashok et al. 2023)	ICLR'24	IMS	Transformer(E)	CI	72	Flow & Copula
TMDM (Li et al. 2023)	ICLR'24	DMS	Transformer(E)	CD	192	Diffusion
VQ-TR (Rasul et al. 2023)	ICLR'24	IMS	Transformer (E-D)	CI	48	Distributional
LDT (Feng et al. 2024)	AAAI'24	DMS	Transformer(E-D)-AE	CD	48	Diffusion

Continued on next page

Model	Venue	IMS/DMS	Backbone	CI/CD	H	Prob. Method
MANF (Feng et al. 2024)	IEEE TKDE'24	DMS	Transformer(E)	CD	30	Flow
GPF-WI (Wang et al. 2024)	CISS'24	DMS	AE	CD	24	VAE

3.3. Related Work on Probabilistic Scoring Rules

This section briefly motivates and introduces typical evaluation metrics for probabilistic forecasts. First, we go over common scoring rules for distributional forecasts, then we focus on scoring measures for interval and quantile forecasts.

3.3.1. Proper Scoring Rules

To this day, probabilistic forecasting commonly follows Gneiting et al. (2007) by training a model on the available data to maximize the sharpness of its predictive distribution subject to calibration. Sharpness describes the concentration of predictive distributions, i.e. the sharper a forecast, the more informative it is. Calibration refers to the consistency between the predictions and the actual observations, as we ideally want the true underlying observations to be calibrated such that they are interchangeable with random draws from the predictive distribution. According to Gneiting and Katzfuss (2014), both concepts can be summarised into a single numerical score by (proper) scoring rules, thus enabling a collective assessment of calibration and sharpness. Generally, probabilistic scoring rules $S(\hat{F}, x)$ are defined as measures that summarise the predictive quality of forecasts by assigning a numerical score, when given the predictive distribution $\hat{F} \in \mathcal{F}$ and the materialization x (Gneiting et al. 2007). While many scoring rules can be used to assess forecast quality, proper scoring rules are the most widely adopted, as they are specifically designed to encourage honest and well-calibrated forecasts (Machete 2013) Furthermore, the application of non-proper scoring rules may lead to erroneous inferences and is thus not recommended (Gneiting and Ranjan 2011). The expected score for $\hat{F}$ under the unknown true underlying distribution P is given as

$$s(\hat{F}, P) = \int S(\hat{F}, x)\ dP(x) \tag{3.1}$$

(Gneiting and Raftery 2007). Furthermore, a scoring rule $S(\hat{F}, x)$ is proper conditioned on the class of distributions $\mathcal{F}$ when

$$s(P, P) \leq s(\hat{F}, P) \quad \forall \hat{F}, P \in \mathcal{F} \tag{3.2}$$

holds and it is strictly proper if equality in Equation 3.2 is only achieved when $\hat{F} = P$ (Gneiting and Raftery 2007; Tyralis and Papacharalampous 2024). Therefore, (strictly) proper scoring rules are minimal in expectation for the ground-truth distribution (Marcotte et al. 2023), hence encouraging forecasters to return this unknown distribution,

which is often referred to as honest reporting (Winkler et al. 1996; Winkler and Murphy 1968). On top of propriety, scoring rules may also be classified into univariate or (fully) multivariate scoring rules (Ziel and Berk 2019). Univariate scoring rules separately evaluate each predictive univariate distribution of multi-step forecasters, whereas (fully) multivariate scoring rules additionally assess the dependencies between the distributions. In the following chapters, we consider scoring rules that we want to minimize, i.e. lower numerical scores represent better predictive performance.

One popular proper score for full predictive distributions is the logarithmic score or logarithmic likelihood, which originates from Good (1952). In a discrete setting, it can be described as

$$\text{logS}(\hat{F}, x) = \log p(x) \tag{3.3}$$

where $\hat{F}$ is the forecast and x the materialization, so that $p(x)$ represents the probability assigned to x by $\hat{F}$ (Bracher et al. 2021). The logarithmic score is a strictly proper scoring rule that is local, i.e. it relies on the predictive distribution exclusively through the observed outcome x as opposed to accounting for probabilities of other potential events that did not materialize (Bernardo 1979; Gneiting and Raftery 2007; Dawid and Musio 2014; Tyralis and Papacharalampous 2024). Following this, one general disadvantage of local scores is that they do not take the distance of the prediction to the materialization into account (Gneiting and Raftery 2007). In contrast to our desired scoring rule properties, larger values of the logarithmic score indicate a better score, hence a common modification is to select the negatively oriented negative logarithmic likelihood (NLL), i.e. $-\log p(x)$, instead. The negative logarithmic score can deteriorate towards $+\infty$ if $p(x) = 0$ (Bracher et al. 2021), thus it heavily penalizes forecasts that assign (near) zero probability to observed outcomes. Furthermore, the NLL lacks robustness and is restricted to predictive densities, which is often an impractical limitation (Gneiting and Raftery 2007). On the left of Figure 3.1 the local behavior of the positively oriented logS is shown visually. Here, the green area represents a predictive distribution for the variable x. The actual observed value for x is marked by the black dashed line and the resulting logS value is highlighted by the blue arrow. This visualizes the local property of the logS since it only depends on the probability assigned to the materialized value of x. Therefore, the logS reaches its optimal value of 0 when the predictive distribution assigns a probability of $p(x) = 1$ to the actual realized value of x.

In contrast to the logS, the continuous ranked probability score (CRPS) is often considered a more robust proper scoring rule alternative. Furthermore, it is strictly proper for distributions with a finite first moment (Gneiting et al. 2007). The CRPS can be defined as

$$\text{CRPS}(\hat{F}, x) = \int_{-\infty}^{\infty} \{\hat{F}(y) - \mathbb{1}(y \geq x)\}^2 \, dy \tag{3.4}$$

where $\hat{F}$ is a predictive cumulative distribution function (CDF) and $\mathbb{1}(y \geq x)$ is the indicator function that is one if $y \geq x$ and zero otherwise (Bracher et al. 2021; Matheson and Winkler 1976). Since the CRPS is defined in terms of the predictive CDF $\hat{F}$,

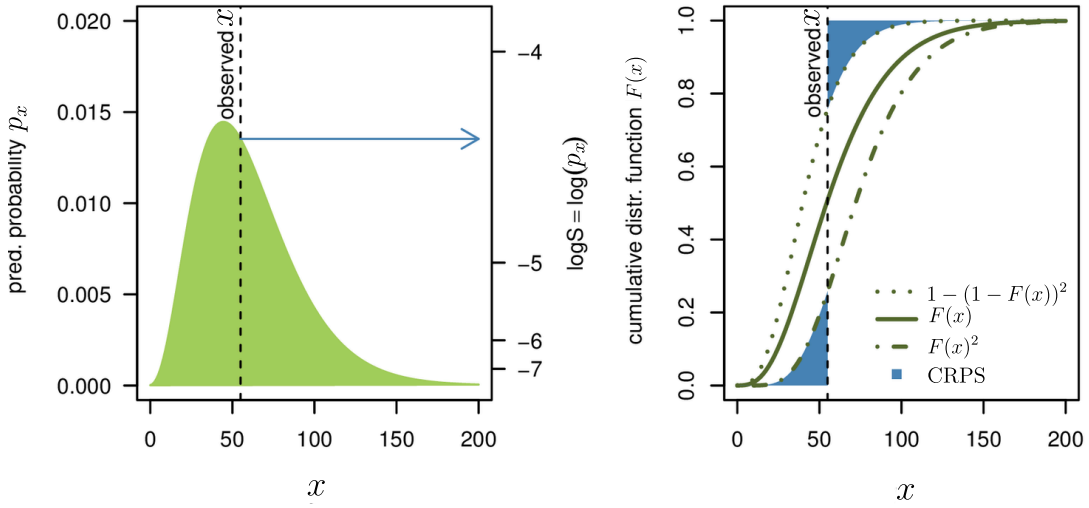


Figure 3.1.: Adapted from Bracher et al. (2021), Page 3/Figure 1. Visualization of the logS (left) and CRPS (right).

there are different possible approximations using samples of the predictive distribution, in which the accuracy of the approximation is closely tied to the number of samples (Gneiting and Raftery 2007; Koochali et al. 2022). One approach is to replace an exact expression of the CDF $\hat{F}$ with an empirical estimate $\hat{F}_E = \frac{1}{n}\sum_{i=1}^{n} I(y \geq x_i)$, which is computed based on n samples, in Equation 3.4. Alternatively, a computation based on the quantile score (QS_α), to be introduced in Equation 3.15 in Section 3.3.2, is defined as

$$\text{CRPS}(\hat{F}, x) = \int_0^1 QS_\alpha(\hat{F}^{-1}(\alpha), x)\, d\alpha \tag{3.5}$$

with the quantile function $\hat{F}^{-1}$ (Salinas et al. 2019). Using samples, the quantile levels α can be estimated, such that the precision of the approximation now depends on the number of samples n as well as the number of quantiles α. Lastly, the integral in Equation 3.4 can also be evaluated in closed form with a kernel score representation:

$$\text{CRPS}(\hat{F}, x) = \mathbb{E}_{\hat{X}\sim\hat{F}}|\hat{X} - x| - \frac{1}{2}\mathbb{E}_{\hat{X},\hat{X}'\sim\hat{F}}|\hat{X} - \hat{X}'| \tag{3.6}$$

where $\hat{X}$ and $\hat{X}'$ are independent copies of a random variable with distribution function $\hat{F}$ and finite first moment (Gneiting and Raftery 2007). In Equation 3.6 the CRPS is represented in the same unit as the observations, and it generalizes to the absolute error in the setting of point forecasts (Gneiting and Katzfuss 2014). On the right of Figure 3.1, the CRPS is visualized. The dark green line corresponds to the predictive CDF $\hat{F}(x)$, while the materialized value x is marked by the black dotted line. The blue-shaded areas depict the components that are integrated to compute the final CRPS value. The left shaded area captures the integral under the $\hat{F}(x)^2$ curve up to the observed value x, i.e. $y < x$ in Equation 3.4. Furthermore, this area penalizes probability mass assigned to

values smaller than the observation. The right shaded area begins at x and represents the region above the dotted $1-(1-\hat{F}(x))^2$ curve. While Equation 3.4 calculates the area under the $(\hat{F}(x)-1)^2$ curve for $y \geq x$, the figure shows the equivalent region above the corresponding inverted curve. This part penalizes probability mass assigned to values greater than or equal to x. Similar to the logS, the optimal CRPS value is reached if we produce a predictive distribution that assigns all probability mass to the observed value of x.

With CRPS-Sum, Salinas et al. (2019) introduced an extension of the CRPS to multivariate data. Instead of averaging the CRPS values of univariate distributions over time steps and series, they account for joint effects by summing across series first:

$$\text{CRPS-Sum}(\hat{F}, x) = \mathbb{E}_t[\text{CRPS}(\hat{F}, \sum_i x_t^{(i)})] \tag{3.7}$$

where $\hat{F}^{-1}$ is obtained by first summing samples across series i and then estimating the CDF. Koochali et al. (2022) investigated the discrimination ability of CRPS-Sum. They highlight that the score is more sensitive to changes in the covariance of the model when the covariance of the data is negative, limiting the interpretation of CRPS-Sum scores. In conclusion, the authors suggest avoiding the CRPS-Sum. Despite that, numerous studies still integrate the CRPS-Sum as their primary multivariate metric (Feng et al. 2024; Zheng and Sun 2024; Zhang et al. 2024; Feng et al. 2024).

Since the CRPS is only defined for univariate predictive distributions, Gneiting and Raftery (2007) introduce the energy score (ES), a generalization to the multivariate case:

$$\text{ES}(\hat{F}, x) = \mathbb{E}_{\hat{X}\sim\hat{F}}||\hat{X} - x||_p^\beta - \frac{1}{2}\mathbb{E}_{\hat{X},\hat{X}'\sim\hat{F}}||\hat{X} - \hat{X}'||_p^\beta \tag{3.8}$$

where $||\cdot||_p$ refers to the L_p-norm. $\hat{X}$ and $\hat{X}'$ are independent copies of a random variable and follow a distribution of $\hat{F} \in \mathcal{F}_\beta$. $\mathcal{F}_\beta$ with $\beta \in (0, 2)$ represents the class of all Borel probability measures P on $\mathbb{R}^m$ that ensure $\mathbb{E}_P||\hat{X}||_p^\beta$ is finite (Gneiting and Raftery 2007). For $\beta = 1$ and $m = 1$ this returns the CRPS. The first term in Equation 3.8 evaluates the quality of individual predictions, whereas the second term assesses diversity (Shahroudi et al. 2024).

In a simulation study to investigate the discriminative ability of the energy score, Pinson and Tastu (2013) found that the energy score cannot distinguish dependency structure differences of bivariate normal distributions well. Scheuerer and Hamill (2015) came to similar results in another simulation study, therefore they studied multivariate proper alternatives to the energy score that better capture the correlation structure. Furthermore, Scheuerer and Hamill (2015) introduce the variogram score, which is based on variograms from geostatistics (Matheron 1963; Cressie 1985) and takes the pairwise differences between the components of the multivariate quantity of interest into account.

Given forecast distribution $\hat{F}$ and H-variate observation vector x, e.g. future observations for forecasting horizon H, the variogram score of order p is defined as:

$$\mathrm{VS}_p(\hat{F}, x) = \sum_{t_1}^{H} \sum_{t_2}^{H} w_{t_1,t_2} \Big(|x_{t_1} - x_{t_2}|^p - \mathbb{E}_{\hat{X} \sim \hat{F}}(|\hat{X}_{t_1} - \hat{X}_{t_1}|^p) \Big)^2 \tag{3.9}$$

with $p > 0$, often p is either equal to $0.5, 1$ or 2 in practice (Alexander et al. 2024). $\hat{X}_{t_1}$ and $\hat{X}_{t_2}$ are the t_1-th and t_2-th components of a random vector $\hat{X}$ with distribution $\hat{F}$. $|x_{t_1} - x_{t_2}|$ is the absolute difference of the multivariate observation and w_{t_1,t_2} are non-negative weights that are often ignored by setting them to $w_{t_1,t_2} = 1 \; \forall t_1, t_2$ (Ziel and Berk 2019; Alexander et al. 2024). However, certain weighting schemes can help mitigate the effect of sampling errors. For example, in time lag situations, pairs with expected weak correlations, e.g. steps that are far apart, can be downweighted by scaling the weights proportional to the inverse time distance (Scheuerer and Hamill 2015). Although Scheuerer and Hamill (2015) show that VS_p is proper relative to the class of distributions for which the $2p$-th moments of all elements are finite, it is not strictly proper (Alexander et al. 2024). Furthermore, the VS_p does not detect a bias present across all components, as it is solely defined by the pairwise differences, meaning the bias cancels out (Alexander et al. 2024).

Over the years, several studies have examined the discrimination ability of proper multivariate scoring rules. While Pinson and Tastu (2013) and Scheuerer and Hamill (2015) find that the energy score cannot detect forecasting errors with respect to the dependency structure, Ziel and Berk (2019) produce results that suggest the opposite. In another study, Alexander et al. (2024) propose that the variogram score with $p = 0.5$ outperforms the energy score and variogram scores with $p = 1$ and $p = 2$ in terms of discrimination ability. Whereas Shahroudi et al. (2024) argued that the energy score is an adequate score for trajectory projection evaluation, which share a high similarity to multivariate time series. However, they also note that the calculation of the energy score has a quadratic complexity $O(K^2)$ in the number of samples K. In the last few years, Marcotte et al. (2023) published an important work which reveals that although the CRPS, ES and VS have good asymptotic properties, they are unreliable compared to the NLL in (smaller) finite sample regions. Furthermore, they criticize that many recent studies (Salinas et al. 2019; Rasul et al. 2020; de Bézenac et al. 2020; Rasul et al. 2021; Tashiro et al. 2021; Tang and Matteson 2021; Drouin et al. 2022) perform experiments across data sets in which the dimensionality of the data is much bigger than the sample size and number of evaluation windows, resulting in regions where the scoring rules are generally unreliable compared to the NLL (Marcotte et al. 2023). Altogether, the different results motivate the use of qualitative evaluation techniques while simultaneously comparing the results of multiple proper scoring rules (Alexander et al. 2024; Koochali et al. 2022). Furthermore, by considering synthetic experiments, where the true underlying distribution P is known, it becomes feasible to directly quantify the distance between the predictive distribution $\hat{F}$ and P. A natural choice for this comparison is the Kullback–Leibler (KL) divergence (Kullback and Leibler 1951; Kullback 1997), defined

as

$$D_{KL}(P \parallel \hat{F}) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{\hat{f}(x)} \tag{3.10}$$

where P and $\hat{F}$ are assumed to be distributions over a continuous random variable, while p and $\hat{f}$ correspond to the densities of P and $\hat{F}$ (Bishop 2019). Not only does this allow for a reliable comparison between forecast and true distributions, it also enables us to assess whether the conclusions drawn from potentially non-reliable scores, e.g. VS or ES, align with those from KL-based evaluations. However, in settings where the true distribution is unknown, KL divergence must be approximated, often through complex or data-intensive methods (Ghimire et al. 2021; Nguyen et al. 2010). Additionally, the unreliability of previously discussed scoring rules, further motivates the need to evaluate specific forecast properties, such as sharpness and calibration, independently. Hence, the following chapter deals with scoring rules designed for quantile and interval forecasts, including metrics for individual properties.

3.3.2. Scoring Rules for Quantile and Interval Forecasts

Although it is common in probabilistic forecasting to predict and output full probability distributions, some methods, e.g. quantile regression (Koenker and Bassett 1978; Koenker 2005), produce direct quantile or interval forecasts. While it is possible to approximate previously mentioned scores, the approximation of some scores, e.g. NLL, may lead to problems when observations fall into the tails of the predictive distribution (Bracher et al. 2021). On top of that, the upcoming quantile and interval scores enable comparison between arbitrary forecasters, which issue sample predictions, and quantile forecasters. Furthermore, Bracher et al. (2021) argue that specifically designed scoring rules for quantile or interval forecasts are often favorable instead.

Beginning with metrics to summarize certain properties of quantile and interval forecasts individually, we can assess calibration by determining the empirical coverage (Christoffersen 1998; Gneiting et al. 2007) defined as

$$\text{Coverage}_{\alpha}(x, \hat{x}) = \frac{1}{H} \sum_{t}^{H} \mathbb{1}(\hat{x}_t^{(\alpha)} \geq x_t) \tag{3.11}$$

where $\hat{x}^{(\alpha)}$ are the forecasts for quantile level α and x is the materialization. Ideally, the coverage for the nominal quantile level α is equal to α, e.g. regarding the 90% quantile forecast we would like to cover 90% of the true observed values (Dheur and Taieb 2023). To jointly evaluate the coverage across multiple quantile levels, Quantile-Quantile (Q-Q) plots (Wilk and Gnanadesikan 1968; Almeida et al. 2018) offer a visual comparison between the predicted quantiles and the empirical quantiles of the observed data. Furthermore, a Q-Q plot places the nominal quantile levels on the x-axis and the corresponding empirical coverage on the y-axis, for example see the left panel of Figure 5.5 in Section 5.2. If the quantile forecasts are well-calibrated, the points will lie closely to the bottom left to top right diagonal. Sharpness, on the other hand, is

an intrinsic property of the forecasts and reflects the concentration or spread of the predictive distribution (Tyralis and Papacharalampous 2024). To assess sharpness, we use both graphical (e.g., prediction interval plots) and numerical summaries of the width of prediction intervals (Gneiting et al. 2007). A quantitative measure of sharpness, as proposed by Bergsma et al. (2022), evaluates the average normalized width of the $[\alpha_1, \alpha_2]$ predictive interval. It is defined as:

$$\text{Sharpness}_{\alpha_1,\alpha_2}(x, \hat{x}) = \sum_{t}^{H} \frac{|\hat{x}_t^{(\alpha_1)} - \hat{x}_t^{(\alpha_2)}|}{|x_t|} \tag{3.12}$$

where the width of the interval is normalized by the magnitude of the true observation $|x_t|$, ensuring comparability across varying scales.

While calibration and sharpness highlight specific aspects of forecast quality, they do not offer a unified evaluation. To address this, Gneiting (2011) introduce a decision-theoretic framework based on statistical functionals, enabling the development of scoring rules that are consistent for targets like quantiles or intervals. These rules allow joint assessment of key properties, such as calibration and sharpness, within a single, coherent metric. A functional describes a potentially set-valued mapping $T : \mathcal{F} \to D$ from a class of probability distributions $\mathcal{F}$ to Euclidean space $D \subseteq \mathbb{R}^d$ (Horowitz and Manski 2006; Huber and Ronchetti 2011), D is often assumed to be the real line $\mathbb{R}$, e.g. the mean functional, quantiles or expectiles (Gneiting and Katzfuss 2014). A scoring rule S is consistent (Murphy and Daan 1985) for the functional T relative to the class $\mathcal{F}$ if for all distributions $\hat{F} \in \mathcal{F}$ and all $t \in T(\hat{F})$

$$\mathbb{E}_{\hat{F}}[S(t, \hat{X})] \leq \mathbb{E}_{\hat{F}}[S(x, \hat{X})], \quad \forall x \in D \tag{3.13}$$

holds, with $\hat{X}$ following $\hat{F}$ (Gneiting 2011). S is strictly consistent when equality in Equation 3.13 implies $x \in T(\hat{F})$. In the context of point forecasts, consistent scoring functions are a special case of proper scoring rules (Gneiting and Katzfuss 2014). Moreover, such functions induce proper scoring rules through a simple and natural construction according to the following theorem from Gneiting (2011):

Theorem 1 (adapted from Gneiting (2011), see Appendix A in Gneiting (2011) for proof)**.** *Given a scoring function s that is consistent for the functional T relative to the class of all probability measures $\mathcal{F}$ on $D \subseteq \mathbb{R}^d$. Then $S(\hat{F}, x) = s(T(\hat{F}), x)$ is a proper scoring rule relative to the class $\mathcal{F}$ with $\hat{F} \in \mathcal{F}$.*

If there exists a strictly consistent scoring function for the functional T relative to $\mathcal{F}$, then T is called elicitable relative to $\mathcal{F}$ (Lambert et al. 2008; Osband and Reichelstein 1985; Gneiting and Katzfuss 2014). For example, the mean functional is elicitable relative to the class of probability distributions on $\mathbb{R}$ with finite second moments, since the squared error $S(x, \hat{x}) = (x - \hat{x})^2$ is strictly consistent in that setting (Gneiting and Katzfuss 2014). Similarly, the α-quantile functional is elicitable subject to regularity conditions (Gneiting 2011; Gneiting and Katzfuss 2014). Moreover, a scoring rule is

only consistent with regard to the α-quantile functional if that rule is expressed in the form of a generalized piecewise linear function of order α as follows:

$$s(x, \hat{x}) = \big(\alpha - \mathbb{1}(x > \hat{x})\big)\big(g(\hat{x}) - g(x)\big) \tag{3.14}$$

with a non-decreasing g (Gneiting 2011; Thomson 1979). The rule is strictly consistent, if g is strictly increasing. When setting $g(z) = z$ (and multiplying by 2) in Equation 3.14, we receive the quantile score (QS), also known as pinball loss or asymmetric piecewise linear scoring function (Koenker and Bassett 1978; Gneiting et al. 2007). For the α-quantile prediction $\hat{x}^\alpha$ and materialization x, it is defined as:

$$\mathrm{QS}_\alpha(\hat{x}, x) = 2 \cdot \big(\alpha - \mathbb{1}(\hat{x}^\alpha > x)\big)\big(x - \hat{x}^\alpha\big) \tag{3.15}$$

The QS approximates the absolute error up to a constant factor, when $\alpha = 0.5$ (Gneiting and Katzfuss 2014). Furthermore, it is a common score used to train and evaluate quantile regression models (Koenker and Bassett 1978; Koenker 2005). To evaluate the overall performance of quantile forecasts across multiple nominal levels, several studies (Gasthaus et al. 2019; Chen et al. 2019; Bergsma et al. 2022; Sprangers et al. 2023) propose a weighted quantile score (wQS). This metric aggregates the quantile scores over a set of quantile levels Q, allowing for a unified assessment rather than evaluating each level in isolation. Formally, the weighted quantile score is defined as:

$$wQS(x, \hat{x}) = \frac{1}{Q} \sum_{\alpha \in Q} \frac{\sum_t^H \mathrm{QS}_\alpha(\hat{x}_t, x_t)}{\sum_t^H |x_t|} \tag{3.16}$$

where the normalization with $\sum_t^H |x_t|$ ensures comparability across different scales or datasets. Since we consider standard central prediction intervals (PIs) throughout this thesis, in which $(1 - \alpha)$ is the nominal coverage rate and the lower and upper bound correspond to the $(\alpha/2)$ and $(1-\alpha/2)$ predictive quantiles respectively, transitioning from quantile to interval forecasts is straightforward. In fact, the QS can be used to construct proper scoring rules for interval forecasts (Gneiting and Katzfuss 2014; Cervera and Muñoz 1996), for instance recall the CRPS approximation using the QS in Equation 3.5. Furthermore, given the quantile scores at the $(\alpha/2)$ and $(1 - \alpha/2)$ level we can compute the interval score (IS) (Winkler 1972) for the central $(1 - \alpha)$ prediction interval based on the QS as follows (Gneiting and Raftery 2007; Bracher et al. 2021):

$$IS_\alpha(\hat{F}, x) = \frac{QS_{\alpha/2}(\hat{x}, x) + QS_{1-\alpha/2}(\hat{x}, x)}{\alpha} \tag{3.17}$$

where $\hat{x}$ encompasses $\hat{x}^{\alpha/2}$ and $\hat{x}^{1-\alpha/2}$, which are the corresponding predictions for the $(\alpha/2)$- and $(1-\alpha/2)$-quantile level of forecast $\hat{F}$. Alternatively the IS may be expressed as:

$$IS_\alpha(\hat{F}, x) = (\hat{u} - \hat{l}) + \frac{2}{\alpha} \cdot (\hat{l} - x) \cdot \mathbb{1}(x < \hat{l}) + \frac{2}{\alpha} \cdot (x - \hat{u}) \cdot \mathbb{1}(x > \hat{u}) \tag{3.18}$$

where $\hat{l} = \hat{x}^{\alpha/2}$ and $\hat{u} = \hat{x}^{1-\alpha/2}$ represent the lower and upper bound of the predictive interval respectively. Besides being a proper score, Bracher et al. (2021) explain three

simple yet important concepts of the IS. First, it uses the length of the $(1-\alpha)$ prediction interval via $(\hat{u}-\hat{l})$, thus the sharpness of the predictive distribution is included. Second, through the implementation of the indicator function $\mathbb{1}(x < \hat{l})$ the observations that are below the lower bound will be penalized. Furthermore, the effective penalty value is dependent upon the magnitude of error $(\hat{l} - x)$ as well as the level of α, where the penalty increases in value for high $(1 - \alpha)$ values. Third, a similar term penalizes observations that are above the upper bound of the prediction interval. Although the IS is an adequate proper scoring rule for interval forecasts, Bracher et al. (2021) describe that the general task is to provide forecasts for multiple distinct levels of prediction intervals $(1 - \alpha_1) < (1 - \alpha_2) < \ldots < (1 - \alpha_K)$. In addition, one is also often tasked to report the predictive median $m = \hat{x}^{\alpha=0.5}$, which can be seen as the $(1 - \alpha_0)$ prediction interval. Thus, Bracher et al. (2021) introduce the weighted interval score (WIS):

$$WIS_{\alpha_{0:K}}(\hat{F}, x) = \frac{1}{K + 0.5} \cdot \left(w_0 \cdot |x - \hat{m}| + \sum_{k=1}^{K} \{w_k \cdot IS_\alpha(\hat{F}, x)\} \right) \tag{3.19}$$

with weights w_i for the PI levels $i = 1, \ldots, K$. If all weights are non-negative and unnormalized, the WIS is proper. Moreover, Bracher et al. (2021) propose that the following selection of weights:

$$w_k = \begin{cases} 1/2 & \text{for } k = 0 \\ \alpha_k/2 & \text{for } k > 1 \end{cases} \tag{3.20}$$

is natural and leads to an approximation of the CRPS by the WIS, when given a wealth of equally spaced levels $\alpha_1, \alpha_2, \ldots, \alpha_K$. Based on the alternative formulation of the IS in Equation 3.17, the WIS is representable through the QS as well (Bracher et al. 2021).

To summarize the discussed evaluation metrics, Table 3.3 provides an overview of the covered scoring rules, highlighting their properties regarding properness and multivariate applicability. Notably, multivariate scoring rules are either unreliable in some contexts, e.g. Energy Score and Variogram Score, or require specific prerequisites, for instance NLL is only applicable if a density forecast is provided and KL divergence requires access to the ground truth distribution. Given these limitations, we will complement the general LTSF setup with evaluations on synthetic examples while also separately analyzing the individual properties of calibration and sharpness. For model training, we adopt the NLL due to its reliability (Marcotte et al. 2023), while quantile-based methods are trained using the Quantile Score (QS). The next chapter introduces the methodological foundation of this work, beginning with the backbone LTSF models, which we aim to extend into the probabilistic domain.

Table 3.3.: Summary of probabilistic scoring rules used in forecasting evaluation. A reference to the formula and common attributes, propriety and ability to evaluate multivariate dependencies, are shown. Multivariate compatibility is indicated with parentheses (✓) where applicability depends on model or distributional assumptions (e.g., NLL, KL divergence).

Score	**Definition**	**Proper**	**Multivariate**
NLL	$-\log\mathrm{S}(F, x) = -\log p(x)$	✓	(✓)
CRPS	3.4, 3.5, 3.6	✓	
CRPS_Sum	3.7	✓	
Energy Score	3.8	✓	✓
Variogram Score	3.9	✓	✓
KL divergence	3.10		(✓)
Coverage	3.11		
Sharpness	3.12		
Quantile Score	3.16	✓	
Weighted Quantile Score	3.15	✓	
WIS	3.19	✓	

4. Methods

This chapter focuses on enabling probabilistic LTSF using a range of deep learning models, while also introducing *single-* and *multi-world* scenarios. We begin by introducing the backbone forecasting architectures, which form the basis for our probabilistic extensions. These include both IMS and DMS strategies, spanning recurrent, MLP-based, and Transformer-based models. Building on these architectures, we then describe how to adapt their outputs to support probabilistic forecasting. This is achieved through modular prediction heads that allow for the estimation of uncertainty via distributional or quantile-based approaches. Finally, we present a formal distinction between *single-world* and *multi-world* scenarios and describe how IMS and DMS models handle them differently.

4.1. Backbone LTSF Models

In this section, we present the backbone models used for LTSF, which we will later modify for the probabilistic evaluation. To streamline model integration and ensure consistency in training and evaluation, we adopt the BasicTS+ (Shao et al. 2025) benchmark framework, hence highlighting specific implementation details and extensions provided by BasicTS+. Although BasicTS+ includes over 30 models, we narrow our focus to three representative architectures: PatchTST (Nie et al. 2022), DLinear (Zeng et al. 2023), and DeepAR (Salinas et al. 2020). Importantly, our probabilistic extensions are fully integrated into the BasicTS+ pipeline, allowing seamless probabilistic evaluation of any supported model with minimal modifications. We include DeepAR as it is the only IMS model available in BasicTS+ and it has been a foundational architecture in probabilistic TSF, inspiring numerous subsequent methods (Rasul et al. 2022, 2021, 2020; Bergsma et al. 2022; Kan et al. 2022). DLinear serves as a representative example of a simple MLP-based DMS model, notable for challenging the performance dominance of prior more complex SOTA models. Finally, PatchTST is the most recent among the selected models and was chosen to reflect the strength of Transformer-based architectures, particularly those leveraging patching mechanisms, in modern point LTSF.

DLinear. Figure 4.1 depicts the architecture of the DLinear model (Zeng et al. 2023). First, since DLinear is a CI model, the input is split up into the individual channels $x^{(i)} \in \mathbb{R}^{1\times L}$. Then, DLinear implements a decomposition scheme (Cleveland et al. 1990), inspired by Autoformer (Wu et al. 2021) and FEDformer (Zhou et al. 2022), in which a moving average kernel decomposes $x^{(i)}$ into a trend component $x_t^{(i)}$ and a remaining seasonal component $x_s^{(i)}$. After that, each component is fed into its respective single-layer

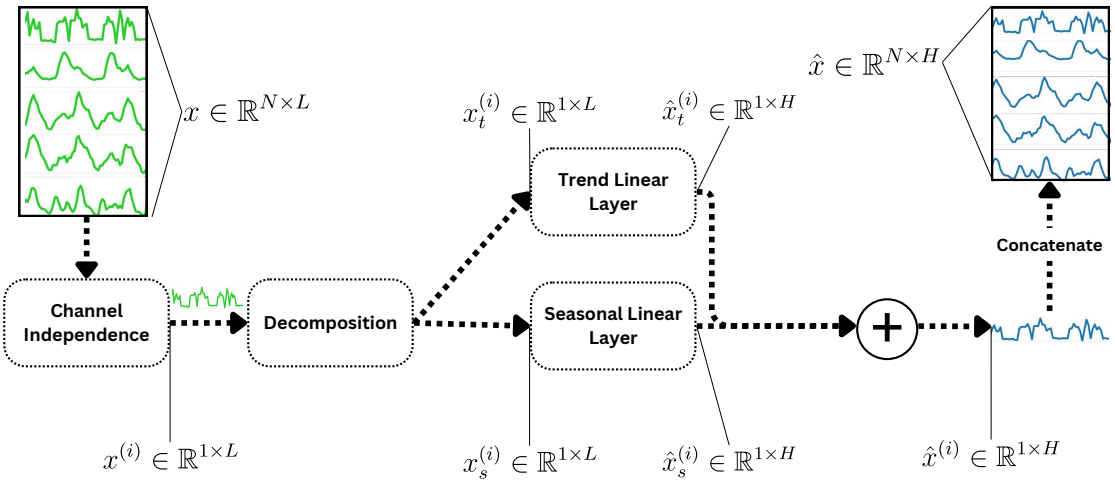


Figure 4.1.: Architecture of the DLinear model as implemented in BasicTS+. The input is decomposed into trend and seasonal components via a moving average kernel, which are then independently processed by linear layers and recombined to produce the final forecast. The implementation also supports optional channel-specific linear layers for localized modeling.

linear layer $W_s, W_t \in \mathbb{R}^{L \times H}$ to produce $\hat{x}_s, \hat{x}_t \in \mathbb{R}^{1 \times H}$. Next, both $\hat{x}_s$ and $\hat{x}_t$ are simply added back together, resulting in the final prediction $\hat{x}^{(i)} \in \mathbb{R}^{1 \times H}$ for time series i. In the end, all predictions of individual series are concatenated to return the multi-channel prediction $\hat{x} \in \mathbb{R}^{N \times H}$. In addition to the general model formulation, BasicTS+ provides an option to assign each time series its own set of linear layers, effectively creating N local models without parameter sharing.

PatchTST. Figure 4.2 depicts the blueprint of the PatchTST model (Nie et al. 2022). On the left, the multi-channel input x is first processed in a channel-independent way, i.e. divided into N single-channel time series. Then, Nie et al. (2022) adopt an optional reversible instance normalization (RevIN) approach (Kim et al. 2021) to circumvent the distribution shift problem often encountered in time series, i.e. statistical properties and data distribution change over time (Chen et al. 2024). Note that, in BasicTS+ RevIN is integrated into the architecture of specific models. Consequently, unlike the scaling operations discussed in the upcoming Section 4.2, it is not universally available across all models. RevIN performs both normalization and de-normalization as follows.

$$\begin{aligned} \text{Normalization} : \tilde{x}^{(i)} &= \frac{x^{(i)} - \tilde{\mu}^{(i)}}{\tilde{\sigma}^{(i)}} \cdot \gamma^{(i)} + \beta^{(i)} \\ \text{De-Normalization} : \hat{x}^{(i)} &= \left(\frac{\hat{x}^{(i)} - \beta^{(i)}}{\gamma^{(i)}} \right) \cdot \tilde{\sigma}^{(i)} + \tilde{\mu}^{(i)} \end{aligned} \tag{4.1}$$

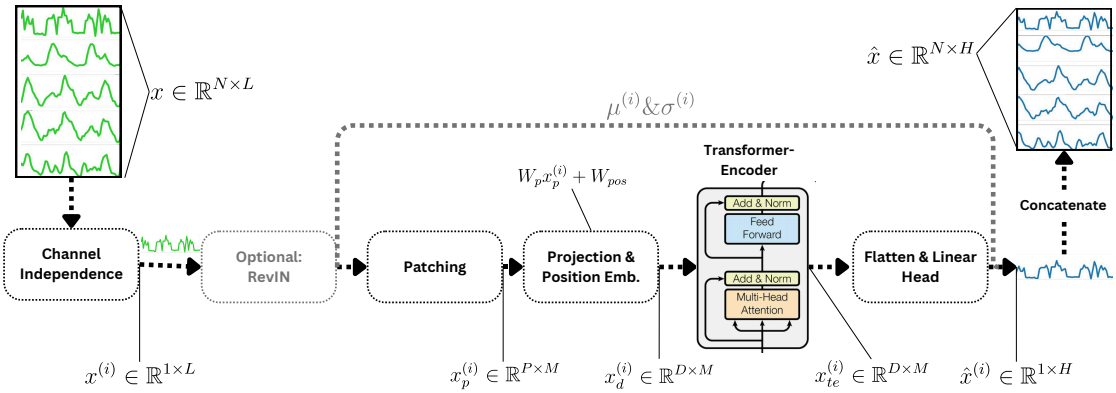


Figure 4.2.: Architecture of the PatchTST model as implemented in BasicTS+. The model operates in a channel-independent manner, optionally applies reversible instance normalization (RevIN), segments input sequences into fixed-length patches, embeds them with learnable position encodings, and processes them via a standard Transformer encoder.

Here, $\tilde{\mu}^{(i)} = \text{mean}(x^{(i)})$ and $\tilde{\sigma}^{(i)} = \text{std}(x^{(i)})$ are the mean and standard deviation of a single series (i.e., over the temporal dimension). Additionally, $\gamma, \beta \in \mathbb{R}^N$ are optional learnable affine parameters, which are set to $\gamma = 1$ and $\beta = 0$ if affine transformation is disabled. In the next step of Figure 4.2, the patching process generates patches $x_p^{(i)} \in \mathbb{R}^{P \times M}$ given the univariate time series $x_{1:L}^{(i)}$. This can be described as follows:

$$x_p^{(i)} = x_{(1+(p-1)\cdot S):(P+(p-1)\cdot S)}^{(i)} \qquad \forall p : 1 \leq p \leq M \tag{4.2}$$

where P denotes the patch length, while the non-overlapping shift between successive patches is represented by the stride S. Therefore, $M = \lfloor \frac{L-P}{S} \rfloor + 2$ represents the total number of patches per time series. If the patch length P is equivalent to the stride S, the patches are strictly non-overlapping. To ensure equal length among patches, the last element $x_L^{(i)}$ is typically padded S times to the end of the time series prior to the patching process. After patching, patches are projected to the embedding dimension D of the Transformer through a trainable weight matrix $W_p \in \mathbb{R}^{D \times P}$. In addition, a learnable position encoding matrix $W_{pos} \in \mathbb{R}^{D \times M}$ is added on top as well. Regarding W_{pos}, the PatchTST integration in BasicTS+ supports various positional encoding initialization and computation schemes, such as sinusoidal, linear, random or exponential positional encoding[1]. After that, PatchTST incorporates the standard vanilla Transformer encoder (Vaswani et al. 2017) followed by a flatten layer with linear head, which returns the single-channel prediction $\hat{x}^{(i)} \in \mathbb{R}^{1 \times H}$. Lastly, all predictions are concatenated into a multi-channel output $\hat{x} \in \mathbb{R}^{N \times H}$. In addition to the architecture shown in Figure 4.2, Nie et al. (2022) propose two additional optional modifications. First, the single-channel input sequence can be decomposed, as seen previously in the DLinear model, before it

[1]See https://github.com/GestaltCogTeam/BasicTS/blob/master/baselines/PatchTST/arch/revin.py.

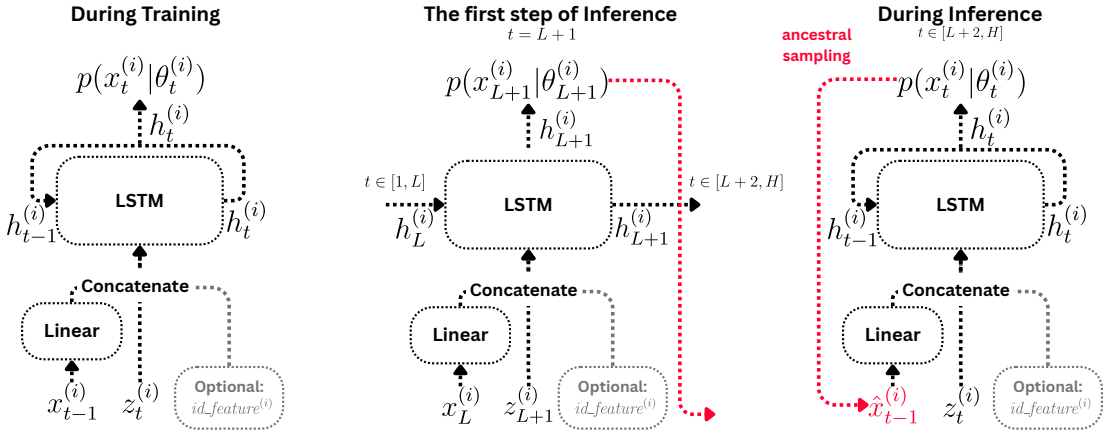


Figure 4.3.: Schematic of the DeepAR model as implemented in BasicTS+, showing training (left), the transition to inference (middle), and recursive forecasting during inference (right). During training, the model uses teacher forcing with observed values, while inference relies on ancestral sampling.

is passed to the RevIN block in Figure 4.2. For this, the series is separated into a trend component, obtained via moving average smoothing, and a residual component that captures short-term variations. Then, these components are processed independently using two instances of the PatchTST backbone and their outputs are added together to form the final single-channel prediction. Second, Nie et al. (2022) offer the option to assign a dedicated flatten and linear head to each channel, while retaining shared parameters across channels for preceding model components.

DeepAR. Figure 4.3 presents the DeepAR model (Salinas et al. 2020), a RNN-based architecture designed for probabilistic, channel-independent time series forecasting. The left side of the figure illustrates the behavior during training, which is also how it processes the input sequence during inference, i.e. time steps $t \in [1, L]$. At each time step t, the model is provided with the previous target value $x_{t-1}^{(i)}$, allowing it to condition its predictions on the observed sequence history. This approach, known as teacher forcing (Williams and Zipser 1989), encourages the model to stay aligned with the ground-truth trajectory during training. The previous target value $x_{t-1}^{(i)}$ is then embedded through a Linear layer, which is an extension introduced in BasicTS+ that was not present in the original DeepAR model. Additionally, known covariates $z_t^{(i)}$, e.g. time-of-day or day-of-week indicators, can be provided as well. Lastly, a learnable embedding $id\_feature^{(i)} \in \mathbb{R}^{D_{id\_feat}}$ can be included to represent channel-specific features, independent of time. These inputs are concatenated and passed into an RNN cell. In BasicTS+, this cell is always implemented as an LSTM cell (Hochreiter and Schmidhuber 1997). The resulting hidden state $h_t^{(i)}$, consisting of both internal LSTM states, is then mapped to the parameters of a parametric distribution $\theta(h_t^{(i)})$. Overall, training is performed

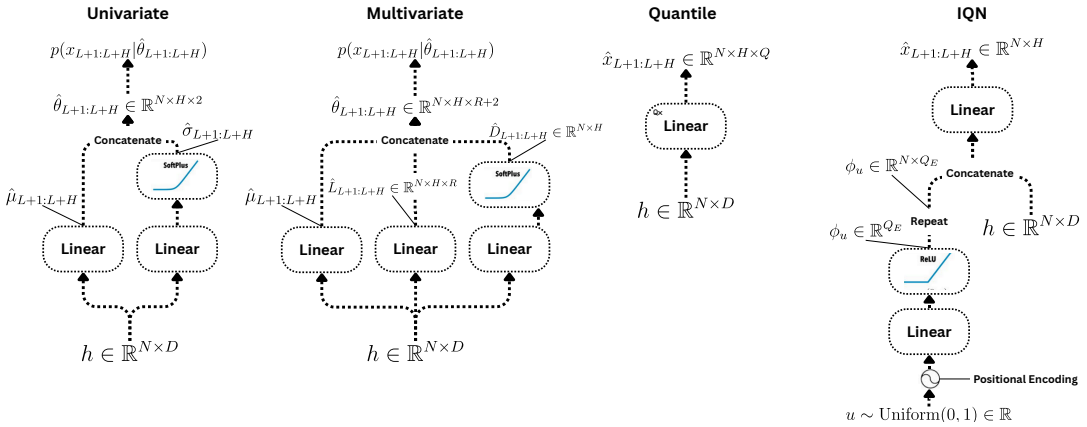


Figure 4.4.: Overview of the implemented probabilistic prediction heads. From left to right: (1) Univariate Gaussian head with separate linear layers for mean and standard deviation, using softplus to ensure positivity; (2) Quantile head that predicts a fixed set of quantiles for each time step; (3) Low-rank multivariate Gaussian head that models temporal dependencies using a low-rank plus diagonal decomposition of the covariance matrix. Each head can be used with DMS or IMS models, and adapted across different forecasting backbones.

by maximizing the log-likelihood of the observed data. Moreover, the implementation details of this distributional forecasting approach will be discussed further in Section 4.2. During inference, the model is unrolled over $t \in [1, L]$ in the same way as during training. As shown in the middle part of Figure 4.3, the last observed value $x_L^{(i)}$ and other relevant inputs are used to generate the hidden state $h_{L+1}^{(i)}$, from which the predictive distribution is created. Since the true values $x_{L+1:L+H}^{(i)}$ are not available during inference, ancestral sampling is used instead to recursively feed back sampled values, e.g. $\hat{x}_{L+1}^{(i)}$ in Figure 4.3. Note, BasicTS+ does not implement the time-series-dependent scale factor originally used in DeepAR to handle varying magnitudes across different series or channels.

4.2. Probabilistic TSF

In this section, we describe how the previously introduced LTSF backbone models can be extended into probabilistic forecasting models by modifying their prediction heads and certain scaling operations. Specifically, to reduce computational complexity, we limit the discussion to two simpler types of probabilistic heads: parametric distributional forecasting heads, trained using the negative log-likelihood loss defined in Equation 3.3, and quantile-based forecasting heads, trained with the quantile score defined in Equation 3.15. All four heads are displayed in Figure 4.4. We assume that the backbone model produces embeddings $h \in \mathbb{R}^{N \times D}$, which are then passed to the probabilistic head. In

the case of DLinear, the probabilistic head is either added after the seasonal and trend components or used to replace each of the two linear layers directly. For PatchTST and DeepAR, the probabilistic heads substitute the original prediction heads. Notably, in DeepAR, when using quantile heads, the iterative sampling process takes the median forecast instead of sample forecasts at each step. Furthermore, Figure 4.4 illustrates the setup for DMS methods, where the prediction head generates the entire forecast horizon H in a single forward pass. The same head architectures can also be used in IMS models by configuring them to output only one step at a time, achieved by setting $H = 1$ during head initialization.

Univariate head. The left side of Figure 4.4 illustrates the univariate probabilistic prediction head, shown here using a Gaussian head as an example. This architecture consists of two separate linear layers: one predicts the mean, and the other predicts the standard deviation. To guarantee that the standard deviation remains strictly positive, the output of its corresponding layer is passed through a softplus activation function, as done in (Salinas et al. 2020, 2019; Sprangers et al. 2023), which is defined as

$$\text{softplus}(x) = \log(1 + e^x) \tag{4.3}$$

Finally, the predicted parameters are concatenated to form $\hat{\theta}_{L+1:L+H} \in \mathbb{R}^{N \times H \times 2}$. Alternative univariate distributional heads follow the same structure but modify the output layers to parameterize different distributions. For instance, the Laplace head mirrors the Gaussian setup but substitutes the likelihood function. The Student's t-distribution head extends the architecture by introducing a third linear layer to estimate the degrees of freedom for each time step. This output is also passed through a softplus activation and shifted by a scalar of 1 to ensure the minimum degree of freedom is at least one.

Multivariate head. To the right of the univariate head in Figure 4.4, the multivariate Gaussian prediction head is shown. The mean is computed analogously to the univariate case via a linear projection. For the covariance, we adopt a low-rank plus diagonal decomposition (Wu et al. 2020), which drastically reduces computational complexity (especially when the rank $R \ll H$) and introduces a natural regularization effect, as the low-rank structure captures dominant modes of variability while the diagonal component accounts for residual uncertainty (similar to probabilistic PCA (Tipping and Bishop 1999)). Formally, the covariance decomposition is defined as:

$$\hat{\Sigma}^{(i)} = (\hat{L}^{(i)}\hat{L}^{(i)})^\top + \hat{D}^{(i)} \tag{4.4}$$

Here, for each time series $i = 1, \ldots, N$, the predicted distribution is modeled as a multivariate Gaussian with mean $\hat{\mu}^{(i)} \in \mathbb{R}^H$ and covariance matrix $\hat{\Sigma}^{(i)} \in \mathbb{R}^{H \times H}$. The low-rank component $\hat{L}^{(i)} \in \mathbb{R}^{H \times R}$ captures shared structure, while $\hat{D}^{(i)} \in \mathbb{R}^H$ represents the diagonal noise. The diagonal matrix is constructed by placing the softplus-activated entries of $\hat{D}^{(i)}$ along the diagonal. Altogether, this construction guarantees positive definiteness of the resulting covariance matrix. Concretely, the term $(\hat{L}^{(i)}\hat{L}^{(i)})^\top$ is a Gram

matrix and therefore positive semi-definite by definition (Horn and Johnson 2012, Observation 7.2.10). The diagonal component, obtained by applying the softplus function to $\hat{D}^{(i)}$, yields strictly positive entries and thus forms a positive-definite diagonal matrix by design (Horn and Johnson 2012). Since the sum of a positive-definite and a positive semi-definite matrix is itself positive-definite (Horn and Johnson 2012, Observation 7.1.3), the resulting covariance matrix $\hat{\Sigma}^{(i)}$ is guaranteed to be positive-definite. The parameters $\hat{\theta}_{L+1:L+H} \in \mathbb{R}^{N \times H \times R+2}$, containing the predicted mean, low-rank factors, and diagonal entries, are passed to the `LowRankMultivariateNormal` class from `torch.distributions`[2] to receive the multivariate density. In contrast to the univariate prediction head, `torch.distributions` does not support alternative low-rank multivariate distribution types, hence limiting us to the low-rank multivariate Gaussian distribution.

Quantile head. Third from the left in Figure 4.4 is the depiction of the Quantile head. In this approach, we pre-determine a fixed set of Q quantile levels. For each of these quantile levels, the backbone embeddings $h \in \mathbb{R}^{N \times D}$ are mapped independently to produce corresponding quantile forecasts. This results in an output tensor $\hat{x}_{L+1:L+H} \in \mathbb{R}^{N \times H \times Q}$, where each slice along the third dimension represents the forecast for a specific quantile level. For the quantile DeepAR model, we feed back the previously predicted median value during inference.

Implicit Quantile Network head. Lastly, the IQN prediction head, shown on the right side of Figure 4.4, is adapted from the architecture proposed by Gouttes et al. (2021) and Dabney et al. (2018). The prediction process begins by sampling a quantile level u from the `Uniform(0, 1)` distribution. This quantile level is then embedded into a representation ϕ_u using positional encoding as defined by:

$$\phi_u = \mathrm{ReLU}\Big(\mathrm{Linear}\big(\cos(u \cdot i \cdot \pi)\big)\Big) \tag{4.5}$$

where $i \in \mathbb{R}^{C_E}$ is a fixed vector of values linearly spaced from 1 to C_E, C_E is a hyperparameter controlling the dimension of the cosine embedding. The linear layer in Equation 4.5 projects from the cosine embedding dimension C_E to a quantile embedding dimension Q_E, another tunable hyperparameter. To match the number of time series instances, $\phi_u \in \mathbb{R}^{Q_E}$ is broadcast into $\phi_u \in \mathbb{R}^{N \times Q_E}$ by repeating the same embedding N times. This quantile embedding is then concatenated with the backbone representation $h \in \mathbb{R}^{N \times D}$, resulting in a combined representation of shape $\mathbb{R}^{N \times (D+Q_E)}$. A final linear projection maps this to the forecast output $\hat{x}_{L+1:L+H} \in \mathbb{R}^{N \times H}$. In addition to the concatenation approach, we optionally support the original formulation from Gouttes et al. (2021), which replaces concatenation with a Hadamard product:

$$\hat{x}_{L+1:L+H} = \mathrm{Linear}\big(h \odot (1 + \phi_u)\big) \tag{4.6}$$

Notably, this formulation requires that $Q_E = D$, which varies across model and hyperparameter choices. During training, the quantile level u is resampled for each training

[2] https://docs.pytorch.org/docs/stable/distributions.html

batch, and the sampled value is stored to compute the quantile loss. Since we are interested in forecasts that represent a consistent quantile level across all prediction steps, we sample a random quantile level per forecast horizon rather than resampling one every time step. However, the IQN DeepAR model is an exception to this, since we follow the IMS approach in Gouttes et al. (2021), by sampling a new quantile level for every time step. The reason for this is that a constant quantile level led to divergent predictions, as shown in the Appendix C in Figure C.5.

Scaling transformations for distributional parameters. Scaling operations play a central role in time series modeling pipelines, due to varying scales across series, e.g. power law distribution in retail sales series (Salinas et al. 2020, 2019), and distribution shifts across time (Fan et al. 2023; Zhang et al. 2024; Kim et al. 2021; Nie et al. 2022). In the BasicTS+ benchmark two commonly used scaling operations, Z-score normalization and Min-Max normalization, are provided. These transformations are implemented such that they surround the model, i.e. scaling is applied before the model is called and inverse transformations (de-scaling) are performed after model outputs are obtained to restore predictions to their original scale. We always apply inverse-scaling to ensure model outputs and resulting losses as well as scores are directly comparable. The scaling operation is first applied to the input batch $x_{1:L} \in \mathbb{R}^{B \times L \times N}$, with batch size B, input sequence length L, and N number of series. For Z-score normalization, the transformation is defined as:

$$\tilde{x}_{1:L} = \frac{x_{1:L} - \tilde{\mu}}{\tilde{\sigma}}, \tag{4.7}$$

where $\tilde{\mu} \in \mathbb{R}^N$ and $\tilde{\sigma} \in \mathbb{R}^N$ are the mean and standard deviation per series computed over the entire train set. Note that BasicTS+ also allows using a single scalar mean and standard deviation shared across all series, i.e. $\tilde{\mu} \in \mathbb{R}$ and $\tilde{\sigma} \in \mathbb{R}$. However, the upcoming re-scaling computations remain formally identical, as these scalars can be interpreted as constant vectors of dimension N, yielding the same expressions. For Min-Max normalization, the transformation is:

$$\tilde{x}_{1:L} = \frac{x_{1:L} - \tilde{x}_{\min}}{\tilde{x}_{\max} - \tilde{x}_{\min}}, \tag{4.8}$$

where $\tilde{x}_{\min}, \tilde{x}_{\max} \in \mathbb{R}^N$ are the minimum and maximum values per series over the train set, ultimately scaling values into the $[0, 1]$ range. In addition to this, the RevIN operation (Kim et al. 2021) is available for the PatchTST model. Unlike the other two scaling methods, RevIN computes the parameters $\tilde{\mu}^{(i)}, \tilde{\sigma}^{(i)}, \gamma, \beta \in \mathbb{R}^N$ per series and per batch, as shown in Equation 4.1. After the model produces outputs in the scaled space, these outputs must be mapped back. While the re-scaling operation for point and quantile forecasts is straightforward, parametric distributional forecasting methods have to be handled differently. For univariate distributions with location and scale parameters, such as Gaussian outputs with predicted mean $\hat{\mu} \in \mathbb{R}^{B \times H \times N}$ and standard deviation $\hat{\sigma} \in \mathbb{R}^{B \times H \times N}$, this descaling is:

- **Z-score de-normalization:**

$$\mu = \hat{\mu} \cdot \tilde{\sigma} + \tilde{\mu}, \quad \sigma = \hat{\sigma} \cdot \tilde{\sigma} \tag{4.9}$$

- **Min-Max de-normalization:**

$$\mu = \hat{\mu} \cdot R + \tilde{x}_{\min}, \quad \sigma = \hat{\sigma} \cdot R, \quad \text{where } R = \tilde{x}_{\max} - \tilde{x}_{\min} \tag{4.10}$$

- **RevIN de-normalization:** (following normalization as in Equation 4.1)

$$\mu^{(i)} = \left(\frac{\hat{\mu}^{(i)} - \beta^{(i)}}{\gamma^{(i)}} \right) \cdot \tilde{\sigma}^{(i)} + \tilde{\mu}^{(i)}, \quad \sigma^{(i)} = \hat{\sigma}^{(i)} \cdot \frac{\tilde{\sigma}^{(i)}}{\gamma^{(i)}} \tag{4.11}$$

In the multivariate setting, we assume the model produces forecasts for each of the N series independently over a forecast horizon of length H. Concretely, for each series $i = 1, \ldots, N$, the predicted distribution is a multivariate Gaussian with parameters $\hat{\mu}^{(i)} \in \mathbb{R}^H$ and covariance $\hat{\Sigma}^{(i)} = (\hat{L}^{(i)}\hat{L}^{(i)})^\top + \hat{D}^{(i)}$, where $\hat{L}^{(i)} \in \mathbb{R}^{H \times r}$ and $\hat{D}^{(i)} \in \mathbb{R}^{H \times H}$ is diagonal. The de-normalization is applied per series, using the original scaling parameters $\tilde{\mu}^{(i)}$, $\tilde{\sigma}^{(i)}$, or $\tilde{x}^{(i)}_{\min}$, $\tilde{x}^{(i)}_{\max}$ of each series.

- **Z-score de-normalization:**

$$\mu^{(i)} = \hat{\mu}^{(i)} \cdot \tilde{\sigma}^{(i)} + \tilde{\mu}^{(i)}, \tag{4.12}$$

$$\Sigma^{(i)} = (\tilde{\sigma}^{(i)})^2 \cdot \hat{\Sigma}^{(i)} = (\tilde{\sigma}^{(i)})^2 \cdot \left((\hat{L}^{(i)}\hat{L}^{(i)})^\top + \hat{D}^{(i)} \right) \tag{4.13}$$

- **Min-Max de-normalization:** Let $R^{(i)} = \tilde{x}^{(i)}_{\max} - \tilde{x}^{(i)}_{\min}$, then simply substitute $\tilde{\mu}^{(i)}$ and $(\tilde{\sigma}^{(i)})^2$ in Equation 4.12 and 4.13 by $\tilde{x}^{(i)}_{\min}$ and $(R^{(i)})^2$ respectively.

- **RevIN de-normalization:** (following normalization as in Equation 4.1)

$$\mu^{(i)} = \left(\frac{\hat{\mu}^{(i)} - \beta^{(i)}}{\gamma^{(i)}} \right) \cdot \tilde{\sigma}^{(i)} + \tilde{\mu}^{(i)}, \tag{4.14}$$

$$\Sigma^{(i)} = \left(\frac{\tilde{\sigma}^{(i)}}{\gamma^{(i)}} \right)^2 \cdot \hat{\Sigma}^{(i)} = \left(\frac{\tilde{\sigma}^{(i)}}{\gamma^{(i)}} \right)^2 \cdot \left(\hat{L}^{(i)}(\hat{L}^{(i)})^\top + \hat{D}^{(i)} \right) \tag{4.15}$$

4.3. Single- and Multi-World Settings

In this section, we motivate, introduce and define the *single-* and *multi-world* scenarios. In many real-world forecasting tasks, such as electricity demand (Hyndman and Fan 2010; Vaghefi et al. 2014), trajectory paths (Yuan and Kitani 2019; Mangalam et al. 2021) or GPU temperature forecasting (Desai et al. 2024; Wang et al. 2024), identical historical contexts can lead to multiple plausible futures. Consider the increasingly important task of forecasting GPU temperatures in data centers, where thermal management is critical to avoid throttling, performance degradation, or hardware failure (Cao and Wang 2017; Zhao et al. 2023). For instance, Figure 4.5 shows how a GPU might report a stable 50°C in the early morning, indicating it is idle. But what happens next depends on whether or

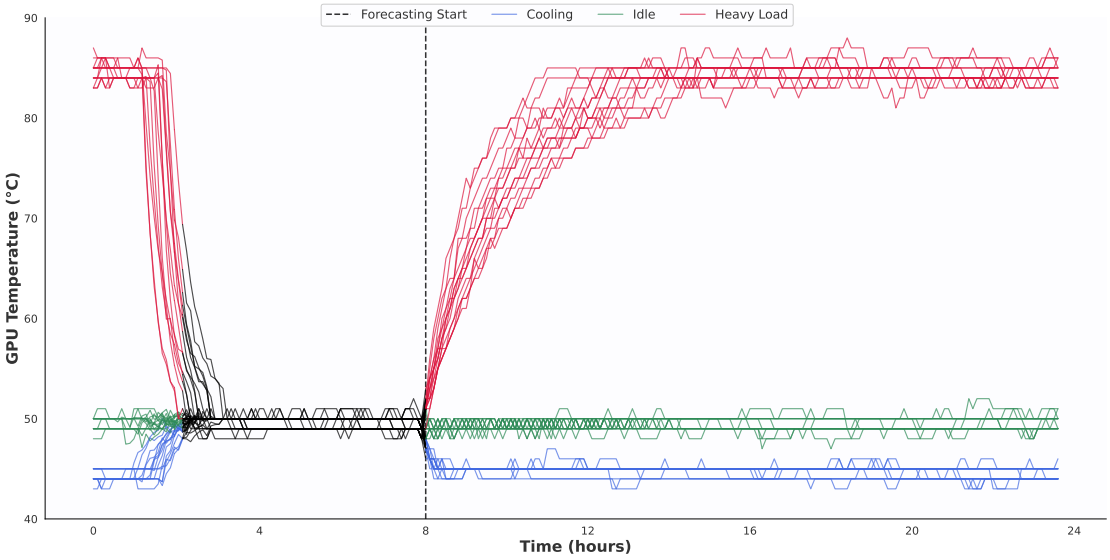


Figure 4.5.: Illustration of a synthetic *multi-world* scenario in GPU temperature forecasting. Given an identical prefix (e.g., a stable 50°C temperature before the forecasting start), several plausible future trajectories emerge depending on external, unobserved factors such as task scheduling. The system may remain idle (green), become active (red), or cool down (blue), despite indistinguishable historical context.

not it gets assigned a task: the temperature could suddenly rise with heavy computation, stay constant if unused, or even drop slightly as cooling kicks in. Importantly, the prefix alone, the historical temperature readings, does not uniquely determine which trajectory will follow and additional context (e.g., scheduling metadata) may not be available. Moreover, in this setting we define *multi-world* scenarios as situations in which a prefix can lead to multiple plausible futures, indistinguishable by historical context alone.

Definition 1 (*Multi-World* Scenario)**.** *Let $\mathcal{X}$ be the observation space and $\mathbf{x}_{1:L} \in \mathcal{X}^L$ a fixed prefix. A multi-world* scenario *arises if the conditional distribution $p(\mathbf{x}_{L+1:L+H}|\mathbf{x}_{1:L})$ is* multi-modal*, meaning there exist subsets $\mathcal{T}_1, \ldots, \mathcal{T}_K$ of the trajectory space $\mathcal{X}^H$ —the set of all sequences of length H over $\mathcal{X}$— such that:*

1. *Each mode has nonzero probability: $p(\mathbf{x}_{L+1:L+H} \in \mathcal{T}_k \mid \mathbf{x}_{1:L}) > 0$,*
2. *The modes are mutually exclusive: $p(\mathcal{T}_i \cap \mathcal{T}_j \mid \mathbf{x}_{1:L}) = 0$ for all $i \neq j$,*
3. *The prefix $\mathbf{x}_{1:L}$ does not determine a unique outcome; formally, there exist $i \neq j$ such that*

$$p(\mathbf{x}_{L+1:L+H} \in \mathcal{T}_i \mid \mathbf{x}_{1:L}) > 0 \quad \textit{and} \quad p(\mathbf{x}_{L+1:L+H} \in \mathcal{T}_j \mid \mathbf{x}_{1:L}) > 0.$$

Revisiting our example in Figure 4.5, $\mathcal{T}_1$ could represent futures where the system becomes active (e.g., GPU heats up), $\mathcal{T}_2$ where it remains idle and $\mathcal{T}_3$ where it cools

down. These outcomes are coherent but mutually exclusive, and any given sample trajectory $\hat{\mathbf{x}}_{L+1:L+H} \in \mathcal{X}^H$ should clearly belong to one of them. In contrast, *single-world* scenarios refer to settings where a given prefix, i.e., the observed historical context, deterministically leads to a narrow and (typically) unimodal conditional distribution $p(\mathbf{x}_{L+1:L+H}|\mathbf{x}_{1:L})$ over future trajectories. For instance, consider restricting the synthetic GPU example to only the heavy load behavior (red): here, the future would be consistently predictable given the prefix, forming a single, dominant mode. Similarly, if we modify the forecasting task by shifting the forecasting point (black dashed line) from hour 8 to 12 in Figure 4.5, the divergence between different futures (e.g., heating, cooling, idling) occurs several steps before the prefix ends. Hence, it is possible to uniquely determine the correct world. Altogether, the key distinction is that while *single-world* settings exhibit predictability conditioned on the observed history, *multi-world* scenarios involve inherent uncertainty, in which multiple plausible outcomes exist, each corresponding to a different external cause that is not encoded in the prefix. Therefore, modeling *multi-world* scenarios requires probabilistic methods which can represent and sample from diverse, coherent trajectories that quantify all possible scenarios reasonably. Accurately quantifying these *multi-world* futures becomes especially important when rare but high-impact scenarios are possible, e.g. thermal spikes leading to GPU burnout or business analysts that want to consider extreme scenarios, as these outcomes are often visually and statistically distinct from the majority and may carry high operational cost or risk (Kan et al. 2022; Wen et al. 2018).

In forecasting tasks, DMS models, which have shown particularly dominant performances in point LTSF (see Table 3.1), generate all future steps in a single forward pass conditioned only on the input context. Hence, probabilistic DMS models that forecast univariate predictions at every time step are unable to model temporal dependencies between predictions, as they assume conditional independence between future steps (Taieb et al. 2012), e.g. $\{p(x_{L+1}|x_{1:L};\, \theta), p(x_{L+2}|x_{1:L};\, \theta), \ldots, p(x_{L+H}|x_{1:L};\, \theta)\}$. In *multi-world* scenarios, this limitation becomes critical: even if a DMS model accurately captures the multimodal distribution at each individual time step (for instance via flexible density estimation (Drouin et al. 2022; Bergsma et al. 2023; Ashok et al. 2023; Bergsma et al. 2022)), its predictions across time can become incoherent, yielding inconsistent sample trajectories that fail to reflect any realistic evolution of the true underlying process. For example, consider Figure 4.6, which visualizes the predictive distribution over time using Gaussian Kernel Density Estimation (KDE) (Parzen 1962; Scott 2015; Wang et al. 2013) applied to the synthetic data from Figure 4.5. The plot focuses on the time window between hours 7 to 9 and depicts the ground truth trajectories in gray, a single sample forecast as a blue dashed line, and the start of the forecast horizon as a vertical black dashed line. The blue right-facing KDE curves represent estimated densities at each time step. These per-timestep densities are estimated conditionally independently, like in the DMS model. Furthermore, the sampled forecast trajectory passes through regions of high marginal density, suggesting plausibility at each individual time step. However, the overall trajectory is inconsistent with the ground truth, revealing a lack of temporal coherence. This example illustrates how models that fail to capture joint

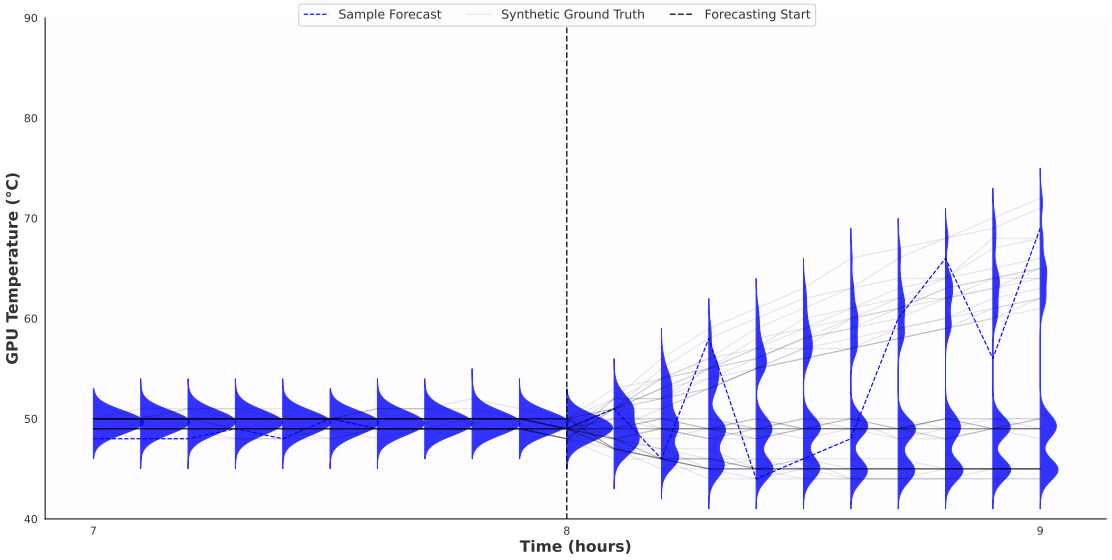


Figure 4.6.: KDE-based density visualization of synthetic GPU temperature forecasts. The plot shows ground truth trajectories (gray), a sample forecast trajectory (blue dashed), and per-timestep KDEs (blue vertical curves) between hour 7 and 9. The forecasting start is marked by a vertical black dashed line.

dependencies over time may produce forecasts that are statistically valid in isolation but unrealistic when viewed as a complete sequence, highlighting the need for coherence across time in *multi-world* forecasting. While there are multiple approaches to model the multivariate temporal dependencies explicitly (see Section 3.2), sophisticated copula or generative methods either require significant modeling complexity, high computational resources or both (Salinas et al. 2019; Drouin et al. 2022; Ashok et al. 2023; Li et al. 2023; Yan et al. 2024). Therefore, we will rely on the low-rank plus diagonal decomposition, which ensures positive-definiteness and reduces the number of parameters (Wu et al. 2020; Horn and Johnson 2012). In contrast, probabilistic IMS methods factor the joint distribution over future time steps into a sequence of one-step-ahead conditional distributions (see Equation 2.5). This yields terms such as $\{p(x_{L+1}|x_{1:L};\ \theta)$, $p(x_{L+2}|x_{1:L}, \hat{x}_{L+1};\ \theta), \ldots, p(x_{L+H}|x_{1:L}, \hat{x}_{L+1:L+H-1};\ \theta)\}$, where $\hat{\mathbf{x}}$ denotes a realization drawn from previous predictive distributions. Typically, $\hat{\mathbf{x}}$ is a sample, as in DeepAR (Salinas et al. 2020) (see Section 4.1). Therefore, they are able to capture temporal dependencies and multimodality via sequential step-by-step predictions. In the next section, we will first discuss the experimental setup. After that, we present the results of our synthetic *multi-world* experiment among other experimental results.

5. Experimental Evaluation

In this chapter, we present a comprehensive evaluation of the proposed models. We begin by outlining the experimental setup, including the datasets used, implementation details, evaluation metrics, and the procedures for hyperparameter tuning. To gain deeper insight into the behaviors and trade-offs between DMS and IMS models, we analyze a controlled synthetic *multi-world* scenario. In the last section of this chapter, we evaluate performance on real-world LTSF benchmarks, providing a thorough probabilistic comparison across multiple criteria.

5.1. Experimental Setup

In this section, we detail the datasets, implementation, evaluation protocol and hyperparameter tuning setup used in our proposed modeling framework.

Selection of Datasets. Before characterizing our chosen experimental datasets, we first motivate their selection. The BasicTS+ benchmark offers a diverse range of LTSF datasets, with their descriptive statistics summarized in Table 5.1. Moreover, BasicTS+ encompasses the Electricity Transformer Temperature (ETT)[1], Weather[2], Electricity[3] and Traffic[4] data sets. In addition to the basic data statistics, Table 5.1 lists forecastability values sourced from Shang et al. (2024), who derived them by subtracting the entropy of the Fourier decomposition of a time series from 1 (Wang et al. 2023; Goerg 2013). Furthermore, higher values represent a series that has greater forecastability, i.e. is regarded as easier to predict. Out of these seven datasets, our aim is to identify those that exhibit the clearest signs of *multi-world* scenarios, where distinct future trajectories may emerge from indistinguishable historical contexts. To explore this, we segment each Z-score standardized time series into overlapping windows of total length $L + H = 96 + 720$, ensuring that the prefix of length $L = 96$ is non-overlapping. This guarantees that each window begins with an entirely unique prefix. We then apply K-means clustering (MacQueen 1967) to the prefix portion of each segment, under the assumption that similar prefixes may be a sign of indistinguishability with respect to the forecasting horizon. After clustering, we evaluate diversity and multi-modality in the forecast horizon by employing both metric-based analysis and visual inspection. To begin with, we first cluster the prefix of each series, however classical clustering based on

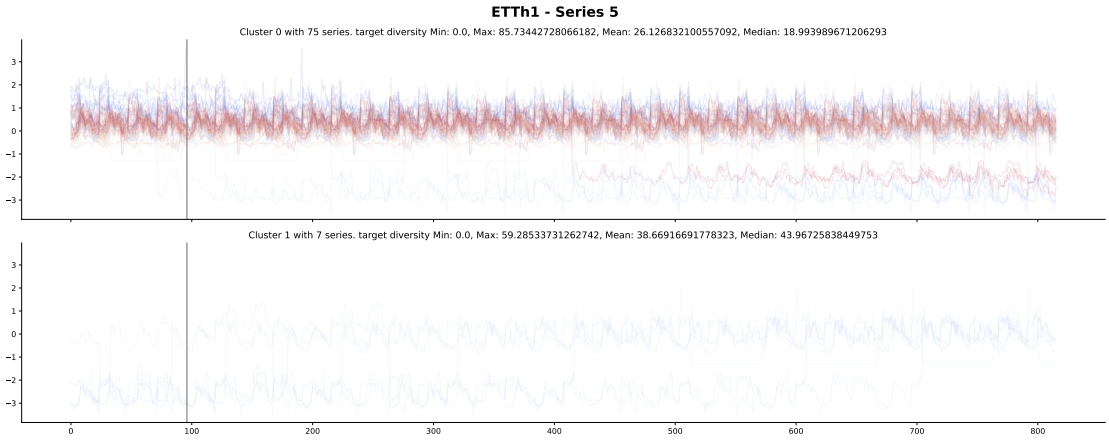
[1] `https://github.com/zhouhaoyi/ETDataset`
[2] `https://www.bgc-jena.mpg.de/wetter/`
[3] `https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014`
[4] `http://pems.dot.ca.gov/`

Table 5.1.: Descriptive statistics of LTSF datasets as provided in the BasicTS+ benchmark. Forecastability values are taken from Shang et al. (2024), where higher values indicate greater predictability. The maximum diversity is computed based on our identification of *multi-world* scenarios, where higher values indicate a stronger alignment with such scenarios.

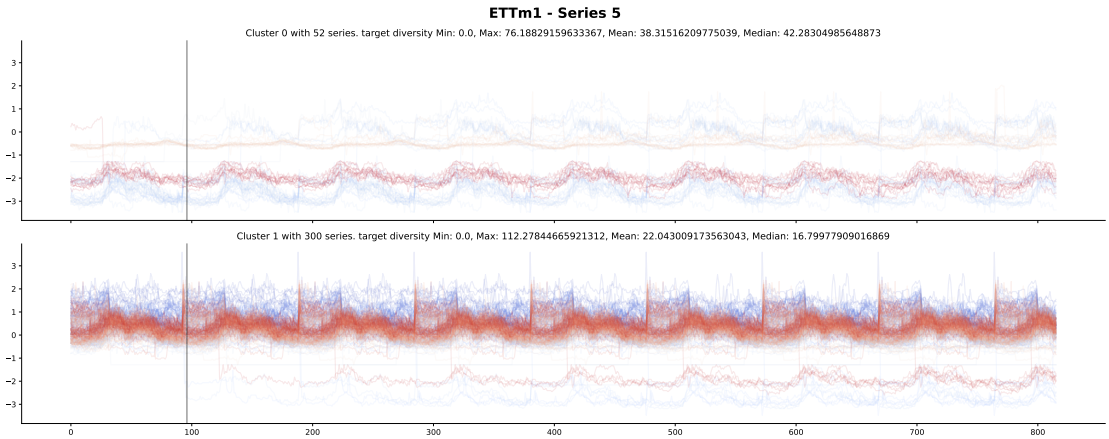
Name	**#Time Series**	**Timesteps per series**	**Frequency (m)**	**Forecastability**	**Max. Diversity**
ETTh1	7	14400	60	0.38	85.7344
ETTh2	7	14400	60	0.45	124.6082
ETTm1	7	57600	15	0.46	112.2784
ETTm2	7	57600	15	0.55	137.2579
Weather	21	52696	10	0.75	324.6322
Traffic	862	17544	60	0.68	121.5532
ExchangeRate	8	7588	1440	-	79.1688

Euclidean distance is often inadequate for time series due to its sensitivity to distortions such as time shifting, scaling, and occlusions (Paparrizos and Gravano 2015; Paparrizos et al. 2024). As a more robust alternative, Dynamic Time Warping (DTW) (Chu et al. 2002; Sakoe and Chiba 1978) was introduced, offering an elastic distance measure that aligns sequences by minimizing the total warping cost (Paparrizos et al. 2024). Following prior work on time series clustering (Izakian et al. 2015; Anh and Thanh 2015; Bothwell et al. 2022; Li et al. 2020), we adopt DTW as the distance metric and integrate it with K-means clustering via the tslearn[5] (Tavenard et al. 2020) and scikit-learn[6] (Pedregosa et al. 2011) libraries. To determine the optimal number of clusters, we vary K from 2 to 5 and evaluate the clustering quality using the Davies-Bouldin score (Davies and Bouldin 1979), which favors well-separated and compact clusters. After determining clusters, we compute pairwise DTW distances within the forecasting horizon of each cluster. This aids as a metric to assess multi-modality and diversity among trajectories that originate from similar pasts, in which we report the mean, median, minimum and maximum distances. Moreover, Table 5.1 reports the maximum diversity values within the forecasting horizon of individual clusters, suggesting that the Weather dataset may be a promising candidate. However, to make a more grounded and informed selection, we further investigate the cluster visualization of those series that exhibit maximum diversity in the forecasting horizon. The resulting illustrations are shown in Figure 5.1 and in Appendix A (Figures A.1, A.2, A.3, A.4 and A.5). To highlight temporal progression, we use a colormap that transitions from blue (older segments) to red (more recent ones), following Liu et al. (2024). While datasets such as Traffic, Weather, and ExchangeRate exhibit high maximum diversity values in Table 5.1, closer inspection reveals that these arise primarily from rare, extreme outliers occurring in only a few time points, see Figures A.3, A.4 and A.5. Moreover, for these datasets, cluster distributions are often highly imbalanced. For instance, in the Weather dataset, 320 series are assigned to the first cluster while only one is assigned to the second. This questions the claim of prefix

[5] `https://tslearn.readthedocs.io/`
[6] `https://scikit-learn.org/`



(a) LULL series of the ETTh1 dataset.



(b) LULL series of the ETTm1 dataset.

Figure 5.1.: Visualization of clustering results for ETTh1 (a) and ETTm1 (b) datasets, highlighting diversity in the forecast horizon. Each subplot shows segments grouped by prefix similarity (using DTW-based K-means clustering), where colors transition from blue (earlier segments) to red (later ones) to indicate temporal progression.

similarity for those clusters, since nearly all series are grouped together. In contrast, the clusters formed within the ETT datasets display more coherent multi-modal structures sustained over longer periods. Among these, we select ETTh1 and ETTm1, not for their highest diversity scores, but for showing more interpretable and balanced multi-modal behavior. For instance, while ETTm2 (Figure A.2, top panel) shows only a single divergent trajectory within cluster 0, ETTm1 (Figure 5.1b, top panel) displays two clearly distinguishable modes within cluster 0. Likewise, similar patterns are observed in the comparison of ETTh1 and ETTh2. Additionally, according to the forecastability scores reported by Shang et al. (2024) in Table 5.1, ETTh1 and ETTm1 rank among the most challenging datasets. Furthermore, this strengthens our selection motivation, since more challenging datasets enable us to assess model performance in scenarios characterized by high uncertainty and complex temporal dynamics, reflecting more realistic and difficult forecasting conditions. Lastly, we provide box plots describing the broader distribution of maximum and mean diversity values across individual series of each dataset in Figures A.6 and A.7 in Appendix A.

Data. In the following, we introduce the experimental LTSF datasets used in our study. Concretely, the ETTh1 and ETTm1 datasets are subsets of the Electricity Transformer Temperature (ETT) dataset (Zhou et al. 2021), developed to support LTSF tasks. These datasets are derived from real-world electricity transformer measurements collected over a two-year period (2016/07–2018/07) in two seperate counties of China and are provided by the Beijing Guowang Fuda Science and Technology Development Company. Consistent with prior studies (Zhou et al. 2021; Nie et al. 2022), we model each county and frequency combination separately. Furthermore, {ETTh1, ETTm1} and {ETTh2, ETTm2} correspond to data from the first and second county respectively. Moreover, in these datasets, each data point includes seven variables, which we model independently to isolate the probabilistic temporal modeling capabilities. The oil temperature (OT) of electricity transformers alongside six power load-related features: High UseFul Load (HUFL), High UseLess Load (HULL), Middle UseFul Load (MUFL), Middle UseLess Load (MULL), Low UseFul Load (LUFL), Low UseLess Load (LULL). In accordance with LTSF practices (Zhou et al. 2021), we set the look-back window to $L = 96$ and the prediction horizon to $H = 720$. In line with the default configuration of the BasicTS+ benchmark and with standard practices in LTSF (Zhou et al. 2021; Nie et al. 2022; Liu et al. 2023; Shang et al. 2024), we use a data split ratio of 60% for training, 20% for validation, and 20% for testing. Moreover, each training, validation, and test sample is generated using a standard TSF sliding window approach, where the window shifts forward by one time step for each sample (Shao et al. 2025). The top panels of Figure 5.2 display the Oil Temperature time series for the ETTh1 and ETTm1 datasets, showing the full span of the training, validation, and test sets concatenated. Each series is visualized using non-overlapping windows of length $96 + 720 = 816$, corresponding to the input context and forecasting horizon in our forecasting setup. These upper two plots highlight the presence of distribution shifts: the newer segments (in red) exhibit weaker variations and have lower values compared to the older ones (in blue), which ap-

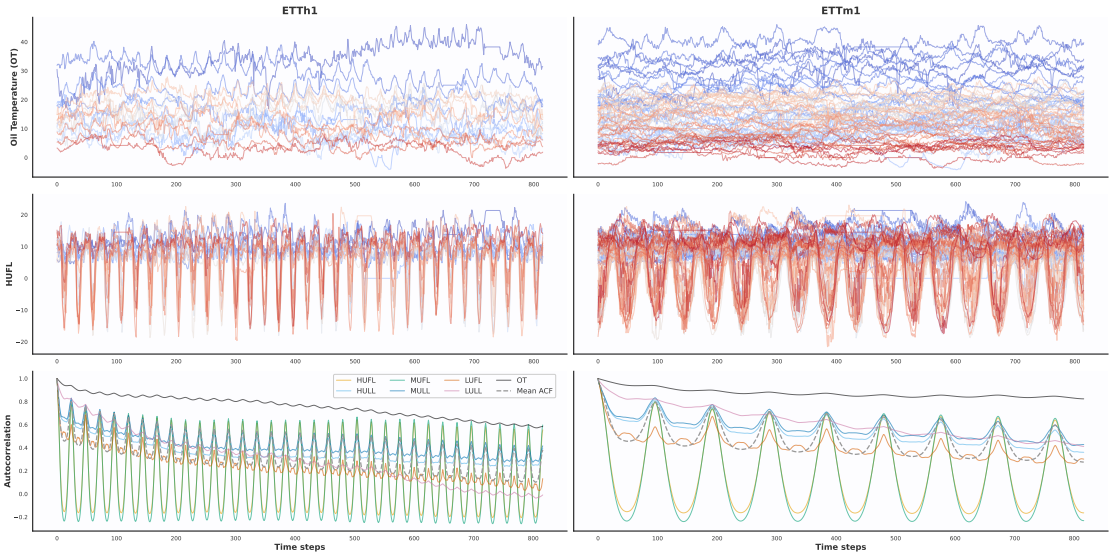


Figure 5.2.: Visualization of the ETTh1 and ETTm1 datasets. **Top:** Oil Temperature (OT) series shown in non-overlapping windows of length $816 = 96 + 720$, with a color gradient from blue (earlier segments) to red (later segments) indicating temporal progression. **Middle:** High UseFul Load (HUFL) series, selected due to its clear cyclical patterns. **Bottom:** Autocorrelation function (ACF) values for all seven variables, with the mean ACF shown as a dashed line. Clear periodic spikes appear at lag 24 (ETTh1) and lag 96 (ETTm1), consistent with daily seasonality.

pear more volatile. In addition, the two plots also highlight the lower forecastability of ETTh1 compared to ETTm1: the hourly aggregation reduces the amount of information available during training while simultaneously requiring the model to make predictions that are spaced further apart in real time. The middle plot shows the High Useful Load (HUFL) series, selected for its clearly observable cyclical patterns. The visualizations highlight this structure: in ETTh1, cycles occur every 24 steps, consistent with daily rhythms given the hourly resolution. In contrast, ETTm1 exhibits cycles every 96 steps, reflecting its finer 15-minute resolution. However, since the window length of 816 is not a perfect multiple of 96, the cycles appear misaligned in the plot. The remaining variables of both data sets are shown in Figure A.8 of Appendix A. Lastly, the bottom two plots in Figure 5.2 display the autocorrelation function (ACF) (Madsen 2007) for each of the seven variables in the datasets, with the mean ACF across variables shown as a dashed line. The ACF quantifies the correlation between a time series and its lagged values and is computed as follows:

$$\text{ACF} = \frac{\sum_{t=1}^{H+L-k}(x_t - \hat{\mu}_x)(x_{t+k} - \hat{\mu}_x)}{\sum_{t=1}^{H+L}(x_t - \hat{\mu}_x)^2} \tag{5.1}$$

where $H+L$ are the number of observations, x_t is the value at time t, k is the lag, and $\hat{\mu}_x$ is the sample mean of the series (Lin et al. 2024). These plots highlight the periodicity present in the data: in line with the findings of Lin et al. (2024), strong spikes appear at every 24th lag in ETTh1 and every 96th lag in ETTm1, corresponding to daily cycles at their respective temporal resolutions. Among the variables, Oil Temperature displays a high positive level of autocorrelation throughout, suggesting strong temporal continuity and a smooth, stable progression over time. This indicates that current values are highly predictive of future values, with minimal short-term fluctuations or noise. In contrast, MUFL and HUFL show greater fluctuations and stronger cyclical trends in their autocorrelation.

Implementation details. As we have already mentioned, we use the BasicTS+ benchmark provided by Shao et al. (2025) to run our experiments in a unified and standardized pipeline. All experiments are conducted using the open-source software Python 3.11.11 (Van Rossum and Drake Jr 1995) with PyTorch 2.8.0 (Paszke et al. 2019) and Easytorch 1.3.2 (Wang 2020), executed on the Linux server provided by the Data and Web Science Group (DWS)[7] of University Mannheim. Furthermore, the selected partition of the DWS-server is equipped with an AMD EPYC 7413 24-Core processor and a single NVIDIA RTX A6000 graphics card, with each experiment repeated five times using different random seeds. In addition to the experiments shown, the hyperparameter search was extended using the bwUniCluster 3.0, supported by the state of Baden-Württemberg through bwHPC[8]. While the software environment on bwUniCluster 3.0 remained identical, hardware configurations varied due to the use of multiple partitions to enable efficient scaling.

Evaluation. We train and tune our models according to their respective loss functions. For models with distributional heads, we use the negative log-likelihood (see Equation 3.3), which is a reliable and standard training objective for univariate and multivariate distributional probabilistic forecasting (Marcotte et al. 2023). For models with quantile heads, we adopt the quantile loss (see Equation 3.15), the default choice for quantile regression (Koenker and Bassett 1978; Koenker 2005). To evaluate our models, we use a comprehensive set of probabilistic scoring rules as outlined in Section 3.3. These include the NLL, CRPS, CRPS_Sum, Energy Score (ES), Variogram Score (VS), Kullback-Leibler Divergence, Coverage, Sharpness, the Weighted Quantile Score (WQS) and the Weighted Interval Score (WIS). With the exception of the NLL, which is directly computed from the predictive distribution, all metrics are evaluated either from forecast samples or from quantile predictions. Furthermore, for distributional methods, we compute each metric (when applicable) in two ways: (1) using 100 samples drawn from the forecast distribution and (2) using quantiles derived from those same samples. This allows for direct and fair comparison with quantile-based models, which predict quantiles explicitly. In line with the M5 competition setup (Makridakis et al.

[7] `https://www.uni-mannheim.de/dws/`
[8] `https://www.bwhpc.de/`

2022; Chen et al. 2022), we use the following quantile levels for training and evaluation: Q = `[0.005, 0.025, 0.165, 0.25, 0.5, 0.75, 0.835, 0.975, 0.995]`. In addition to probabilistic metrics, we report common point forecasting metrics used in LTSF benchmarks (Nie et al. 2022; Jia et al. 2023; Liu et al. 2023). Specifically, the Mean Squared Error (MSE) and Mean Absolute Error (MAE), which are computed using the median forecast: the median of the predictive samples and the median quantile prediction for the quantile method since it cannot produce reasonable samples. The metrics are defined as:

$$\text{MSE} = \frac{1}{H}\sum_{t=1}^{H}(x_t - \hat{x}_t^{\alpha=0.5})^2, \quad \text{MAE} = \frac{1}{H}\sum_{t=1}^{H}|x_t - \hat{x}_t^{\alpha=0.5}|, \tag{5.2}$$

where H is the prediction horizon, and x_t and $\hat{x}_t^{\alpha=0.5}$ are the observed and predicted median values, respectively. Unless stated otherwise, all reported results are averaged over 5 independent runs with different random seeds to ensure robustness.

Hyperparameter tuning. We perform hyperparameter optimization via the wandb[9] python library (Biewald 2020). For this, we employ the Bayesian hyperband approach (Li et al. 2018; Wang et al. 2018), which enables efficient exploration of the hyperparameter space by allocating more resources to promising configurations while eliminating underperforming ones. Moreover, the optimization in wandb begins with a few random trials to initialize the model. Then, informed search proceeds via Bayesian optimization (Dewancker et al. 2016), which uses a Gaussian Process surrogate model, sourced from scikit-learn[10] (Pedregosa et al. 2011). Simultaneously, the pruning process of the hyperband strategy is defined by the `eta` and `min_iter` parameters (set to 3 and 2, respectively), resulting in evaluation checkpoints at epochs $[2, 6, 12, 24, 48, 96]$. At each checkpoint, only the best-performing $r = \frac{1}{\texttt{eta}}$ fraction of configurations with respect to the validation loss are allowed to continue. Each configuration is trained for up to 100 epochs, with additional early stopping applied if the model shows no improvement over the last 5 epochs, as in Shang et al. (2024). Overall, we tune 22 unique configurations, resulting from the combination of 2 datasets, 3 models, and 4 probabilistic heads per model (except DeepAR, which has 3 probabilistic heads due to its inability to model temporal multivariate parametric distributions directly). For each configuration, we explore 100 different hyperparameter combinations. The hyperparameters along with their ranges and values are summarized in Tables 5.2, B.1, and B.2. To reduce the search space dimensionality, we fix the optimizer to ADAM (Kingma and Ba 2014) and use a multi-step learning rate scheduler that decreases the learning rate by multiplying it with a hyperparameter gamma after 5 and 25 epochs. General parameters, such as the learning rate of ADAM, scalers or batch size, are detailed in Table 5.2. Parameters specific to each model and distributional head are listed in Tables B.1 and B.2 of Appendix B, respectively.

[9] `https://www.wandb.com/`

[10] `https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.GaussianProcessRegressor.html`

Table 5.2.: Overview of hyperparameters and their respective value ranges or distributions that are consistently applied across all model architectures during hyperparameter optimization.

Parameter	Range / Values
Norm_Each_Channel	[True, False]
Scaler	[ZScoreScaler, MinMaxScaler, None]
ADAM.lr	Uniform(2.5e-4, 2.5e-2)
ADAM.weight_decay	Uniform(1.0e-5, 1.0e-3)
MultiStepLR_Scheduler.gamma	Uniform(0.01, 0.7)
Batch_Size	[16, 32, 64, 128]

5.2. Simple Multi-World Example

In this section, we analyze the behavior of DMS and IMS models in a controlled, synthetic *multi-world* setting. As illustrated in Figure 5.3, all trajectories begin with an identical prefix. At a fixed branching point, the future diverges along one of two possible sine-paths (orange curves), each subject to additive Gaussian noise. Since the prefix is invariant and insufficient to distinguish which future will unfold, this scenario represents the *multi-world* scenario, where multiple futures are indistinguishable from the past. We evaluate four simple models in this environment. Three of them are DMS models: a univariate model that forecasts conditionally independent univariate Gaussian distributions at each time step, a multivariate model that predicts a full-rank multivariate Gaussian over the full forecast horizon and a low-rank multivariate model with rank $H/2$ over the forecasting horizon. All DMS models share a common backbone architecture: a two-layer MLP with hidden size 64 and ReLU activations. In addition, we implement an IMS model that forecasts univariate Gaussians at each step, trained with teacher forcing (Williams and Zipser 1989). It uses a single LSTM cell (hidden size 64), followed by a ReLU and a small MLP to output the mean and standard deviation. During inference, it employs ancestral sampling (Salinas et al. 2020), where each prediction is conditioned on previously sampled values. All models are trained for 400 epochs using the NLL loss, with the ADAM optimizer and a learning rate of 0.003. The task is to predict the next 50 time steps based on the fixed 50-step prefix. Figure 5.4 shows 20 generated samples from each trained model, alongside ground truth trajectories (truncated to the forecasting range, since the prefix is fixed and uninformative). The behavior of the univariate DMS model stands out: its assumption of conditional independence leads to unrealistic zigzagging trajectories, not present in the actual data. By contrast, the other three models (IMS and both multivariate DMS variants) produce samples that more closely resemble the ground truth. However, the multivariate DMS models exhibit noticeably higher variance in their trajectories, likely due to inaccuracies in estimating the covariance structure, leading to compounding variances across time steps. Hence, these samples often lie far outside the true data distribution. The IMS model shows

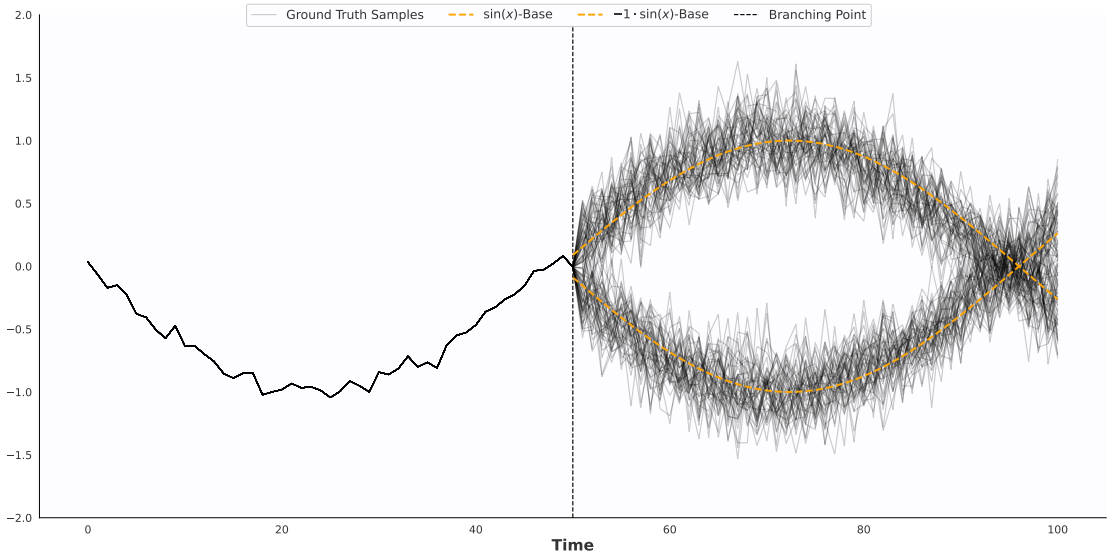


Figure 5.3.: Illustration of the synthetic multi-world setting. All trajectories share a common prefix (first 50 steps), after which the system branches into two possible futures (orange), each perturbed by Gaussian noise. This setup represents a multi-modal prediction task where the future is not identifiable from the past.

the highest alignment with the underlying structure, the two orange paths. However, its forecasts are smoother and less noisy than the true data, suggesting that it captures the base paths well but underestimates variance. In Figure C.1 (Appendix C), we visualize central predictive intervals based on 100 generated samples per model. Here, the univariate DMS model produces intervals that are visually closest to the ground truth distribution, whereas both multivariate DMS models again overestimate the variance and produce overly broad intervals. Since the forecasts of the IMS model are smooth, the 50 and 90 % interval lie close to each other. Despite these insights, Figure C.1 shows that constructing meaningful intervals for multi-modal distributions is inherently challenging, since standard quantile-based approaches often assume unimodality and contiguous support, which can obscure or entirely miss the presence of distinct modes (Deliu and Liseo 2024). Several strategies exist to address this, though each comes with trade-offs. Kernel Density Estimation (KDE) (Parzen 1962) is commonly used to estimate the density (Rosenblatt 1956; Chen et al. 2024; Mészáros et al. 2024; Olsen et al. 2024), but its accuracy depends heavily on the kernel function and the kernel bandwidth selection (Chen et al. 2024). Following the common setup of Gaussian KDE (Scott 2015; Olsen et al. 2024), we visualize the intervals of the KDE with a bandwidth of 0.1 in Figure C.2. Nonetheless, the accuracy of KDE is highly sensitive to the choice of bandwidth: over-smoothing can blur separate modes into a single peak, while under-smoothing may yield noisy, unreliable estimates—especially in higher-dimensional settings (Chen et al. 2024). Contrary to KDE, k-nearest neighbor (kNN) density estimation (Loftsgaarden

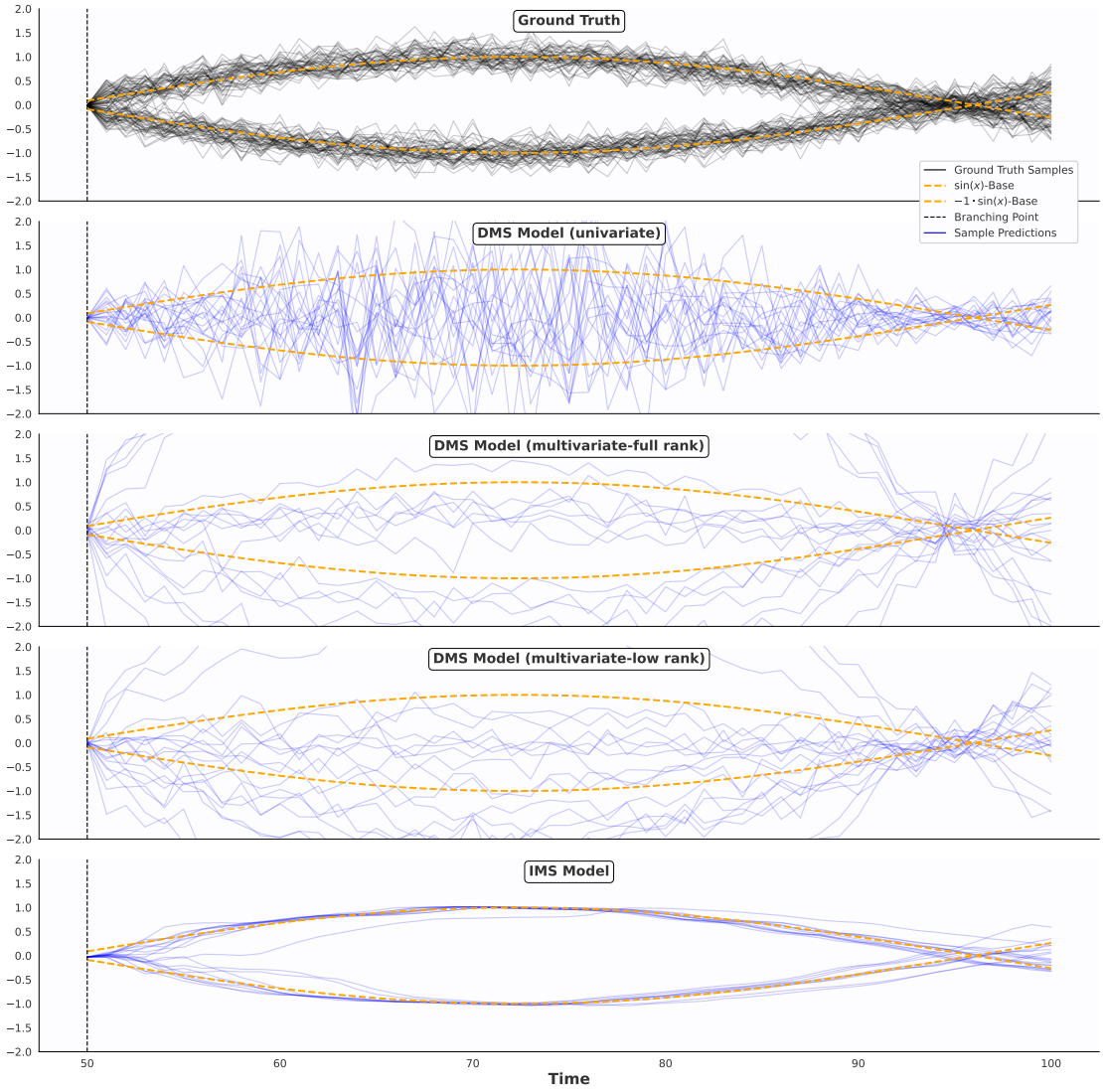


Figure 5.4.: Ground truth and sample trajectories generated by each model truncated to the prediction range. The first panel shows the ground truth distribution, whereas each following panel shows 20 generated sample forecasts (blue) and ground truth base trajectories (orange). The univariate DMS model produces unrealistic zigzagging paths due to conditional independence assumptions, while multivariate DMS and IMS models better reflect the true modes. IMS forecasts are closest to the ground truth samples, but too smooth.

and Quesenberry 1965) estimates the density at a point based on the distance to its k-th nearest neighbor (Loftsgaarden and Quesenberry 1965; Zhao and Lai 2022). This nonparametric method adapts to the shape of any continuous underlying distribution, however, kNN-based estimates are only asymptotically accurate and can be sensitive in finite-sample regimes (Dasgupta and Kpotufe 2014; Zhao and Lai 2021; Olsen et al. 2024). In our approach, we use $k = 10$, where the intervals are shown in Figure C.3. Overall, Figures C.2 and C.3 both illustrate that the IMS model aligns closely with the ground truth distribution, while the other models tend to assign high density to regions that appear implausible. Table 5.3 presents quantitative metrics for all models. How-

Table 5.3.: Forecasting performance metrics across models on our synthetic *multi-world* example. Comparison of univariate, multivariate evaluation metrics and estimated KL divergences. We downscaled the VS values, for better interpretation. The best performances for each metric are highlighted in bold, while the second-best results are underlined.

Metric	**Univariate DMS**	**Multivariate DMS (full rank)**	**Multivariate DMS (low rank)**	**IMS**
KL divergence (KDE)	1275.6065	210.6685	<u>187.7011</u>	**101.4629**
KL divergence (kNN)	60.1763	21.3201	<u>11.7709</u>	**-8.2508**
CRPS	**0.3933**	0.6129	0.4756	<u>0.3943</u>
VS_0.5 ($\times 10^2$)	3.0365	4.7093	<u>2.3564</u>	**1.411**
VS_1 ($\times 10^2$)	<u>7.0590</u>	30.0169	10.76	**2.2303**
VS_2 ($\times 10^3$)	<u>2.473</u>	91.9942	20.9243	**0.4558**
ES	<u>3.6286</u>	5.0963	3.9637	**3.2639**
wQS	**0.4054**	0.7712	0.5546	<u>0.4198</u>
WIS	**0.252**	0.4786	0.3441	<u>0.2615</u>

ever, the interval and quantile metrics employed in our study primarily rely on central predictive intervals. Moreover, rewriting these scoring metrics to properly account for the quantiles and intervals of multi-modal distributions lies beyond the scope of this work, but represents a promising direction for future research. In Table, 5.3, except for KL divergence, each score is computed by iteratively comparing all 100 generated samples against one ground truth trajectory, with results averaged across all ground truth samples. To approximate KL divergence in the absence of a parametric ground truth distribution, we compare the KL divergence of KDE and kNN-based density estimators. Both KL divergence approximations underscore the superior performance of the IMS model, while highlighting significantly poorer results of the univariate DMS model. Notably, the univariate DMS model performs best on several univariate metrics (e.g., CRPS, wQS, WIS), all of which ignore multivariate correlations. In contrast, multivariate metrics consistently rank the IMS model highest, confirming that it best captures the temporal distribution structure. The low-rank DMS model outperforms the full-rank variant, suggesting overparameterization in the latter. To further assess model performance, we independently analyze calibration and sharpness in Figure 5.5.

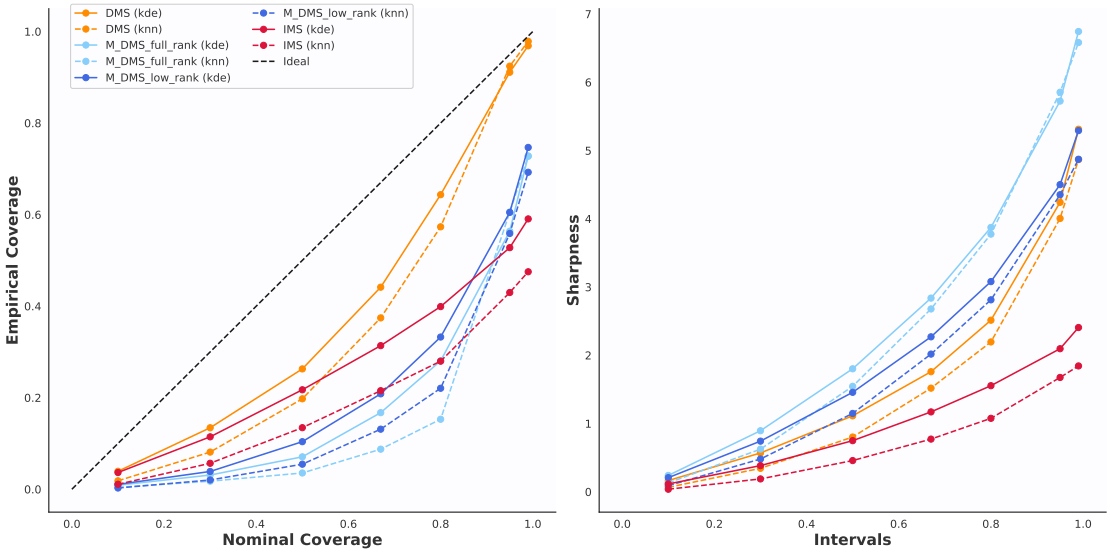


Figure 5.5.: Calibration (left) and sharpness (right) curves for our four models. The coverage is determined on the basis of intervals and not quantiles due to the multi-modal data structure. The intervals are approximated via KDE (solid lines) and kNN (dashed lines).

Both metrics are computed using KDE and kNN-based intervals, which better capture multi-modal predictive distributions than standard central intervals. For reference, coverage and sharpness results using central intervals are included in Figure C.4 (Appendix C). In the left panel of Figure 5.5, we first want to emphasize that coverage is evaluated based on intervals and not quantiles, as quantiles are difficult to define in multi-modal settings (Deliu and Liseo 2024). Among all models, the univariate DMS (orange) model is best calibrated, following the diagonal the closest. The IMS model shows good calibration at lower nominal levels but diverges at higher levels, likely due to its smooth trajectory predictions and resulting narrow intervals, leading to under-coverage. Among the multivariate DMS variants, the low-rank model outperforms the full-rank version. The kNN-based intervals are generally less calibrated, as they tend to be narrower and more stratified (see Figures C.2 and C.3). On the right side of Figure 5.5, we display sharpness, where lower values indicate more precise forecasts. The IMS model achieves the sharpest intervals, again due to its smooth predictions. The univariate DMS model follows, with the multivariate DMS models producing the widest intervals due to higher variance. Altogether, both the univariate DMS and IMS models perform best overall: the former excels in calibration, while the latter produces sharper predictions. However, these metrics do not capture temporal coherence, where the IMS model holds a clear advantage, as shown in Figure 5.4.

In summary, despite making the evaluation via standard central interval metrics difficult, the multi-modal nature of *multi-world* scenarios revealed important aspects about IMS

and DMS models. First, the conditional independence assumption of the univariate DMS model hinders coherent and realistic sample forecasts. Second, in contrast, the IMS model emerges as the most reliable across qualitative sample inspection and quantitative multivariate metrics. Third, univariate metrics fail to reveal these distinctions, whereas multivariate evaluations correctly capture the advantages of the IMS model. These results underscore the limitations of DMS methods compared to IMS models in *multi-world* settings.

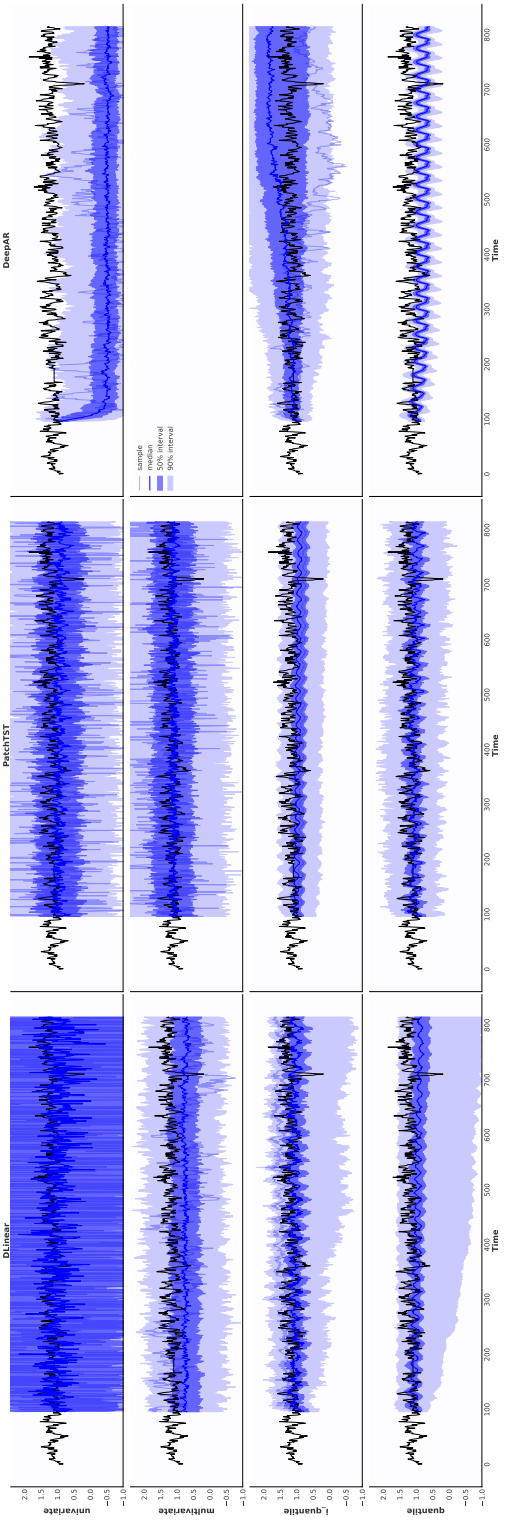
5.3. Probabilistic LTSF

In this section, we evaluate the performance of our selected models (PatchTST, DLinear and DeepAR) augmented with various probabilistic prediction heads (univariate, multivariate, quantile and IQN) on the task of probabilistic LTSF. We start with a qualitative analysis of the forecast outputs, followed by a quantitative evaluation using probabilistic scoring rules and other performance metrics.

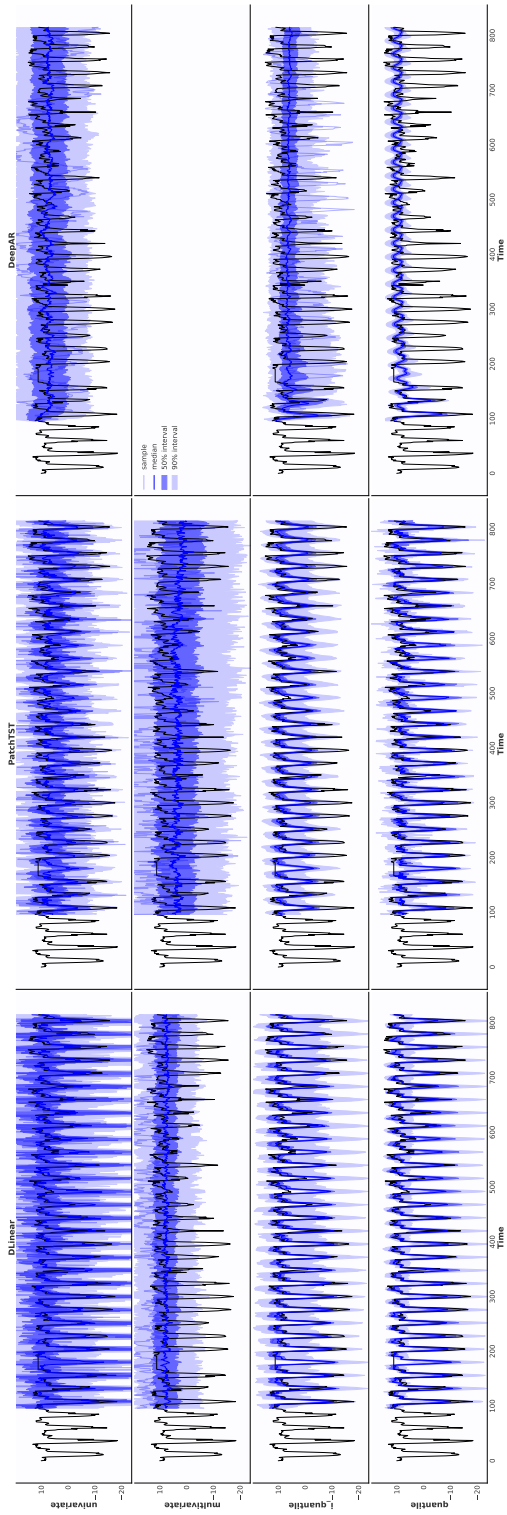
5.3.1. Qualitative Evaluation.

We begin the qualitative analysis by presenting forecast examples generated by the models trained on the ETTh1 dataset. Two representative test cases are shown: one reflecting the *multi-world* time series of Figures 5.1a and A.2 while the other exhibits a strong cyclical trend, making it potentially easier to predict. We then repeat this analysis for forecasts on the ETTm1 dataset.

ETTh1. Figures 5.6a and 5.6b display interval forecasts for two time series from the ETTh1 dataset, additionally corresponding sample forecasts are visualized in Appendix C Figure C.6 and C.7. Each individual panel shows the ground truth series in black (including input and forecast horizon), the median forecast in dark blue, a single forecast sample in light blue and shaded blue regions indicating the 50% and 90% prediction intervals. These panels are arranged by model (columns) and probabilistic head (rows). For models that produce samples (univariate, multivariate and IQN), intervals are computed empirically based on their samples. Figure 5.6a shows the predictions for the LULL time series of ETTh1, which was previously identified as a *multi-world* scenario (see Figure 5.1a). However, the detected multi-modal structure is not present throughout the test set, restricting our assessment of the *multi-world* case. Starting with DLinear, the univariate variant produces wide intervals, likely due to the heavy tails of the Student-t distribution, and a median forecast that fails to capture the temporal dynamics, with the sample forecast exhibiting unrealistic zigzagging. The multivariate DLinear model performs marginally better, with a slightly more realistic sample path and a median forecast that nonetheless underestimates the series, possibly due to the influence of the observed multi-modal patterns in the training data in Figure 5.1a. This suggests that (low-rank) joint modeling does not sufficetly and robustly address temporal structure here. The IQN variant shows stronger performance, with a median forecast that closely



(a) Interval forecasts for a selected ETTh1 series (LULL). Each panel shows the ground truth (black), the median forecast (dark blue), a single forecast sample (light blue), and shaded 50% and 90% prediction intervals. Panels are organized by model (columns) and probabilistic head (rows).

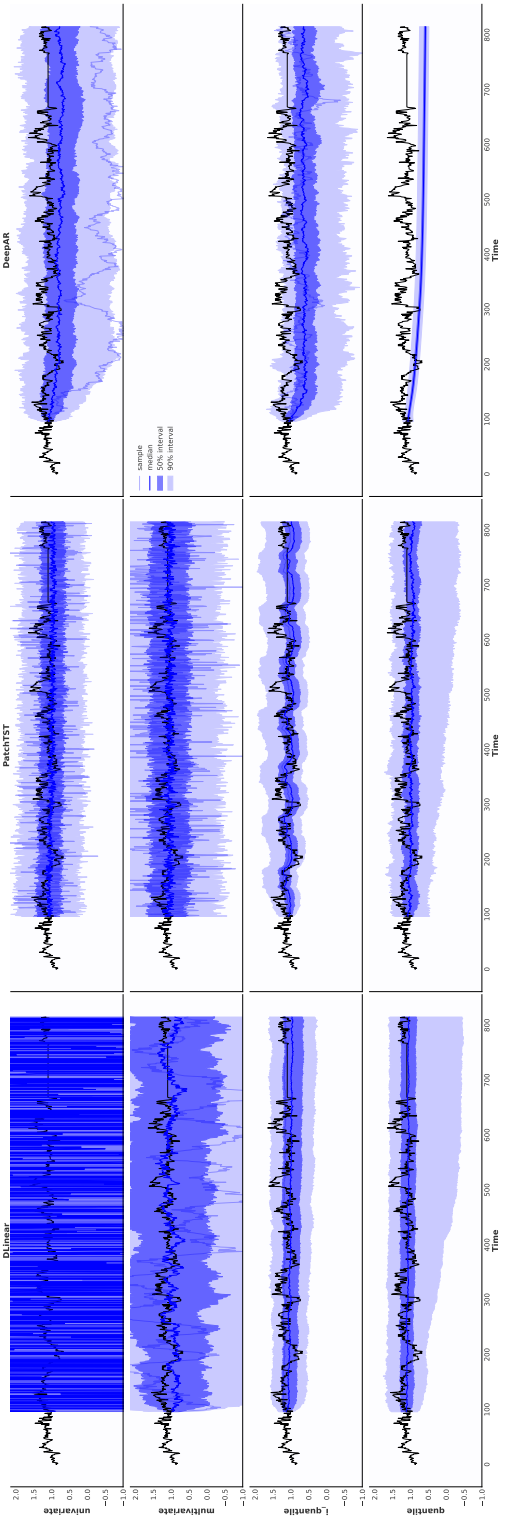


(b) Interval forecasts for a second ETTh1 series (HUFL) with strong periodic structure. Layout and color scheme are consistent with the previous figure.

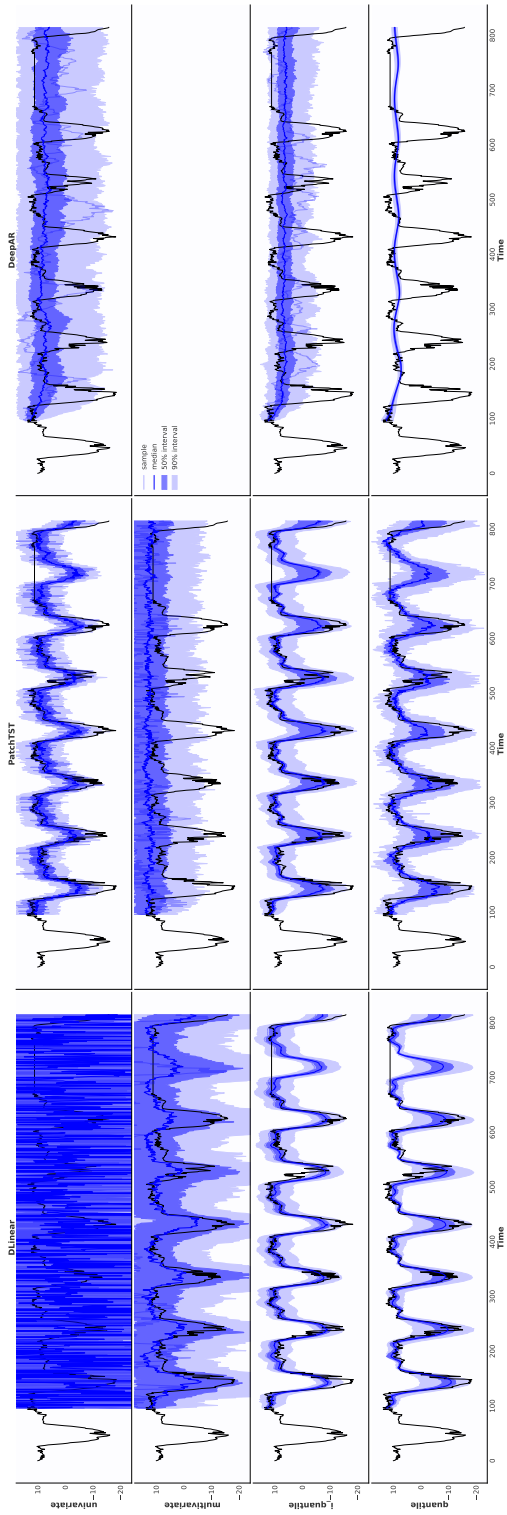
Figure 5.6.: Comparison of probabilistic interval forecasts for two different ETTh1 test series using multiple model and head configurations.

follows the ground truth, albeit with a slight downward drift. Its samples are more structured, suggesting improved temporal coherence (see Figure C.6). Quantile DLinear behaves similarly but produces smoother trajectories. PatchTST univariate forecasts are more centered, likely due to its Gaussian output assumption, though sample forecasts still show zigzag artifacts. Interestingly, even multivariate PatchTST samples display this pattern. Unlike DLinear, however, the quantile and IQN forecasts of PatchTST do not show a downward drift. Among them, the IQN forecasts appear smoother than the quantile ones. DeepAR behaves quite differently. The univariate DeepAR forecasts are heavily influenced by lower trends present during training and fail to follow the actually observed dynamics. However, its sample forecast avoids the zigzag artifact and presents a smoother trajectory (also seen across multiple samples in Figure C.6). In contrast, the DeepAR IQN model tends to drift upward, possibly due to error accumulation, while the general sample shape in Figure C.6 looks reasonable in comparison to other methods. Lastly, the quantile DeepAR model produces more stable and consistent predictions, in which forecasts do not drift far away into either direction, but still tend to drift downwards. Figure 5.6b presents a second series which, in contrast, exhibits strong periodicity. Here, DMS models generally capture the cyclical trend, with the exception of multivariate DLinear and PatchTST, which fail to capture the repeating patterns. The quantile versions of both models, however, align closely with the ground truth and produce well-calibrated intervals. While the univariate DeepAR median forecast misses the periodicity, its sample trajectory and several samples shown in Figure C.7 appear relatively realistic, though they also misalign with the ground truth periodicity. The IQN and quantile DeepAR model capture the trend in the beginning, but flatten out towards the end. Furthermore, the quantile DeepAR forecasts arguably provide the best overall performance within this model class, although their predictions are phase-shifted (e.g., predicting peaks where valleys occur) and underestimate the amplitude of the valleys.

ETTm1. Interval forecasts for two selected time series from the ETTm1 dataset are presented in Figures 5.7a and 5.7b, with corresponding sample forecasts shown in Appendix C in Figures C.9 and C.10. Although ETTm1 is generally easier to predict than ETTh1, as reflected in the larger forecastability scores in Table 5.1, some models still struggle to generate reasonable forecasts. Notably, the univariate DLinear and quantile-based DeepAR models perform poorly, with the former producing overly noisy forecasts and the latter yielding overly smooth or flat predictions, as seen in both Figures 5.7a and 5.7b. Among the two shown series, the one in Figure 5.7a, chosen due to the detected *multi-world* scenario identified in the clustering analysis (Figure 5.1b), appears more challenging. Although most models produce median forecasts that generally follow the overall trajectory, several methods, such as the quantile and IQN variants of DLinear, tend to generate overly flat median predictions, failing to capture the peaks and valleys of the ground truth series. Furthermore, the DeepAR models are an exception, as their median forecasts do not align with the ground truth to begin with; instead, they consistently underestimate the series. Overall, the multivariate DLinear and IQN PatchTST models produce the most credible forecasts in Figure 5.7a, although none



(a) Interval forecasts for a selected ETTm1 test series (LULL). Each panel shows the ground truth (black), the median forecast (dark blue), a single forecast sample (light blue), and 50% and 90% prediction intervals (shaded areas). Panels are organized by model (columns) and probabilistic head (rows).



(b) Interval forecasts for a second ETTm1 test series (HUFL) exhibiting strong cyclical patterns.

Figure 5.7.: Comparison of probabilistic interval forecasts for two ETTm1 test series across different model and probabilistic head configurations.

of the models are without flaws. Closer inspection of individual samples in Figure C.9 reveals that the univariate DeepAR model attempts to accommodate the *multi-world* nature by including sample paths that follow a lower mode. Similarly, quantile methods produce low-bound predictions that reflect this uncertainty. Again, the sample paths generated by univariate DeepAR appear more realistic compared to the zigzagging seen in univariate PatchTST and DLinear. Additionally, the multivariate DLinear model also generates comparatively plausible samples, whereas the multivariate PatchTST continues to exhibit zigzagging behavior. In contrast to Figure 5.7a, more models successfully capture the cyclical patterns in Figure 5.7b, with quantile and IQN variants of DMS methods offering particularly well-aligned forecasts. However, the univariate DLinear, multivariate PatchTST and all DLinear variants still struggle to model these repeating cycles. Interestingly, the sample forecasts from the univariate PatchTST in Figure C.10, while individually unrealistic due to zigzagging, collectively capture the underlying cyclical behavior more effectively than the univariate DeepAR samples. This highlights our expected trade-off: while IMS models tend to produce more realistic individual samples, DMS models better capture the overall long-term temporal structure. In summary, the DMS quantile and IQN models provide the most accurate and visually convincing forecasts in Figures 5.7b and C.10.

5.3.2. Quantitative Evaluation.

Table 5.4 presents the mean and standard deviation of performance metrics computed over five independent runs for each model. It includes results for all 11 model variants, evaluated on both datasets and across a range of probabilistic scoring metrics. For each run, we compute results based on 100 sampled forecasts and their emerging empirical quantiles. However, since our quantile models were trained on a limited set of quantile levels, generating samples from them would not yield representative distributions. As such, we exclude them from sample-based evaluation metrics, since their results are possibly unreliable (Marcotte et al. 2023; Bracher et al. 2021).

ETTh1. Across all quantile-based metrics, i.e. CRPS_Q, CRPS_Sum_Q, wQS, WIS, MSE and MAE, the quantile DLinear model shows the best performance. Interestingly, the quantile PatchTST and quantile DeepAR model also perform the best within their individual model category with respect to these univariate metrics. One possible reason for this is that the quantile-based methods were explicitly trained to produce forecasts at the same quantile levels used for evaluation. In contrast, the univariate, multivariate and IQN methods aim to model broader aspects of the distribution, with quantile-based metrics relying on estimated quantiles derived from samples. It is therefore not surprising that the quantile methods outperform the others, as they are directly optimized for the evaluated quantile levels. With the exception of only a few examples, the IQN methods are the second best performing methods and actually the best models with respect to the sample-based multivariate metrics. Furthermore, the IQN DLinear model performs the best according to the univariate sample-based CRPS and multivariate sample-based metrics (VS and ES). This shows that, although consistently targeting a fixed quantile

Table 5.4.: Probabilistic forecasting results for ETTh1 and ETTm1 datasets. The table reports the mean and standard deviation (±) over 5 independent runs for various probabilistic metrics, including CRPS, CRPS_Q, CRPS_Sum, CRPS_Sum_Q, VS at multiple orders, ES, wQS, WIS, MSE, and MAE. Results are shown across four prediction head types (Univariate, Multivariate, Quantile, and IQN) and three model backbones (PatchTST, DLinear, DeepAR). The best results for each metric are highlighted in bold while the second best are underlined.

	Metric	**Univariate**			**Multivariate**		**Quantile**			**IQN**		
		PatchTST	DLinear	DeepAR	PatchTST	DLinear	PatchTST	DLinear	DeepAR	PatchTST	DLinear	DeepAR
ETTh1	CRPS	1.97 ± .02	2.73 ± .01	3.18 ± 1.29	1.71 ± .05	2.67 ± .01	-	-	-	1.83 ± .04	**1.37 ± .01**	2.34 ± 0.2
	CRPS_Q	1.34 ± .01	2.18 ± .01	2.34 ± 1.18	1.19 ± .04	1.79 ± .01	1.05 ± .02	**0.87 ± .01**	2.25 ± .57	1.27 ± .03	0.96 ± .01	1.61 ± 0.15
	CRPS_Sum	8.19 ± .20	10.95 ± .03	15.96 ± 11.76	9.07 ± .37	9.25 ± .04	-	-	-	8.42 ± .48	**7.94 ± .16**	12.01 ± 3.25
	CRPS_Sum_Q	4.63 ± .06	7.64 ± .03	11.94 ± 10.33	5.78 ± .15	6.37 ± .02	4.24 ± .13	**3.85 ± .02**	6.34 ± .32	4.36 ± .04	4.16 ± .03	6.94 ± 1.27
	VS_0.5	47.23 ± .81	133.73 ± .84	73.47 ± 47.88	24.80 ± 5.97	61.94 ± .04	-	-	-	38.49 ± .88	**21.17 ± .12**	44.82 ± 11.41
	VS_1 ($\times 10^2$)	13.90 ± .37	69.68 ± .72	29.56 ± 27.12	12.91 ± 13.68	18.61 ± .10	-	-	-	8.00 ± .21	**4.55 ± .03**	11.52 ± 4.12
	VS_2 ($\times 10^6$)	0.81 ± .04	23.49 ± .45	3.31 ± 4.22	50.55 ± 96.83	0.98 ± .02	-	-	-	0.20 ± .01	**0.14 ± .00**	0.49 ± .29
	ES	7.86 ± .11	11.26 ± .04	13.69 ± 6.90	6.94 ± .29	11.19 ± .02	-	-	-	7.82 ± .18	**5.27 ± .02**	9.6 ± 1.41
	wQS	0.29 ± .00	0.47 ± .00	0.51 ± .26	0.26 ± .01	0.39 ± .00	0.23 ± .00	**0.19 ± .00**	0.49 ± .12	0.28 ± .01	0.21 ± .00	0.35 ± 0.03
	WIS	1.34 ± .01	2.18 ± .01	2.34 ± 1.18	1.19 ± .04	1.79 ± .01	1.05 ± .02	**0.87 ± .01**	2.25 ± .57	1.27 ± .03	0.96 ± .01	1.61 ± 0.15
	MSE	18.84 ± .76	12.23 ± .03	58.55 ± 55.17	17.29 ± .58	40.22 ± .05	13.51 ± .19	**10.24 ± .18**	26.44 ± 8.80	17.10 ± .49	10.57 ± .05	29.31 ± 7.84
	MAE	2.71 ± .04	2.06 ± .00	4.12 ± 1.37	2.35 ± .03	3.83 ± .00	2.21 ± .04	**1.77 ± .03**	3.10 ± .62	2.52 ± .05	1.83 ± .01	3.23 ± 0.23
	# Parameters ($\times 10^6$)	1.44	1.96	0.21	21.06	188.06	4.69	33.68	0.22	0.03	4.96	0.07
	Epoch Time (s)	10.71 ± 2.82	14.02 ± 0.07	75.14 ± 19.83	397.23 ± 2.64	1,086.71 ± 3.05	23.30 ± 1.57	3.63 ± 1.89	163.29 ± 35.29	2.11 ± 0.43	6.03 ± 3.60	25.19 ± 4.47
	# Epochs	17 ± 6.1	16.6 ± 3.5	18.8 ± 4.83	19.8 ± 16.65	11 ± 0.0	17 ± 1.57	18.2 ± 4.66	41.2 ± 32.26	16.6 ± 2.15	18.6 ± 3.56	44.4 ± 17.45
ETTm1	CRPS	1.55 ± .02	37.46 ± 4.34	3.35 ± 1.13	2.17 ± .00	3.12 ± .12	-	-	-	1.4 ± .03	**1.27 ± .01**	3.61 ± 1.20
	CRPS_Q	1.08 ± .02	29.45 ± 3.42	2.42 ± .92	1.57 ± .00	2.36 ± .1	1.09 ± .04	**0.83 ± .00**	2.36 ± 0.19	0.98 ± .02	0.95 ± .01	2.60 ± 1
	CRPS_Sum	**6.32 ± .19**	155.16 ± 21.62	11.39 ± 5.31	25.86 ± .17	11.77 ± .40	-	-	-	6.50 ± .19	9.52 ± .13	19.1 ± 12.28
	CRPS_Sum_Q	4.44 ± .08	115.98 ± 13.39	8.37 ± 3.5	9.1 ± .02	8.37 ± .43	4.29 ± .06	**3.91 ± .01**	9.96 ± 1.39	4.19 ± .01	4.73 ± .05	13.64 ± 9.1
	VS_0.5	28.01 ± .83	4,964.18 ± 653.96	78.05 ± 45.09	26.36 ± .03	157.26 ± 15.17	-	-	-	25.55 ± 1.24	**17.7 ± .29**	72.85 ± 32.20
	VS_1 ($\times 10^2$)	6.41 ± .23	24,244.89 ± 4,061.11	29.71 ± 24.98	7.66 ± .02	91.05 ± 13.57	-	-	-	5.29 ± .26	**3.91 ± .07**	24.93 ± 19.42
	VS_2 ($\times 10^6$)	0.21 ± .01	1,269,123.25 ± 252,738.55	2.97 ± 3.42	0.48 ± .00	23.12 ± 6.25	-	-	-	0.16 ± .01	**0.13 ± .00**	2.11 ± 2.82
	ES	6.19 ± .12	142.24 ± 16.48	13.25 ± 5.26	8.90 ± .01	12.92 ± .58	-	-	-	5.93 ± 0.14	**4.93 ± .06**	14.98 ± 5.58
	wQS	0.23 ± .00	6.39 ± .74	0.52 ± .2	0.34 ± .00	0.51 ± .02	0.24 ± .01	**0.18 ± .00**	0.51 ± .04	0.21 ± .01	0.21 ± .00	0.56 ± .22
	WIS	1.08 ± .02	29.45 ± 3.42	2.42 ± .92	1.57 ± .00	2.36 ± .1	1.09 ± .04	**0.83 ± .00**	2.36 ± .19	0.98 ± .02	0.95 ± .01	2.6 ± 1
	MSE	14.08 ± .49	2,625.69 ± 482.95	59.76 ± 41.5	32.84 ± .05	22.31 ± 1.19	13.89 ± .87	**10.20 ± .03**	21.03 ± 0.83	12.18 ± .27	10.32 ± .08	69.48 ± 49.89
	MAE	2.15 ± .04	30.22 ± 3.18	4.43 ± 1.41	2.84 ± .00	2.80 ± .08	2.16 ± .10	**1.61 ± .00**	2.81 ± .2	1.93 ± .03	1.65 ± .01	4.8 ± 1.45
	# Parameters ($\times 10^6$)	3.42	1.18	0.01	2.78	139.06	70.1	33.68	0.23	0.3	5.31	0.08
	Epoch Time (s)	20.03 ± .04	5.5 ± 1.82	490.46 ± 212.38	1,713.74 ± 8.66	796.62 ± 9.00	32.61 ± 1.47	15.84 ± 10	390.76 ± 155.84	7.48 ± .06	8.63 ± 7.23	326.85 ± 202.91
	# Epochs	15.0 ± 1.67	12.0 ± .89	14.0 ± 2.28	11.4 ± 1.36	16.8 ± 5.38	19.6 ± 9.22	14.2 ± 1.17	51.8 ± 16.04	15.4 ± 3.93	16.2 ± 2.4	11.8 ± 2.99

level across time steps may not represent true random draws from the predictive distribution, they effectively capture the structural shape of the forecast trajectory more precisely. In comparison, the IQN PatchTST model performs slightly worse than the multivariate PatchTST model in terms of VS_0.5 and ES, while the IQN DeepAR manages to outperform the quantile DeepAR method in certain metrics. Interestingly, the samples generated by the univariate DeepAR model, albeit modeling temporal dynamics between time steps, perform worse than their DMS counterparts with respect to the VS and ES metrics. This is likely due to the weaknesses of IMS methods in LTSF, where error accumulation leads to degraded performance. As a result, even relatively clear cyclical patterns are not modeled consistently (see Figure C.7). In order to compare our performances with those of original models, we evaluated the MSE and MAE of our median forecasts and compared them with the results published in the original point LTSF studies. However, Nie et al. (2022) and Zeng et al. (2023) report the MSE and MAE results on normalized data, which tends to result in seemingly low error scores, a point criticized by Shao et al. (2025). As an alternative, Shao et al. (2025) propose evaluating model performances on re-normalized data to allow for more interpretable comparisons. However, Shao et al. (2025) only report best-case performance for a forecast horizon of $H = 336$, with varying look-back windows of $L = 96, 192, 336$ or 720. As a result, their reported results provide only a rough estimate of overall model performance on a forecasting horizon of $H = 720$. In their evaluation on ETTh1, they report an MAE (MSE) of 1.94 (11.83), 1.60 (9.49), and 1.58 (9.36) for DeepAR, PatchTST, and DLinear, respectively. Our best DLinear model achieves relatively comparable performance with an MAE (MSE) of 1.77 (10.24), while predicting a longer forecast horizon than considered in their setup. In contrast, our best PatchTST and DeepAR models show worse MAE and MSE scores than those reported by Shao et al. (2025). Looking at the descriptive statistics of the number of parameters and epoch durations for ETTh1 in Table 5.4, we can see that multivariate methods have the largest number of parameters. This also leads to them having the longest training times overall. Interestingly, among the probabilistic approaches, the DLinear variants tend to have the highest parameter counts, whereas the PatchTST and DeepAR models are often significantly smaller in comparison, especially for the quantile and IQN heads. Despite DeepAR models being among the smallest in terms of parameter size, their recurrent operations result in them having the longest per-epoch runtime (excluding the multivariate variants) while they also require the highest number of training epochs to converge. Notably, the overall smallest model is the IQN PatchTST variant. To independently assess calibration and sharpness, Figure 5.8 presents the quantile-quantile (QQ) plots for each model in the first three panels, followed by the sharpness plot on the right. In the QQ plots, the x-axis denotes the nominal quantile levels, while the y-axis represents the empirical coverage. Ideally, a well-calibrated model produces a coverage curve that closely follows the diagonal, indicating accurate quantile estimates. The color and line styles differentiate the model types: blue, orange, and red correspond to DLinear, PatchTST, and DeepAR, respectively. Solid lines represent the univariate models, dashed lines indicate multivariate models, dash-dotted lines correspond to IQN variants, and dotted lines represent quantile models. While a combined coverage plot for all models is provided in the appendix

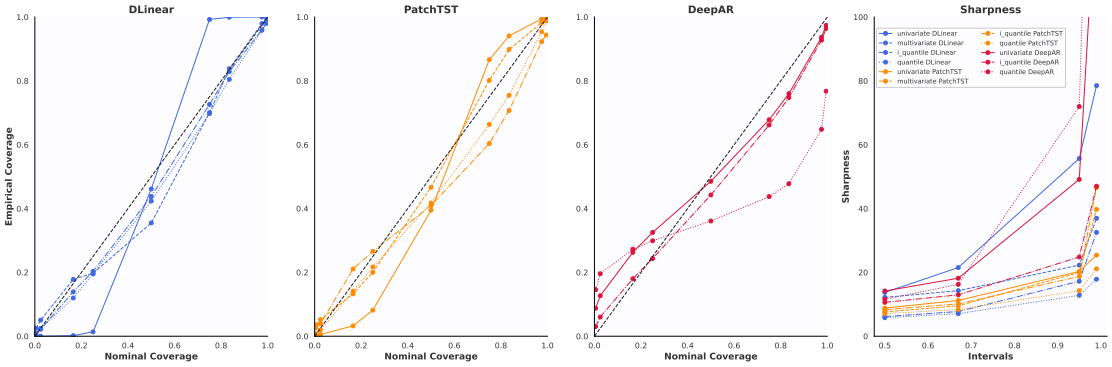


Figure 5.8.: Quantile calibration (left three panels) and interval sharpness (right panel) curves for all model variants on the ETTh1 dataset. The calibration plots show empirical coverage versus nominal quantile levels; ideal calibration aligns with the diagonal. Line color and style encode model architecture and variant type: blue, orange, and red represent DLinear, PatchTST, and DeepAR; solid and dashed lines denote univariate and multivariate models; dash-dotted and dotted lines indicate IQN and quantile variants, respectively. Sharpness curves summarize average prediction interval width across intervals, where narrower intervals indicate sharper forecasts.

(see Figure C.12), we opt to show the separated versions here due to visual clutter in the aggregated view. Starting with the DLinear models on the left of Figure 5.8, we observe that the univariate version exhibits the worst calibration, particularly underestimating coverage at lower quantile levels and overestimating it at higher ones. The median calibration, however, aligns closest with the diagonal, indicating relatively accurate central forecasts. This behavior is also evident in Figure 5.6, where the median forecast lies near the ground truth, yet the 50% prediction interval (spanning the 25% to 75% quantiles) is already excessively wide. In contrast, the multivariate DLinear model shows improved calibration in the tails but performs poorly around the central quantile levels. This is reflected in its median forecasts being slightly misaligned with the actual series in Figure 5.6, while the interval bounds remain tighter and less prone to the downward drift seen in quantile-based methods. Among the DLinear variants, the IQN model demonstrates the best overall calibration, closely followed by the quantile model. Regarding PatchTST, the univariate model performs worst across nearly all quantile levels. Unlike the well-calibrated DLinear IQN model, the PatchTST IQN variant suffers from poor calibration, particularly at higher quantiles, where it tends to predict bounds that are systematically too low. This undercoverage is visible in the upper interval bounds of Figure 5.6a. The multivariate PatchTST model arguably shows the most balanced calibration among the PatchTST variants. Lastly, the DeepAR models again exhibit distinct behavior. The univariate DeepAR model shows the best calibration within its class; however, the IQN variant achieves better calibration at lower nominal quantile levels. The quantile DeepAR model performs the worst in terms of calibration, not just

within the DeepAR family, but across all model classes. It is particularly poorly calibrated at higher nominal quantile levels, consistently underestimating them. This is also partly visible in Figure C.6, where the upper forecasts of the quantile DeepAR model fail to capture all values, despite representing a high nominal quantile. Turning to sharpness, the rightmost panel of Figure 5.8 aggregates all models into a single plot, as their sharpness curves are visually more distinguishable. The sharpest forecasts, those with the narrowest prediction intervals, are produced by the quantile DLinear and quantile PatchTST models. In contrast, the quantile DeepAR model produces the widest prediction intervals. This seems counterintuitive when examining Figures 5.6a, 5.6b, C.6, and C.7, where both the intervals and sample forecasts of quantile DeepAR appear comparatively narrow to other methods. Considering the interplay between calibration and sharpness, among the DLinear models, the quantile and IQN variants demonstrate the best calibration and sharpness. For PatchTST, the quantile model exhibits the highest sharpness, whereas the multivariate version shows the best calibration. Finally, the IQN DeepAR model achieves the most balanced performance in terms of both sharpness and calibration, outperforming its univariate and quantile variants.

ETTm1. Table 5.4 reveals that the univariate DLinear model performs notably bad (in line with Figure 5.7a and 5.7b) and appears to be an outlier, so we exclude it from meaningful comparisons. In contrast, the best-performing method is again quantile DLinear, consistent with observations from ETTh1. The second-best model is IQN PatchTST, closely followed by IQN DLinear. Among the DMS models, multivariate variants perform the worst, especially on multivariate metrics, indicating the inherent difficulty in modeling temporal dependencies directly. Next, it is worth noting that multivariate PatchTST outperforms all DeepAR variants, suggesting that, as expected, DMS models surpass IMS approaches in capturing long-term temporal dependencies. With the exception of PatchTST, where the IQN outperforms the quantile method, IQN models again tend to perform slightly worse than quantile counterparts across the board. When examining MAE and MSE performance, we observe a noticeable performance drop compared to the results reported in Shao et al. (2025), where DeepAR, PatchTST, and DLinear achieved MAE (MSE) scores of 2.21 (16.16), 1.37 (7.73), and 1.38 (7.84), respectively. In our evaluations, the corresponding scores are DeepAR 2.81 (21.03), PatchTST 1.93 (12.18), and DLinear 1.61 (10.2). While this indicates a degradation in performance, it can be attributed to our use of a longer forecasting horizon, making the task inherently more challenging. In terms of computational resources, the largest models are again the DMS methods. The multivariate DLinear model stands out as the largest, followed by the quantile PatchTST model. Correspondingly, multivariate methods also require the longest training times, while DeepAR models tend to have the longest training duration within each probabilistic head class, reaffirming the inefficiency of their recurrent operations compared to more recent alternatives. Lastly, we again analyze sharpness and coverage separately in Figure 5.9. On the left, we observe that the quantile and IQN DLinear models are well calibrated, closely aligning with the diagonal. In contrast, the univariate and multivariate DLinear models show poor calibration, with

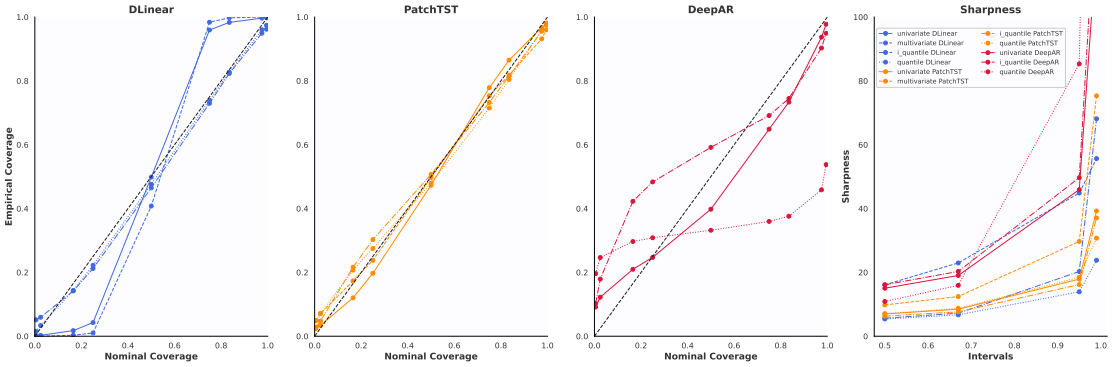


Figure 5.9.: QQ plots to evaluate calibration (left three panels) and interval sharpness curves (right panel) for all model variants on the ETTm1 dataset. Sharpness curves summarize average prediction interval width across intervals, where narrower intervals indicate sharper forecasts.

under-coverage at low quantiles and over-coverage at higher ones. This is also reflected in the wide gaps between the upper and lower bounds of their forecast intervals, as seen in Figures 5.7a and 5.7b. Interestingly, despite the insights from those interval plots, the univariate DLinear model is better calibrated than the multivariate version and provides the best calibrated median forecast among all DLinear variants. Regarding PatchTST models, all variants are very well calibrated, closely following the diagonal in the calibration plot. In stark contrast, DeepAR models, particularly the quantile and IQN versions, deviate significantly from the diagonal, indicating poor calibration. In addition to this, the sharpness plot reveals that the quantile DLinear model produces the sharpest forecast intervals, followed by the IQN PatchTST model. The univariate DLinear model is excluded from this plot due to its excessively large sharpness values, which aligns with the extreme interval widths observed in Figures 5.7a and 5.7b. Outside of this, the DeepAR models also exhibit relatively wide forecast intervals, indicating less confident predictions. Altogether, this analysis is consistent with our findings in Table 5.4, where the quantile DLinear and IQN PatchTST models were ranked as the top-performing methods.

5.3.3. Key Takeaways

Our experiments on probabilistic LTSF yield two key insights. First, the results affirm the strength of DMS methods in long-term forecasting, demonstrating their superior ability to model long-range dependencies in probabilistic LTSF tasks by consistently outperforming the IMS model. However, a closer inspection of the sample quality reveals a notable shortcoming: DMS models often produce zigzagging and unrealistic forecast samples, due to a lack of learning conditional temporal dependencies between predictions (see Figure C.9). Attempts to address this through low-rank multivariate Gaussian output heads yielded only marginal improvements in sample quality, e.g. in multivariate DLinear (Figure C.10), and in some cases, such as multivariate PatchTST (Figure C.9),

it showed no improvement at all. Moreover, incorporating multivariate prediction heads consistently degraded performance across DMS models. In contrast, IMS methods, while generally underperforming in terms of standard and probabilistic metrics, tend to generate more realistic and coherent forecast samples. This exposes a fundamental trade-off: DMS models excel in forecasting accuracy, but IMS models provide better sample quality, particularly when temporal consistency and structure are important. Second, our results consistently show that quantile-based forecasting methods outperform parametric distributional approaches across several backbone architectures. A key reason for this is that quantile methods make fewer assumptions about the underlying output distribution, offering greater flexibility in capturing diverse data behaviors. Additionally, they are more closely aligned with the evaluation procedure, as the same quantiles predicted during training are used for testing. This performance advantage is further rooted in the compatibility between quantile-based training and the original objectives of point LTSF models. In fact, most LTSF DMS models were initially optimized using loss functions like MAE or MSE (Zhou et al. 2021; Nie et al. 2022; Zeng et al. 2023), and notably, the quantile loss at the 0.5 quantile level is mathematically equivalent to MAE (Gneiting and Katzfuss 2014), making quantile-based training naturally compatible with these original objectives. This makes quantile training not only intuitive but also better aligned with the original optimization goal of the backbone point LTSF model. Moreover, as argued by Han et al. (2024), MAE is a robust objective that is particularly well-suited for long-term forecasting tasks, further reinforcing the advantage of quantile-based approaches in this setting.

6. Conclusions

This thesis investigated the intersection of point-based LTSF and probabilistic forecasting, with a particular emphasis on DMS and IMS decoding strategies. We began by surveying the rapidly evolving landscape of LTSF and probabilistic TSF, identifying that recent LTSF methods predominantly adopt a DMS strategy. However, this strategy neglects temporal coherence in probabilistic settings due to their conditionally independent nature. To investigate this further, we introduced the concept of *multi-world* scenarios to highlight limitations of DMS decoding in capturing joint temporal uncertainty. Furthermore, a simple synthetic *multi-world* example demonstrated that IMS-based models produce more coherent sample trajectories, whereas DMS models fail to provide suitable sample forecasts. In addition to this, we investigated the transition of point LTSF to probabilistic LTSF by extending several state-of-the-art LTSF models with probabilistic forecasting techniques, including distributional forecasting, quantile regression and IQNs. Here, the superior LTSF ability of DMS models was affirmed on a probabilistic basis, despite producing poor quality samples. Furthermore, in our analysis, quantile-based approaches consistently outperformed both univariate and multivariate parametric distributional forecasting methods.

Outlook. A first straightforward extension of this work lies in expanding the scope and depth of the probabilistic LTSF evaluation. While the BasicTS+ benchmark (Shao et al. 2025) includes a broad range of datasets and LTSF models, our study implemented only a subset of them. Evaluating a more comprehensive set of models and datasets could enhance the robustness and generalizability of our findings. Additionally, exploring a wider variety of look-back window lengths L and forecasting horizons H, including those used in the original LTSF studies (Zhou et al. 2021; Nie et al. 2022), would allow for a more thorough assessment. Extending this further to ultra-long forecasting scenarios, as investigated in recent works such as Jia et al. (2023) and Shang et al. (2024), may uncover additional challenges and insights. Given that the introduction of *multi-world* scenarios is another central contribution of this work, further analysis beyond the initial synthetic example presents a promising direction. A natural next step would be to design more sophisticated synthetic experiments that allow evaluation over extended forecasting horizons, facilitating the development of methods to improve DMS performance under such conditions. While traces of *multi-world* dynamics have been observed in real-world LTSF datasets, establishing a general paradigm for identifying and classifying *multi-world* scenarios in arbitrary real-world datasets could prove valuable, particularly in detecting when probabilistic DMS methods are likely to fail. For instance, developing improved metrics or clustering techniques, e.g. based on Gaussian Mixture Models (McDowell et al. 2018; Zhang et al. 2021), may help uncover and characterize

underlying multi-modal structures more effectively. Another natural extension of this work is to leverage the gained insights to improve the underlying DMS and IMS models. One promising direction is to explore architectural enhancements, such as hybrid DMS-IMS models where one strategy is layered on top of the other. This idea is conceptually related to SMARTformer (Li et al. 2023), which generates subsequences iteratively but refines the full sequence in a non-autoregressive manner. Alternatively, ensemble methods that combine forecasts from both DMS and IMS models could be explored, similar to the multi-model distribution ensemble approach proposed by Zhou et al. (2023). Such combinations may offer robustness by leveraging the complementary strengths of both strategies. On a related note, in Section 3.2, we presented various probabilistic forecasting approaches, though our study only considered more basic methods. Extending this work to incorporate recent advances in probabilistic time series forecasting would be another natural next step. For example, adopting flexible density estimation techniques (Drouin et al. 2022; Bergsma et al. 2022; Ashok et al. 2023; Bergsma et al. 2023) instead of fixed parametric forms could allow our models to better capture complex multi-modal structures, although this may not necessarily translate into improved performances of DMS models in *multi-world* scenarios (see Figure 4.6). To improve this instead, DMS methods could benefit from more sophisticated multivariate distributional modeling. Approaches such as modeling multivariate temporal dependencies via copula models, as in TACTiS and TACTiS-2 (Drouin et al. 2022; Ashok et al. 2023), offer increased flexibility in capturing joint behaviors. Similarly, normalizing flows built upon our low-rank multivariate Gaussian base distribution can also enhance this flexibility (Feng et al. 2024; Rasul et al. 2020; Drouin et al. 2022; Ashok et al. 2023). Alternatively, generative methods, including diffusion models (Rasul et al. 2021; Shen and Kwok 2023; Shen et al. 2023; Li et al. 2023), VAEs (Li et al. 2022; Feng et al. 2024; Tong et al. 2022) and GANs (Wu et al. 2020; Koochali et al. 2021), also enable the estimation of highly expressive data distributions without being constrained by fixed parametric assumptions. Motivated by the strong performance of quantile-based methods, the GQ-Former approach (Jawed and Schmidt-Thieme 2022) presents a promising direction for exploration in *multi-world* scenarios. Its novel multi-task loss promotes both sharpness and diversity in quantile estimates, enabling the model to capture multiple modes within complex multivariate joint distributions. Alternatively, modeling the multivariate quantile function could improve upon the simpler quantile-based methods considered here, for example taking inspiration from MQF^2 (Kan et al. 2022), providing richer and more flexible uncertainty quantification.

Bibliography

Albahli, S. (2025, January). LSTM vs. Prophet: Achieving Superior Accuracy in Dynamic Electricity Demand Forecasting. *Energies 18*(2), 278. Publisher: Multidisciplinary Digital Publishing Institute.

Alcaraz, J. L. and N. Strodthoff (2022, December). Diffusion-based Time Series Imputation and Forecasting with Structured State Space Models. *Transactions on Machine Learning Research*.

Alexander, C., M. Coulon, Y. Han, and X. Meng (2024, March). Evaluating the discrimination ability of proper multi-variate scoring rules. *Annals of Operations Research 334*(1), 857–883.

Alexandrov, A., K. Benidis, M. Bohlke-Schneider, V. Flunkert, J. Gasthaus, T. Januschowski, D. C. Maddix, S. Rangapuram, D. Salinas, J. Schulz, L. Stella, A. C. Türkmen, and Y. Wang (2020). GluonTS: Probabilistic and Neural Time Series Modeling in Python. *Journal of Machine Learning Research 21*(116), 1–6.

Almeida, A., A. Loy, and H. Hofmann (2018, December). ggplot2 Compatible Quantile-Quantile Plots in R. *The R Journal*.

Amos, B., L. Xu, and J. Z. Kolter (2017, July). Input Convex Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 146–155. PMLR. ISSN: 2640-3498.

Anh, D. T. and L. H. Thanh (2015). An efficient implementation of k-means clustering for time series data with DTW distance. *International Journal of Business Intelligence and Data Mining 10*(3), 213–232.

Ashok, A., Marcotte, V. Zantedeschi, N. Chapados, and A. Drouin (2023, October). TACTiS-2: Better, Faster, Simpler Attentional Copulas for Multivariate Time Series. In *Proceedings of the Twelfth International Conference on Learning Representations*.

Bai, S., J. Z. Kolter, and V. Koltun (2018, April). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. arXiv:1803.01271.

Bengio, Y., P. Simard, and P. Frasconi (1994, March). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks 5*(2), 157–166.

Benidis, K., S. S. Rangapuram, V. Flunkert, Y. Wang, D. Maddix, C. Turkmen, J. Gasthaus, M. Bohlke-Schneider, D. Salinas, L. Stella, F.-X. Aubet, L. Callot, and T. Januschowski (2022, December). Deep Learning for Time Series Forecasting: Tutorial and Literature Survey. *ACM Computing Surveys 55*(6). Place: New York, NY, USA Publisher: Association for Computing Machinery.

Bergsma, S., T. Zeyl, and L. Guo (2023). SutraNets: Sub-series Autoregressive Networks for Long-Sequence, Probabilistic Forecasting. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), *Advances in Neural Information Processing Systems*, Volume 36, pp. 30518–30533. Curran Associates, Inc.

Bergsma, S., T. Zeyl, J. Rahimipour Anaraki, and L. Guo (2022). C2FAR: Coarse-to-Fine Autoregressive Networks for Precise Probabilistic Forecasting. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), *Advances in Neural Information Processing Systems*, Volume 35, pp. 21900–21915. Curran Associates, Inc.

Bernardo, J. M. (1979, May). Expected Information as Expected Utility. *The Annals of Statistics 7*(3), 686–690. Publisher: Institute of Mathematical Statistics.

Biewald, L. (2020). Experiment Tracking with Weights and Biases. URL: https://www.wandb.com/.

Biloš, M., K. Rasul, A. Schneider, Y. Nevmyvaka, and S. Günnemann (2023, July). Modeling temporal data as continuous functions with stochastic process diffusion. In *Proceedings of the 40th International Conference on Machine Learning*, Volume 202 of *ICML'23*, Honolulu, Hawaii, USA, pp. 2452–2470. PMLR.

Bishop, C. M. (2019). *Pattern recognition and machine learning*. Information Science and Statistics. New York, NY: Springer Science+Business Media, LLC. OCLC: 1334664824.

Bontempi, G., S. Ben Taieb, and Y.-A. Le Borgne (2013). Machine Learning Strategies for Time Series Forecasting. In M.-A. Aufaure and E. Zimányi (Eds.), *Business Intelligence: Second European Summer School, eBISS 2012, Brussels, Belgium, July 15-21, 2012, Tutorial Lectures*, pp. 62–77. Berlin, Heidelberg: Springer.

Borovykh, A., S. Bohte, and C. W. Oosterlee (2018, September). Conditional Time Series Forecasting with Convolutional Neural Networks. arXiv:1703.04691.

Bothwell, S., A. Kaizer, R. Peterson, D. Ostendorf, V. Catenacci, and J. Wrobel (2022, October). Pattern-Based Clustering of Daily Weigh-In Trajectories Using Dynamic Time Warping. *Biometrics 79*(3), 2719–2731. https://academic.oup.com/biometrics/article-pdf/79/3/2719/56040291/biometrics_79_3_2719.pdf.

Box, G. (2013). Box and Jenkins: Time Series Analysis, Forecasting and Control. In T. C. Mills (Ed.), *A Very British Affair: Six Britons and the Development of Time*

Series Analysis During the 20th Century, pp. 161–215. London: Palgrave Macmillan UK. ISBN: 978-1-137-29126-4.

Box, G. E., G. M. Jenkins, G. C. Reinsel, and G. M. Ljung (2015). *Time series analysis: forecasting and control*. Wiley Series in Probability and Statistics. John Wiley & Sons.

Box, G. E. P. and D. A. Pierce (1970, December). Distribution of Residual Autocorrelations in Autoregressive-Integrated Moving Average Time Series Models. *Journal of the American Statistical Association 65*(332), 1509–1526.

Bracher, J., E. L. Ray, T. Gneiting, and N. G. Reich (2021, December). Evaluating epidemic forecasts in an interval format. *PLOS Computational Biology 17*(2), e1008618. Publisher: Public Library of Science.

Brown, T., B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei (2020). Language Models are Few-Shot Learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), *Advances in Neural Information Processing Systems*, Volume 33, Vancouver, Canada, pp. 1877–1901. Curran Associates, Inc.

Cai, W., Y. Liang, X. Liu, J. Feng, and Y. Wu (2024, March). MSGNet: Learning Multi-Scale Inter-series Correlations for Multivariate Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 38, pp. 11141–11149.

Cao, Y.-P. and H.-F. Wang (2017, January). A GPU Heterogeneous Cluster Scheduling Model for Preventing Temperature Heat Island. *ITM Web of Conferences 11*, 07003.

Cervera, J. L. and J. Muñoz (1996, May). Proper Scoring Rules for Fractiles. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith (Eds.), *Bayesian Statistics 5: Proceedings of the Fifth Valencia International Meeting*. Oxford University Press. ISBN: 978-0-19-852356-7.

Challu, C., K. G. Olivares, B. N. Oreshkin, F. G. Ramirez, M. M. Canseco, and A. Dubrawski (2023, June). NHITS: Neural Hierarchical Interpolation for Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 37, pp. 6989–6997.

Chen, J.-Q., Y.-L. He, Y.-C. Cheng, P. Fournier-Viger, and J. Z. Huang (2024, June). A multiple kernel-based kernel density estimator for multimodal probability density functions. *Engineering Applications of Artificial Intelligence 132*, 107979.

Chen, M., L. Shen, H. Fu, Z. Li, J. Sun, and C. Liu (2024, August). Calibration of Time-Series Forecasting: Detecting and Adapting Context-Driven Distribution Shift. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and*

Data Mining, KDD '24, New York, NY, USA, pp. 341–352. Association for Computing Machinery.

Chen, P., Y. Zhang, Y. Cheng, Y. Shu, Y. Wang, Q. Wen, B. Yang, and C. Guo (2023, October). Pathformer: Multi-scale Transformers with Adaptive Pathways for Time Series Forecasting. In *International Conference on Learning Representations*.

Chen, S.-A., C.-L. Li, S. O. Arik, N. C. Yoder, and T. Pfister (2023). TSMixer: An All-MLP Architecture for Time Series Forecasting. *Transactions on Machine Learning Research*.

Chen, Y., Y. Kang, Y. Chen, and Z. Wang (2019, August). Probabilistic forecasting with temporal convolutional neural network. In *MileTS '19: 5th KDD Workshop on Mining and Learning from Time Series*, Anchorage, Alaska, USA, pp. 11. ACM.

Chen, Z., S. Feng, Z. Zhang, X. Xiao, X. Gao, and P. Zhao (2024, December). SDformer: Similarity-driven Discrete Transformer For Time Series Generation. In *Advances in Neural Information Processing Systems*, Volume 37, pp. 132179–132207. Curran Associates, Inc.

Chen, Z., A. Gaba, I. Tsetlin, and R. L. Winkler (2022). Evaluating quantile forecasts in the M5 uncertainty competition. *International Journal of Forecasting 38*(4), 1531–1545.

Chevillon, G. (2007). Direct Multi-Step Estimation and Forecasting. *Journal of Economic Surveys 21*(4), 746–785. https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-6419.2007.00518.x.

Chevillon, G. and D. F. Hendry (2005, April). Non-parametric direct multi-step estimation for forecasting economic processes. *International Journal of Forecasting 21*(2), 201–218.

Cho, K., B. v. Merrienboer, Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, pp. 1724–1734. Association for Computational Linguistics.

Christoffersen, P. F. (1998). Evaluating Interval Forecasts. *International Economic Review 39*(4), 841–862. Publisher: [Economics Department of the University of Pennsylvania, Wiley, Institute of Social and Economic Research, Osaka University].

Chu, S., E. Keogh, D. Hart, and M. Pazzani (2002, April). Iterative Deepening Dynamic Time Warping for Time Series. In *Proceedings of the 2002 SIAM International Conference on Data Mining (SDM)*, Proceedings, pp. 195–212. Society for Industrial and Applied Mathematics.

Cirstea, R.-G., C. Guo, B. Yang, T. Kieu, X. Dong, and S. Pan (2022, July). Triformer: Triangular, Variable-Specific Attentions for Long Sequence Multivariate Time Series Forecasting. In *International Joint Conference on Artificial Intelligence*, Volume 3, pp. 1994–2001. ISSN: 1045-0823.

Cleveland, R. B., W. S. Cleveland, J. E. McRae, and I. Terpenning (1990). STL: A Seasonal-Trend Decomposition. *Journal of Official Statistics 6*(1), 3–73.

Cox, D. R. (1961). Prediction by Exponentially Weighted Moving Averages and Related Methods. *Journal of the Royal Statistical Society: Series B (Methodological) 23*(2), 414–422. https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.2517-6161.1961.tb00424.x.

Cressie, N. (1985, July). Fitting variogram models by weighted least squares. *Journal of the International Association for Mathematical Geology 17*(5), 563–586.

Dabney, W., G. Ostrovski, D. Silver, and R. Munos (2018, July). Implicit Quantile Networks for Distributional Reinforcement Learning. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1096–1105. PMLR. ISSN: 2640-3498.

Das, A., W. Kong, A. Leach, S. K. Mathur, R. Sen, and R. Yu (2023). Long-term Forecasting with TiDE: Time-series Dense Encoder. *Transactions on Machine Learning Research*.

Dasgupta, S. and S. Kpotufe (2014). Optimal rates for k-NN density and mode estimation. In *Advances in Neural Information Processing Systems*, Volume 27. Curran Associates, Inc.

Davies, D. L. and D. W. Bouldin (1979). A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-1*(2), 224–227.

Dawid, A. P. and M. Musio (2014, August). Theory and applications of proper scoring rules. *METRON 72*(2), 169–183.

de Bézenac, E., S. S. Rangapuram, K. Benidis, M. Bohlke-Schneider, R. Kurle, L. Stella, H. Hasson, P. Gallinari, and T. Januschowski (2020). Normalizing Kalman Filters for Multivariate Time Series Analysis. In *Advances in Neural Information Processing Systems*, Volume 33, pp. 2995–3007. Curran Associates, Inc.

De Gooijer, J. G. and R. J. Hyndman (2006, January). 25 years of time series forecasting. *International Journal of Forecasting 22*(3), 443–473.

Deisenroth, M. P., R. D. Turner, M. F. Huber, U. D. Hanebeck, and C. E. Rasmussen (2012, July). Robust Filtering and Smoothing with Gaussian Processes. *IEEE Transactions on Automatic Control 57*(7), 1865–1871.

Deliu, N. and B. Liseo (2024, August). Alternative Approaches for Estimating Highest-Density Regions. *International Statistical Review*, insr.12592. Publisher: John Wiley & Sons Ltd.

Deng, J., F. Ye, D. Yin, X. Song, I. Tsang, and H. Xiong (2024, December). Parsimony or Capability? Decomposition Delivers Both in Long-term Time Series Forecasting. In *Advances in Neural Information Processing Systems*, Volume 37, pp. 66687–66712. Curran Associates, Inc.

Desai, T., J. Shah, and G. A. Prabhu (2024). Keeping GPUs Cool: GPU Temperature Prediction Using LSTM. In *2024 IEEE 31st International Conference on High Performance Computing, Data and Analytics Workshop (HiPCW)*, pp. 111–112.

Devaney, R. (2018, March). *An introduction to chaotic dynamical systems* (2 ed.). Boca Raton: CRC press.

Dewancker, I., M. McCourt, and S. Clark (2016, December). Bayesian Optimization for Machine Learning : A Practical Guidebook. arXiv:1612.04858.

Dhariwal, P. and A. Nichol (2021). Diffusion Models Beat GANs on Image Synthesis. In *Advances in Neural Information Processing Systems*, Volume 34, pp. 8780–8794. Curran Associates, Inc.

Dheur, V. and S. B. Taieb (2023, July). A Large-Scale Study of Probabilistic Calibration in Neural Network Regression. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 7813–7836. PMLR. ISSN: 2640-3498.

Dinh, L., J. Sohl-Dickstein, and S. Bengio (2017, February). Density estimation using Real NVP. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*.

Doerr, A., C. Daniel, M. Schiegg, N.-T. Duy, S. Schaal, M. Toussaint, and T. Sebastian (2018, July). Probabilistic Recurrent State-Space Models. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1280–1289. PMLR. ISSN: 2640-3498.

Donahue, C., J. McAuley, and M. Puckette (2018, September). Adversarial Audio Synthesis. In *Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)*.

Donghao, L. and W. Xue (2023, October). ModernTCN: A Modern Pure Convolution Structure for General Time Series Analysis. In *Proceedings of the 12th International Conference on Learning Representations (ICLR 2024)*.

Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021)*.

Drouin, A., Marcotte, and N. Chapados (2022, June). TACTiS: Transformer-Attentional Copulas for Time Series. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 5447–5493. PMLR. ISSN: 2640-3498.

Du, H., S. Du, and W. Li (2023). Probabilistic time series forecasting with deep non-linear state space models. *CAAI Transactions on Intelligence Technology 8*(1), 3–13. https://onlinelibrary.wiley.com/doi/pdf/10.1049/cit2.12085.

Du, Y. and I. Mordatch (2019). Implicit Generation and Modeling with Energy Based Models. In *Advances in Neural Information Processing Systems*, Volume 32. Curran Associates, Inc.

Durbin, J. and S. J. Koopman (2012, May). *Time Series Analysis by State Space Methods* (2 ed.). Oxford University Press. ISBN: 978-0-19-964117-8.

Ekambaram, V., A. Jati, N. Nguyen, P. Sinthong, and J. Kalagnanam (2023, August). TSMixer: Lightweight MLP-Mixer Model for Multivariate Time Series Forecasting. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, New York, NY, USA, pp. 459–469. Association for Computing Machinery.

Fan, W., P. Wang, D. Wang, D. Wang, Y. Zhou, and Y. Fu (2023, June). Dish-TS: A General Paradigm for Alleviating Distribution Shift in Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 37, pp. 7522–7529. AAAI Press.

Feng, S., C. Miao, K. Xu, J. Wu, P. Wu, Y. Zhang, and P. Zhao (2024, May). Multi-Scale Attention Flow for Probabilistic Time Series Forecasting. *IEEE Transactions on Knowledge and Data Engineering 36*(5), 2056–2068.

Feng, S., C. Miao, Z. Zhang, and P. Zhao (2024, March). Latent Diffusion Transformer for Probabilistic Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 38, pp. 11979–11987. AAAI Press.

Feng, Y., H. You, Z. Zhang, R. Ji, and Y. Gao (2019, January). Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 33 of *AAAI'19*, Honolulu, Hawaii, USA, pp. 3558–3565. AAAI Press.

Gao, J., W. Hu, and Y. Chen (2023, May). Client: Cross-variable Linear Integrated Enhanced Transformer for Multivariate Long-Term Time Series Forecasting. arXiv:2305.18838.

Gasthaus, J., K. Benidis, Y. Wang, S. S. Rangapuram, D. Salinas, V. Flunkert, and T. Januschowski (2019, April). Probabilistic Forecasting with Spline Quantile Function RNNs. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pp. 1901–1910. PMLR. ISSN: 2640-3498.

Ghimire, S., A. Masoomi, and J. Dy (2021). Reliable Estimation of KL Divergence using a Discriminator in Reproducing Kernel Hilbert Space. In *Advances in Neural Information Processing Systems*, Volume 34, pp. 10221–10233. Curran Associates, Inc.

Gneiting, T. (2011, June). Making and Evaluating Point Forecasts. *Journal of the American Statistical Association 106*(494), 746–762. https://doi.org/10.1198/jasa.2011.r10138.

Gneiting, T., F. Balabdaoui, and A. E. Raftery (2007, April). Probabilistic Forecasts, Calibration and Sharpness. *Journal of the Royal Statistical Society Series B: Statistical Methodology 69*(2), 243–268.

Gneiting, T. and M. Katzfuss (2014, January). Probabilistic Forecasting. *Annual Review of Statistics and Its Application 1*(1), 125–151.

Gneiting, T. and A. E. Raftery (2007, March). Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association 102*(477), 359–378.

Gneiting, T. and R. Ranjan (2011, July). Comparing Density Forecasts Using Threshold- and Quantile-Weighted Scoring Rules. *Journal of Business & Economic Statistics 29*(3), 411–422. https://doi.org/10.1198/jbes.2010.08110.

Gneiting, T., D. Wolffram, J. Resin, K. Kraus, J. Bracher, T. Dimitriadis, V. Hagenmeyer, A. I. Jordan, S. Lerch, K. Phipps, and M. Schienle (2023, March). Model Diagnostics and Forecast Evaluation for Quantiles. *Annual Review of Statistics and Its Application 10*(1), 597–621.

Goerg, G. (2013, June). Forecastable Component Analysis. In S. Dasgupta and D. McAllester (Eds.), *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, Georgia, USA, pp. 64–72. PMLR.

Gong, Z., Y. Tang, and J. Liang (2024, October). PatchMixer: A Patch-Mixing Architecture for Long-Term Time Series Forecasting. In *Proceedings of the 6th Data Science Meets Optimisation Workshop at the Thirty-Third International Joint Conference on Artificial Intelligence*. Workshop Paper.

Good, I. J. (1952). Rational Decisions. *Journal of the Royal Statistical Society. Series B (Methodological) 14*(1), 107–114. Publisher: [Royal Statistical Society, Oxford University Press].

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. The MIT Press.

Goodfellow, I. J., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014). Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, Volume 27. Curran Associates, Inc.

Gouttes, A., K. Rasul, M. Koren, J. Stephan, and T. Naghibi (2021). Probabilistic Time Series Forecasting with Implicit Quantile Networks. In *Proceedings of the Time Series Workshop at 38th International Conference on Machine Learning*. PMLR. Workshop Paper.

Graikos, A., N. Malkin, N. Jojic, and D. Samaras (2022, December). Diffusion Models as Plug-and-Play Priors. In *Advances in Neural Information Processing Systems*, Volume 35, pp. 14715–14728. Curran Associates, Inc.

Graves, A. (2014, June). Generating Sequences With Recurrent Neural Networks. arXiv:1308.0850.

Größer, J. and O. Okhrin (2022). Copulae: An overview and recent developments. *WIREs Computational Statistics 14*(3), e1557. https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wics.1557.

Hamilton, J. D. (1994). *Time Series Analysis*. Princeton University Press. ISBN: 978-0-691-21863-2.

Han, L., X.-Y. Chen, H.-J. Ye, and D.-C. Zhan (2024, December). SOFTS: Efficient Multivariate Time Series Forecasting with Series-Core Fusion. In *Advances in Neural Information Processing Systems*, Volume 37, pp. 64145–64175. Curran Associates, Inc.

Han, L., H.-J. Ye, and D.-C. Zhan (2024, November). The Capacity and Robustness Trade-Off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting. *IEEE Transactions on Knowledge and Data Engineering 36*(11), 7129–7142. Conference Name: IEEE Transactions on Knowledge and Data Engineering.

Han, W., T. Zhu, L. Chen, H. Ning, Y. Luo, and Y. Wan (2024). MCformer: Multivariate Time Series Forecasting With Mixed-Channels Transformer. *IEEE Internet of Things Journal 11*(17), 28320–28329.

Harsha, P., R. Natarajan, and D. Subramanian (2021, July). A Prescriptive Machine-Learning Framework to the Price-Setting Newsvendor Problem. *INFORMS Journal on Optimization 3*(3), 227–253. Publisher: INFORMS.

Hewamalage, H., C. Bergmeir, and K. Bandara (2021, January). Recurrent Neural Networks for Time Series Forecasting: Current Status and Future Directions. *International Journal of Forecasting 37*(1), 388–427.

Ho, J., A. Jain, and P. Abbeel (2020). Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, Volume 33, pp. 6840–6851. Curran Associates, Inc.

Hochreiter, S. and J. Schmidhuber (1997, November). Long Short-Term Memory. *Neural Computing 9*(8), 1735–1780.

Horn, R. A. and C. R. Johnson (2012). Chapter 7: Positive Definite and Semidefinite Matrices. In *Matrix Analysis* (2 ed.)., pp. 425–516. Cambridge University Press.

Horowitz, J. L. and C. F. Manski (2006, June). Identification and estimation of statistical functionals using incomplete data. *Journal of Econometrics 132*(2), 445–459.

Hu, J., Y. Hu, W. Chen, M. Jin, S. Pan, Q. Wen, and Y. Liang (2024, December). Attractor Memory for Long-Term Time Series Forecasting: A Chaos Perspective. In *Advances in Neural Information Processing Systems*, Volume 37, pp. 20786–20818. Curran Associates, Inc.

Huang, C.-W., R. T. Q. Chen, C. Tsirigotis, and A. Courville (2020, October). Convex Potential Flows: Universal Probability Distributions with Optimal Transport and Convex Optimization. In *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020)*.

Huang, C.-W., D. Krueger, A. Lacoste, and A. Courville (2018, July). Neural Autoregressive Flows. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 2078–2087. PMLR. ISSN: 2640-3498.

Huang, Q., L. Shen, R. Zhang, J. Cheng, S. Ding, Z. Zhou, and Y. Wang (2024, March). HDMixer: Hierarchical Dependency with Extendable Patch for Multivariate Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 38, pp. 12608–12616.

Huang, Q., L. Shen, R. Zhang, S. Ding, B. Wang, Z. Zhou, and Y. Wang (2023, December). CrossGNN: Confronting Noisy Multivariate Time Series Via Cross Interaction Refinement. In *Advances in Neural Information Processing Systems*, Volume 36, pp. 46885–46902. Curran Associates, Inc.

Huang, Q., Z. Zhou, K. Yang, G. Lin, Z. Yi, and Y. Wang (2024, August). LeRet: Language-Empowered Retentive Network for Time Series Forecasting. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, Volume 5, pp. 4165–4173. ISSN: 1045-0823.

Huber, P. J. and E. M. Ronchetti (2011, September). *Robust statistics* (2 ed.).

Hyndman, R. J. and S. Fan (2010). Density Forecasting for Long-Term Peak Electricity Demand. *IEEE Transactions on Power Systems 25*(2), 1142–1153.

Hyndman, R. J. and Y. Khandakar (2008, July). Automatic Time Series Forecasting: The forecast Package for R. *Journal of Statistical Software 27*, 1–22.

Hyndman, R. J., A. B. Koehler, J. K. Ord, and R. D. Snyder (2008). *Forecasting with Exponential Smoothing: The State Space Approach*. Switzerland: Springer.

Hyndman, R. J., A. B. Koehler, R. D. Snyder, and S. Grose (2002, July). A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting 18*(3), 439–454.

Hyvärinen, A. (2005). Estimation of Non-Normalized Statistical Models by Score Matching. *Journal of Machine Learning Research 6*(24), 695–709.

Ilbert, R., A. Odonnat, V. Feofanov, A. Virmaux, G. Paolo, T. Palpanas, and I. Redko (2024, July). SAMformer: Unlocking the Potential of Transformers in Time Series Forecasting with Sharpness-Aware Minimization and Channel-Wise Attention. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 20924–20954. PMLR. ISSN: 2640-3498.

Ing, C.-K. (2007, July). Accumulated prediction errors, information criteria and optimal forecasting for autoregressive time series. *The Annals of Statistics 35*(3), 1238–1277. Publisher: Institute of Mathematical Statistics.

Izakian, H., W. Pedrycz, and I. Jamal (2015). Fuzzy clustering of time series data using dynamic time warping distance. *Engineering Applications of Artificial Intelligence 39*, 235–244.

Januschowski, T., J. Gasthaus, Y. Wang, D. Salinas, V. Flunkert, M. Bohlke-Schneider, and L. Callot (2020, January). Criteria for classifying forecasting methods. *International Journal of Forecasting 36*(1), 167–177.

Jawed, S. and L. Schmidt-Thieme (2022, December). GQFormer: A Multi-Quantile Generative Transformer for Time Series Forecasting. In *2022 IEEE International Conference on Big Data (Big Data)*, pp. 992–1001.

Jayanthi, G. and P. Jothilakshmi (2021, January). Traffic time series forecasting on highways - a contemporary survey of models, methods and techniques. *International Journal of Logistics Systems and Management 39*(1), 77–110. Publisher: Inderscience Publishers.

Jeon, J., J. Kim, H. Song, S. Cho, and N. Park (2022, December). GT-GAN: General Purpose Time Series Synthesis with Generative Adversarial Networks. In *Advances in Neural Information Processing Systems*, Volume 35, pp. 36999–37010. Curran Associates, Inc.

Jia, Y., Y. Lin, X. Hao, Y. Lin, S. Guo, and H. Wan (2023, December). WITRAN: Water-wave Information Transmission and Recurrent Acceleration Network for Long-range Time Series Forecasting. In *Advances in Neural Information Processing Systems*, Volume 36, pp. 12389–12456. Curran Associates, Inc.

Jia, Y., Y. Lin, J. Yu, S. Wang, T. Liu, and H. Wan (2024, December). PGN: The RNN's New Successor is Effective for Long-Range Time Series Forecasting. In *Advances in Neural Information Processing Systems*, Volume 37, pp. 84139–84168. Curran Associates, Inc.

Jiang, H., Z. Li, and Q. Li (2021, July). Approximation Theory of Convolutional Architectures for Time Series Modelling. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 4961–4970. PMLR. ISSN: 2640-3498.

Jiang, Y., S. Chang, and Z. Wang (2021). TransGAN: Two Pure Transformers Can Make One Strong GAN, and That Can Scale Up. In *Advances in Neural Information Processing Systems*, Volume 34, pp. 14745–14758. Curran Associates, Inc.

Jin, M., S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, and Q. Wen (2023, October). Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. In *Proceedings of the Twelfth International Conference on Learning Representations*.

Julier, S. and J. Uhlmann (2004, March). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE 92*(3), 401–422.

Kalchbrenner, N., L. Espeholt, K. Simonyan, A. v. d. Oord, A. Graves, and K. Kavukcuoglu (2017, March). Neural Machine Translation in Linear Time. arXiv:1610.10099.

Kalman, R. E. (1960, March). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering 82*(1), 35–45.

Kan, K., F.-X. Aubet, T. Januschowski, Y. Park, K. Benidis, L. Ruthotto, and J. Gasthaus (2022, May). Multivariate Quantile Function Forecaster. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pp. 10603–10621. PMLR. ISSN: 2640-3498.

Kang, B. G., D. Lee, H. Kim, D. Chung, and S. Yoon (2024, December). Introducing Spectral Attention for Long-Range Dependency in Time Series Forecasting. In *Advances in Neural Information Processing Systems*, Volume 37, pp. 136509–136544. Curran Associates, Inc.

Kim, D., J. Park, J. Lee, and H. Kim (2024, December). Are Self-Attentions Effective for Time Series Forecasting? In *Advances in Neural Information Processing Systems*, Volume 37, pp. 114180–114209. Curran Associates, Inc.

Kim, S., A. Gholami, A. Shaw, N. Lee, K. Mangalam, J. Malik, M. W. Mahoney, and K. Keutzer (2022, December). Squeezeformer: An Efficient Transformer for Automatic Speech Recognition. In *Advances in Neural Information Processing Systems*, Volume 35, pp. 9361–9373. Curran Associates, Inc.

Kim, T., J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo (2021, October). Reversible Instance Normalization for Accurate Time-Series Forecasting against Distribution Shift. In *Proceedings of the Tenth International Conference on Learning Representations*.

Kingma, D. and J. Ba (2014, December). Adam: A Method for Stochastic Optimization. In *Proceedings of the Third International Conference on Learning Representations*.

Kingma, D. P. and M. Welling (2014). Auto-Encoding Variational Bayes. In Y. Bengio and Y. LeCun (Eds.), *Proceedings of the Second International Conference on Learning Representations*, Banff, AB, Canada.

Kline, D. M. (2004). Methods for Multi-Step Time Series Forecasting Neural Networks. In *Neural Networks in Business Forecasting*, pp. 226–250. IGI Global Scientific Publishing.

Ko, J. and D. Fox (2011, January). Learning GP-BayesFilters via Gaussian process latent variable models. *Autonomous Robots 30*(1), 3–23.

Koenker, R. (2005, May). *Quantile Regression*. Cambridge University Press. ISBN: 978-1-139-44471-2.

Koenker, R. and G. Bassett (1978). Regression Quantiles. *Econometrica 46*(1), 33–50. Publisher: [Wiley, Econometric Society].

Kolassa, S. (2020, January). Why the "best" point forecast depends on the error or accuracy measure. *International Journal of Forecasting 36*(1), 208–211.

Kollovieh, M., A. F. Ansari, M. Bohlke-Schneider, J. Zschiegner, H. Wang, and Y. B. Wang (2023, December). Predict, Refine, Synthesize: Self-Guiding Diffusion Models for Probabilistic Time Series Forecasting. In *Advances in Neural Information Processing Systems*, Volume 36, pp. 28341–28364. Curran Associates, Inc.

Kong, Z., W. Ping, J. Huang, K. Zhao, and B. Catanzaro (2020, October). DiffWave: A Versatile Diffusion Model for Audio Synthesis. In *Proceedings of the Ninth International Conference on Learning Representations*.

Koochali, A., A. Dengel, and S. Ahmed (2021). If You Like It, GAN It—Probabilistic Multivariate Times Series Forecast with GAN. *Engineering Proceedings 5*(1). Publisher: Multidisciplinary Digital Publishing Institute.

Koochali, A., P. Schichtel, A. Dengel, and S. Ahmed (2022, January). Random Noise vs. State-of-the-Art Probabilistic Forecasting Methods: A Case Study on CRPS-Sum Discrimination Ability. *Applied Sciences 12*(10), 5104. Publisher: Multidisciplinary Digital Publishing Institute.

Koopman, B. O. (1931, May). Hamiltonian Systems and Transformation in Hilbert Space. *Proceedings of the National Academy of Sciences 17*(5), 315–318. Publisher: Proceedings of the National Academy of Sciences.

Kullback, S. (1997). *Information Theory and Statistics*. New York, NY, USA: Dover Publications, Inc.

Kullback, S. and R. A. Leibler (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics 22*(1), 79–86. Publisher: Institute of Mathematical Statistics.

Lai, G., W.-C. Chang, Y. Yang, and H. Liu (2018, June). Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, New York, NY, USA, pp. 95–104. Association for Computing Machinery.

Lambert, N. S., D. M. Pennock, and Y. Shoham (2008, July). Eliciting properties of probability distributions. In *Proceedings of the 9th ACM conference on Electronic commerce*, EC '08, New York, NY, USA, pp. 129–138. Association for Computing Machinery.

Lara-Benítez, P., M. Carranza-García, and J. C. Riquelme (2021, March). An Experimental Review on Deep Learning Architectures for Time Series Forecasting. *International Journal of Neural Systems 31*(03), 2130001.

LeCun, Y., S. Chopra, R. Hadsell, M. Ranzato, and F. J. Huang (2006, August). A Tutorial on Energy-Based Learning. *Predicting structured data*.

LeCun, Y., C. Cortes, and C. J. Burges (1998). The MNIST Database of Handwritten Digits. URL: http://yann.lecun.com/exdb/mnist/.

Li, L., K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar (2018). Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *Journal of Machine Learning Research 18*(185), 1–52.

Li, L., J. Zhang, J. Yan, Y. Jin, Y. Zhang, Y. Duan, and G. Tian (2021, May). Synergetic Learning of Heterogeneous Temporal Sequences for Multi-Horizon Probabilistic Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 35, pp. 8420–8428.

Li, S., X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan (2019). Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting. In *Advances in Neural Information Processing Systems*, Volume 32. Curran Associates, Inc.

Li, T., X. Wu, and J. Zhang (2020, March). Time Series Clustering Model based on DTW for Classifying Car Parks. *Algorithms 13*(3), 57. Publisher: Multidisciplinary Digital Publishing Institute.

Li, X., J. Thickstun, I. Gulrajani, P. S. Liang, and T. B. Hashimoto (2022, December). Diffusion-LM Improves Controllable Text Generation. In *Advances in Neural Information Processing Systems*, Volume 35, pp. 4328–4343. Curran Associates, Inc.

Li, Y., W. Chen, X. Hu, B. Chen, B. Sun, and M. Zhou (2023, October). Transformer-Modulated Diffusion Models for Probabilistic Multivariate Time Series Forecasting. In *Proceedings of the Ninth the Twelfth International Conference on Learning Representations*.

Li, Y., X. Lu, Y. Wang, and D. Dou (2022, December). Generative Time Series Forecasting with Diffusion, Denoise, and Disentanglement. In *Advances in Neural Information Processing Systems*, Volume 35, pp. 23009–23022. Curran Associates, Inc.

Li, Y., S. Qi, Z. Li, Z. Rao, L. Pan, and Z. Xu (2023, August). SMARTformer: Semi-Autoregressive Transformer with Efficient Integrated Window Attention for Long

Time Series Forecasting. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, Volume 3, pp. 2169–2177. ISSN: 1045-0823.

Li, Y., R. Yu, C. Shahabi, and Y. Liu (2018). Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *Proceedings of the Sixth International Conference on Learning Representations*.

Lim, B., S. Arık, N. Loeff, and T. Pfister (2021, October). Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting 37*(4), 1748–1764.

Lim, B. and S. Zohren (2021, February). Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 379*(2194), 20200209. Publisher: Royal Society.

Lin, S., W. Lin, X. Hu, W. Wu, R. Mo, and H. Zhong (2024, December). CycleNet: Enhancing Time Series Forecasting through Modeling Periodic Patterns. In *Advances in Neural Information Processing Systems*, Volume 37, pp. 106315–106345. Curran Associates, Inc.

Lin, S., W. Lin, W. Wu, H. Chen, and J. Yang (2024, July). SparseTSF: Modeling Long-term Time Series Forecasting with *1k* Parameters. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 30211–30226. PMLR. ISSN: 2640-3498.

Lin, S., W. Lin, W. Wu, F. Zhao, R. Mo, and H. Zhang (2023, August). SegRNN: Segment Recurrent Neural Network for Long-Term Time Series Forecasting. arXiv:2308.11200.

Lin, T.-Y., P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie (2017, July). Feature Pyramid Networks for Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, H., Z. Dai, D. So, and Q. V. Le (2021). Pay Attention to MLPs. In *Advances in Neural Information Processing Systems*, Volume 34, pp. 9204–9215. Curran Associates, Inc.

Liu, J., L. Yang, H. Li, and S. Hong (2024, December). Retrieval-Augmented Diffusion Models for Time Series Forecasting. In *Advances in Neural Information Processing Systems*, Volume 37, pp. 2766–2786. Curran Associates, Inc.

Liu, S., H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu, and S. Dustdar (2021, October). Pyraformer: Low-Complexity Pyramidal Attention for Long-Range Time Series Modeling and Forecasting. In *Proceedings of the Tenth International Conference on Learning Representations*.

Liu, Y., T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long (2023, October). iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *Proceedings of the Twelfth International Conference on Learning Representations*.

Liu, Y., C. Li, J. Wang, and M. Long (2023, December). Koopa: Learning Non-stationary Time Series Dynamics with Koopman Predictors. In *Advances in Neural Information Processing Systems*, Volume 36, pp. 12271–12290. Curran Associates, Inc.

Liu, Y., X. Qiao, Y. Pei, and L. Wang (2024, July). Deep Functional Factor Models: Forecasting High-Dimensional Functional Time Series via Bayesian Nonparametric Factorization. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 31709–31727. PMLR. ISSN: 2640-3498.

Liu, Y., G. Qin, X. Huang, J. Wang, and M. Long (2024, December). AutoTimes: Autoregressive Time Series Forecasters via Large Language Models. In *Advances in Neural Information Processing Systems*, Volume 37, pp. 122154–122184. Curran Associates, Inc.

Liu, Y., H. Wu, J. Wang, and M. Long (2022, December). Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting. In *Advances in Neural Information Processing Systems*, Volume 35, pp. 9881–9893. Curran Associates, Inc.

Loftsgaarden, D. O. and C. P. Quesenberry (1965, June). A Nonparametric Estimate of a Multivariate Density Function. *The Annals of Mathematical Statistics 36*(3), 1049–1051. Publisher: Institute of Mathematical Statistics.

Long, B., F. Tan, and M. Newman (2023, March). Forecasting the Monkeypox Outbreak Using ARIMA, Prophet, NeuralProphet, and LSTM Models in the United States. *Forecasting 5*(1), 127–137. Publisher: Multidisciplinary Digital Publishing Institute.

Lu, J., X. Han, Y. Sun, and S. Yang (2024, July). CATS: Enhancing Multivariate Time Series Forecasting by Constructing Auxiliary Time Series as Exogenous Variables. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 32990–33006. PMLR. ISSN: 2640-3498.

Luo, D. and X. Wang (2024, December). DeformableTST: Transformer for Time Series Forecasting without Over-reliance on Patching. In *Advances in Neural Information Processing Systems*, Volume 37, pp. 88003–88044. Curran Associates, Inc.

Luo, Y., S. Qiu, X. Tao, Y. Cai, and J. Tang (2025). Energy-Calibrated VAE with Test Time Free Lunch. In A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol (Eds.), *Computer Vision – ECCV 2024*, Cham, pp. 326–344. Springer Nature Switzerland.

Lütkepohl, H. (2005). Vector Autoregressive Moving Average Processes. In H. Lütkepohl (Ed.), *New Introduction to Multiple Time Series Analysis*, pp. 419–446. Berlin, Heidelberg: Springer.

Ma, X., X. Li, L. Fang, T. Zhao, and C. Zhang (2024, March). U-Mixer: An Unet-Mixer Architecture with Stationarity Correction for Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 38, pp. 14255–14262.

Machete, R. L. (2013, October). Contrasting probabilistic scoring rules. *Journal of Statistical Planning and Inference 143*(10), 1781–1790.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, Volume 5, pp. 281–298. University of California press.

Madsen, H. (2007, November). *Time Series Analysis*. New York: Chapman and Hall/CRC.

Mahalakshmi, G., S. Sridevi, and S. Rajaram (2016, January). A survey on forecasting of time series data. In *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)*, pp. 1–8.

Makridakis, S., E. Spiliotis, V. Assimakopoulos, Z. Chen, A. Gaba, I. Tsetlin, and R. L. Winkler (2022). The M5 uncertainty competition: Results, findings and conclusions. *International Journal of Forecasting 38*(4), 1365–1385.

Mangalam, K., Y. An, H. Girase, and J. Malik (2021). From Goals, Waypoints & Paths to Long Term Human Trajectory Forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15233–15242.

Marcotte, E., V. Zantedeschi, A. Drouin, and N. Chapados (2023, July). Regions of Reliability in the Evaluation of Multivariate Probabilistic Forecasts. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 23958–24004. PMLR. ISSN: 2640-3498.

Matheron, G. (1963, December). Principles of geostatistics. *Economic Geology 58*(8), 1246–1266.

Matheson, J. E. and R. L. Winkler (1976, June). Scoring Rules for Continuous Probability Distributions. *Management Science 22*(10), 1087–1096.

McDowell, I. C., D. Manandhar, C. M. Vockley, A. K. Schmid, T. E. Reddy, and B. E. Engelhardt (2018, January). Clustering gene expression time series data using an infinite Gaussian process mixture model. *PLOS Computational Biology 14*(1), e1005896. Publisher: Public Library of Science.

Mehtab, S. and J. Sen (2020). A Time Series Analysis-Based Stock Price Prediction Using Machine Learning and Deep Learning Models. *International Journal of Business Forecasting and Marketing Intelligence 6*(4), 272.

Mirza, M. and S. Osindero (2014, November). Conditional Generative Adversarial Nets. arXiv:1411.1784.

Montero-Manso, P. and R. J. Hyndman (2021, October). Principles and algorithms for forecasting groups of time series: Locality and globality. *International Journal of Forecasting 37*(4), 1632–1653.

Mukherjee, S., D. Shankar, A. Ghosh, N. Tathawadekar, P. Kompalli, S. Sarawagi, and K. Chaudhury (2018). AR-MDN: Associative and Recurrent Mixture Density Networks for eRetail Demand Forecasting. *Proceedings of the VLDB Endowment 11*(5).

Murphy, A. H. and H. Daan (1985). Forecast Evaluation. In *Probability, Statistics, And Decision Making In The Atmospheric Sciences*. CRC Press. Num Pages: 59.

Mészáros, A., J. F. Schumann, J. Alonso-Mora, A. Zgonnikov, and J. Kober (2024, August). ROME: Robust Multi-Modal Density Estimator. In K. Larson (Ed.), *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 4751–4759. International Joint Conferences on Artificial Intelligence Organization.

Neal, R. M. (2011). MCMC Using Hamiltonian Dynamics. In *Handbook of Markov Chain Monte Carlo*, pp. 50. Chapman and Hall/CRC. ISBN: 978-0-429-13850-8.

Nguyen, N. and B. Quanz (2021, May). Temporal Latent Auto-Encoder: A Method for Probabilistic Multivariate Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 35, pp. 9117–9125.

Nguyen, X., M. J. Wainwright, and M. I. Jordan (2010, November). Estimating Divergence Functionals and the Likelihood Ratio by Convex Risk Minimization. *IEEE Transactions on Information Theory 56*(11), 5847–5861.

Nie, Y., N. H. Nguyen, P. Sinthong, and J. Kalagnanam (2022, September). A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *Proceedings of the Eleventh International Conference on Learning Representations*.

Nikitin, A., L. Iannucci, and S. Kaski (2024, December). TSGM: A Flexible Framework for Generative Modeling of Synthetic Time Series. In *Advances in Neural Information Processing Systems*, Volume 37, pp. 129042–129061. Curran Associates, Inc.

Ning, Y., H. Kazemi, and P. Tahmasebi (2022). A comparative machine learning study for time series oil production forecasting: ARIMA, LSTM, and Prophet. *Computers & Geosciences 164*, 105126.

Nix, D. and A. Weigend (1994, June). Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, Volume 1, pp. 55–60 vol.1.

Olsen, K., R. M. H. Lindrup, and M. Mørup (2024, April). Think Global, Adapt Local: Learning Locally Adaptive K-Nearest Neighbor Kernel Density Estimators. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, pp. 4114–4122. PMLR. ISSN: 2640-3498.

Oreshkin, B. N., D. Carpov, N. Chapados, and Y. Bengio (2019, September). N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *Proceedings of the Eighth International Conference on Learning Representations*.

Osband, K. and S. Reichelstein (1985). Information-eliciting compensation schemes. *Journal of Public Economics 46*, 107–115.

Papamakarios, G., E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan (2021). Normalizing Flows for Probabilistic Modeling and Inference. *Journal of Machine Learning Research 22*(57), 1–64.

Papamakarios, G., T. Pavlakou, and I. Murray (2017). Masked Autoregressive Flow for Density Estimation. In *Advances in Neural Information Processing Systems*, Volume 30. Curran Associates, Inc.

Paparrizos, J. and L. Gravano (2015, May). k-Shape: Efficient and Accurate Clustering of Time Series. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, New York, NY, USA, pp. 1855–1870. Association for Computing Machinery.

Paparrizos, J., F. Yang, and H. Li (2024, December). Bridging the Gap: A Decade Review of Time-Series Clustering Methods. arXiv:2412.20582.

Park, Y., D. Maddix, F.-X. Aubet, K. Kan, J. Gasthaus, and Y. Wang (2022, May). Learning Quantile Functions without Quantile Crossing for Distribution-free Time Series Forecasting. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pp. 8127–8150. PMLR. ISSN: 2640-3498.

Parzen, E. (1962). On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics 33*(3), 1065–1076. Publisher: Institute of Mathematical Statistics.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Volume 32. Curran Associates, Inc.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research 12*, 2825–2830.

Piao, X., Z. Chen, T. Murayama, Y. Matsubara, and Y. Sakurai (2024, August). Fredformer: Frequency Debiased Transformer for Time Series Forecasting. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, New York, NY, USA, pp. 2400–2410. Association for Computing Machinery.

Pinson, P. and J. Tastu (2013). Discrimination ability of the Energy score. Technical Report 15, Technical University of Denmark, Kgs. Lyngby.

Pourahmadi, M. (2011, August). Covariance Estimation: The GLM and Regularization Perspectives. *Statistical Science 26*(3), 369–387. Publisher: Institute of Mathematical Statistics.

Qin, Y., D. Song, H. Chen, W. Cheng, G. Jiang, and G. W. Cottrell (2017). A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 2627–2633.

Raeesi, M., M. S. Mesgari, and P. Mahmoudi (2014, October). Traffic time series forecasting by feedforward neural network: a case study based on traffic data of monroe. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XL-2/W3*, 219–223.

Rangapuram, S. S., M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski (2018). Deep State Space Models for Time Series Forecasting. In *Advances in Neural Information Processing Systems*, Volume 31. Curran Associates, Inc.

Rasul, K., A. Bennett, P. Vicente, U. Gupta, H. Ghonia, A. Schneider, and Y. Nevmyvaka (2023, October). VQ-TR: Vector Quantized Attention for Time Series Forecasting. In *Proceedings of the Twelfth International Conference on Learning Representations*.

Rasul, K., Y.-J. Park, M. N. Ramström, and K.-M. Kim (2022, May). VQ-AR: Vector Quantized Autoregressive Probabilistic Time Series Forecasting. arXiv:2205.15894.

Rasul, K., C. Seward, I. Schuster, and R. Vollgraf (2021, July). Autoregressive Denoising Diffusion Models for Multivariate Probabilistic Time Series Forecasting. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 8857–8868. PMLR. ISSN: 2640-3498.

Rasul, K., A.-S. Sheikh, I. Schuster, U. M. Bergmann, and R. Vollgraf (2020, October). Multivariate Probabilistic Time Series Forecasting via Conditioned Normalizing Flows. In *Proceedings of the Ninth International Conference on Learning Representations*.

Ray, E. L., N. Wattanachit, J. Niemi, A. H. Kanji, K. House, E. Y. Cramer, J. Bracher, A. Zheng, T. K. Yamana, X. Xiong, and others (2020). Ensemble forecasts of coronavirus disease 2019 (COVID-19) in the US. *MedRXiv*, 2020–08. Publisher: Cold Spring Harbor Laboratory Press.

Rezende, D. J., S. Mohamed, and D. Wierstra (2014, June). Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 1278–1286. PMLR. ISSN: 1938-7228.

Ronneberger, O., P. Fischer, and T. Brox (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F.

Frangi (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Cham, pp. 234–241. Springer International Publishing.

Rosenblatt, M. (1956). Estimation of a probability density-function and mode. *The Annals of Mathematical Statistics 27*, 832. Place: Baltimore, Md. Publisher: Waverly Press.

Roweis, S. and Z. Ghahramani (1999, February). A Unifying Review of Linear Gaussian Models. *Neural Computation 11*(2), 305–345.

Sakoe, H. and S. Chiba (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing 26*(1), 43–49.

Salinas, D., M. Bohlke-Schneider, L. Callot, R. Medico, and J. Gasthaus (2019). High-dimensional multivariate forecasting with low-rank Gaussian Copula Processes. In *Advances in Neural Information Processing Systems*, Volume 32. Curran Associates, Inc.

Salinas, D., V. Flunkert, J. Gasthaus, and T. Januschowski (2020, July). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting 36*(3), 1181–1191.

Scheuerer, M. and T. M. Hamill (2015, April). Variogram-Based Proper Scoring Rules for Probabilistic Forecasts of Multivariate Quantities. *Monthly Weather Review 143*(4), 1321–1334. Publisher: American Meteorological Society.

Scott, D. W. (2015, March). *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons. ISBN: 978-0-429-13850-8.

Seeger, M. W., D. Salinas, and V. Flunkert (2016). Bayesian Intermittent Demand Forecasting for Large Inventories. In *Advances in Neural Information Processing Systems*, Volume 29. Curran Associates, Inc.

Semenoglou, A.-A., E. Spiliotis, S. Makridakis, and V. Assimakopoulos (2021, July). Investigating the accuracy of cross-learning time series forecasting methods. *International Journal of Forecasting 37*(3), 1072–1084.

Sen, R., H.-F. Yu, and I. S. Dhillon (2019). Think Globally, Act Locally: A Deep Neural Network Approach to High-Dimensional Time Series Forecasting. In *Advances in Neural Information Processing Systems*, Volume 32. Curran Associates, Inc.

Shahroudi, N., M. Lepson, and M. Kull (2024, July). Evaluation of Trajectory Distribution Predictions with Energy Score. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 44322–44341. PMLR. ISSN: 2640-3498.

Shang, Z., L. Chen, B. Wu, and D. Cui (2024, December). Ada-MSHyper: Adaptive Multi-Scale Hypergraph Transformer for Time Series Forecasting. In *Advances in Neural Information Processing Systems*, Volume 37, pp. 33310–33337. Curran Associates, Inc.

Shao, Z., F. Wang, Y. Xu, W. Wei, C. Yu, Z. Zhang, D. Yao, T. Sun, G. Jin, X. Cao, G. Cong, C. S. Jensen, and X. Cheng (2025, January). Exploring Progress in Multivariate Time Series Forecasting: Comprehensive Benchmarking and Heterogeneity Analysis. *IEEE Transactions on Knowledge and Data Engineering 37*(1), 291–305.

Shen, L., W. Chen, and J. Kwok (2023, October). Multi-Resolution Diffusion Models for Time Series Forecasting. In *Proceedings of the Twelfth International Conference on Learning Representations*.

Shen, L. and J. Kwok (2023, July). Non-autoregressive Conditional Diffusion Models for Time Series Prediction. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 31016–31029. PMLR. ISSN: 2640-3498.

Shen, L., Y. Wei, Y. Wang, and H. Qiu (2024, April). Take an Irregular Route: Enhance the Decoder of Time-Series Forecasting Transformer. *IEEE Internet of Things Journal 11*(8), 14344–14356.

Sklar, M. (1959). Fonctions de répartition à n dimensions et leurs marges. In *Annales de l'ISUP*, Volume 8, pp. 229–231.

Sohl-Dickstein, J., E. Weiss, N. Maheswaranathan, and S. Ganguli (2015, June). Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 2256–2265. PMLR. ISSN: 1938-7228.

Sohn, K., H. Lee, and X. Yan (2015). Learning Structured Output Representation using Deep Conditional Generative Models. In *Advances in Neural Information Processing Systems*, Volume 28. Curran Associates, Inc.

Song, Y., C. Durkan, I. Murray, and S. Ermon (2021). Maximum Likelihood Training of Score-Based Diffusion Models. In *Advances in Neural Information Processing Systems*, Volume 34, pp. 1415–1428. Curran Associates, Inc.

Sprangers, O., S. Schelter, and M. de Rijke (2023, January). Parameter-efficient deep probabilistic forecasting. *International Journal of Forecasting 39*(1), 332–345.

Sun, F.-K. and D. S. Boning (2022, May). FreDo: Frequency Domain-based Long-Term Time Series Forecasting. arXiv:2205.12301.

Sun, Y., L. Ma, Y. Liu, S. Wang, J. Zhang, Y. Zheng, H. Yun, L. Lei, Y. Kang, and L. Ye (2022, July). Memory Augmented State Space Model for Time Series Forecasting. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, Volume 4, pp. 3451–3457. ISSN: 1045-0823.

Sutskever, I., O. Vinyals, and Q. V. Le (2014). Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems*, Volume 27. Curran Associates, Inc.

Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich (2015, June). Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9. ISSN: 1063-6919.

Tabak, E. G. and C. V. Turner (2013). A Family of Nonparametric Density Estimation Algorithms. *Communications on Pure and Applied Mathematics 66*(2), 145–164. https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.21423.

Taieb, S. B. and A. F. Atiya (2016, January). A Bias and Variance Analysis for Multistep-Ahead Time Series Forecasting. *IEEE Transactions on Neural Networks and Learning Systems 27*(1), 62–76.

Taieb, S. B., G. Bontempi, A. F. Atiya, and A. Sorjamaa (2012, June). A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert Systems with Applications 39*(8), 7067–7083.

Taieb, S. B. and R. J. Hyndman (2012). Recursive and direct multi-step forecasting: the best of both worlds.

Tang, B. and D. S. Matteson (2021). Probabilistic Transformer For Time Series Analysis. In *Advances in Neural Information Processing Systems*, Volume 34, pp. 23592–23608. Curran Associates, Inc.

Tapak, L., O. Hamidi, M. Fathian, and M. Karami (2019, June). Comparative evaluation of time series models for predicting influenza outbreaks: application of influenza-like illness data from sentinel sites of healthcare centers in Iran. *BMC Research Notes 12*(1), 353.

Tarima, S. and Z. Zenkova (2020, October). Use of Uncertain Additional Information in Newsvendor Models. In *2020 5th International Conference on Logistics Operations Management (GOL)*, pp. 1–6.

Tashiro, Y., J. Song, Y. Song, and S. Ermon (2021). CSDI: Conditional Score-based Diffusion Models for Probabilistic Time Series Imputation. In *Advances in Neural Information Processing Systems*, Volume 34, pp. 24804–24816. Curran Associates, Inc.

Tavenard, R., J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Rußwurm, K. Kolar, and E. Woods (2020). Tslearn, A Machine Learning Toolkit for Time Series Data. *Journal of Machine Learning Research 21*(118), 1–6.

Taylor, S. J. and B. Letham (2018, January). Forecasting at Scale. *The American Statistician 72*(1), 37–45.

Theis, L., A. van den Oord, and M. Bethge (2016, May). A note on the evaluation of generative models. In *Proceedings of the Fourth International Conference on Learning Representations*, pp. 1–10.

Thomson, W. (1979). Eliciting production possibilities from a well-informed manager. *Journal of Economic Theory 20*(3), 360–380.

Tiao, G. C. and R. S. Tsay (1994). Some advances in non-linear and adaptive modelling in time-series. *Journal of Forecasting 13*(2), 109–131. https://onlinelibrary.wiley.com/doi/pdf/10.1002/for.3980130206.

Tipping, M. E. and C. M. Bishop (1999). Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology 61*(3), 611–622. https://academic.oup.com/jrsssb/article-pdf/61/3/611/49589634/jrsssb_61_3_611.pdf.

Tolstikhin, I. O., N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, M. Lucic, and A. Dosovitskiy (2021). MLP-Mixer: An all-MLP Architecture for Vision. In *Advances in Neural Information Processing Systems*, Volume 34, pp. 24261–24272. Curran Associates, Inc.

Tong, J., L. Xie, and K. Zhang (2022). Probabilistic Decomposition Transformer for Time Series Forecasting. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pp. 478–486. https://epubs.siam.org/doi/pdf/10.1137/1.9781611977653.ch54.

Touvron, H., P. Bojanowski, M. Caron, M. Cord, A. El-Nouby, E. Grave, G. Izacard, A. Joulin, G. Synnaeve, J. Verbeek, and H. Jegou (2023, April). ResMLP: Feedforward Networks for Image Classification With Data-Efficient Training. *IEEE transactions on pattern analysis and machine intelligence 45*(4), 5314–5321.

Tyralis, H. and G. Papacharalampous (2024, March). A review of predictive uncertainty estimation with machine learning. *Artificial Intelligence Review 57*(4), 94.

Uria, B., I. Murray, and H. Larochelle (2013). RNADE: The real-valued neural autoregressive density-estimator. In *Advances in Neural Information Processing Systems*, Volume 26. Curran Associates, Inc.

Vaghefi, A., M. A. Jafari, E. Bisse, Y. Lu, and J. Brouwer (2014). Modeling and forecasting of cooling and electricity load demand. *Applied Energy 136*, 186–196.

Vahdat, A. and J. Kautz (2020). NVAE: A Deep Hierarchical Variational Autoencoder. In *Advances in Neural Information Processing Systems*, Volume 33, pp. 19667–19679. Curran Associates, Inc.

Van den Oord, A., S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu (2016, September). WaveNet: A Generative Model for Raw Audio. arXiv:1609.03499.

Van den Oord, A., O. Vinyals, and K. Kavukcuoglu (2017). Neural Discrete Representation Learning. In *Advances in Neural Information Processing Systems*, Volume 30. Curran Associates, Inc.

Van Rossum, G. and F. L. Drake Jr (1995). *Python tutorial*, Volume 620. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, and I. Polosukhin (2017). Attention is All you Need. In *Advances in Neural Information Processing Systems*, Volume 30. Curran Associates, Inc.

Wan, C., J. Lin, J. Wang, Y. Song, and Z. Y. Dong (2017). Direct Quantile Regression for Nonparametric Probabilistic Forecasting of Wind Power Generation. *IEEE Transactions on Power Systems 32*(4), 2767–2778.

Wang, H., X. Jiang, and M. Ebrahimi (2024). Scheduled Sampling for Recursive Multi-Step GPU Temperature Forecasting. In *2024 International Conference on Machine Learning and Applications (ICMLA)*, pp. 991–996.

Wang, H., J. Peng, F. Huang, J. Wang, J. Chen, and Y. Xiao (2022, September). MICN: Multi-scale Local and Global Context Modeling for Long-term Series Forecasting. In *Proceedings of the Eleventh International Conference on Learning Representations*.

Wang, J., J. Xu, and X. Wang (2018, January). Combination of Hyperband and Bayesian Optimization for Hyperparameter Optimization in Deep Learning. arXiv:1801.01596.

Wang, Q., S. Li, and R. Li (2018, October). Forecasting energy demand in China and India: Using single-linear, hybrid-linear, and non-linear time series forecast techniques. *Energy 161*, 821–831.

Wang, S., Z. Deng, F.-l. Chung, and W. Hu (2013, April). From Gaussian kernel density estimation to kernel methods. *International Journal of Machine Learning and Cybernetics 4*(2), 119–137.

Wang, S., H. Wu, X. Shi, T. Hu, H. Luo, L. Ma, J. Y. Zhang, and J. Zhou (2023, October). TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting. In *Proceedings of the Twelfth International Conference on Learning Representations*.

Wang, X., L. Tong, and Q. Zhao (2024). Generative Probabilistic Time Series Forecasting and Applications in Grid Operations. In *2024 58th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–6.

Wang, X., T. Zhou, Q. Wen, J. Gao, B. Ding, and R. Jin (2023, October). CARD: Channel Aligned Robust Blend Transformer for Time Series Forecasting. In *Proceedings of the Twelfth International Conference on Learning Representations*.

Wang, Y. (2020). EasyTorch: Simple and powerful pytorch framework. URL: https://github.com/cnstark/easytorch.

Weiss, A. A. (1991, April). Multi-step estimation and forecasting in dynamic models. *Journal of Econometrics 48*(1), 135–149.

Welling, M. and Y. W. Teh (2011, June). Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, ICML'11, Madison, WI, USA, pp. 681–688. PMLR.

Wen, Q., T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun (2023, August). Transformers in Time Series: A Survey. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, Volume 6, pp. 6778–6786. ISSN: 1045-0823.

Wen, R. and K. Torkkola (2019, July). Deep Generative Quantile-Copula Models for Probabilistic Forecasting. In *Proceedings of the Time Series Workshop at 36th International Conference on Machine Learning*, Long Beach, California. PMLR. Workshop Paper.

Wen, R., K. Torkkola, B. Narayanaswamy, and D. Madeka (2018). A Multi-Horizon Quantile Recurrent Forecaster. In *Time Series Workshop at 31st Conference on Neural Information Processing Systems*, Long Beach, California. Curran Associates, Inc. Workshop Paper.

Wilk, M. B. and R. Gnanadesikan (1968). Probability Plotting Methods for the Analysis of Data. *Biometrika 55*(1), 1–17. Publisher: [Oxford University Press, Biometrika Trust].

Williams, R. J. and D. Zipser (1989, June). A learning algorithm for continually running fully recurrent neural networks. *Neural Comput. 1*(2), 270–280.

Wilson, A. G. and Z. Ghahramani (2010). Copula Processes. In *Advances in Neural Information Processing Systems*, Volume 23. Curran Associates, Inc.

Winkler, R. L. (1972, March). A Decision-Theoretic Approach to Interval Estimation. *Journal of the American Statistical Association 67*(337), 187–191. https://www.tandfonline.com/doi/pdf/10.1080/01621459.1972.10481224.

Winkler, R. L. and A. H. Murphy (1968, October). "Good" Probability Assessors. *Journal of Applied Meteorology and Climatology 7*(5), 751–758. Publisher: American Meteorological Society Section: Journal of Applied Meteorology and Climatology.

Winkler, R. L., J. Muñoz, J. L. Cervera, J. M. Bernardo, G. Blattenberger, J. B. Kadane, D. V. Lindley, A. H. Murphy, R. M. Oliver, and D. Ríos-Insua (1996, June). Scoring rules and the evaluation of probabilities. *Test 5*(1), 1–60.

Wu, H., T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long (2022, September). TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *Proceedings of the Eleventh International Conference on Learning Representations*.

Wu, H., J. Xu, J. Wang, and M. Long (2021). Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan (Eds.), *Advances in Neural Information Processing Systems*, Volume 34, pp. 22419–22430. Curran Associates, Inc.

Wu, S., X. Xiao, Q. Ding, P. Zhao, Y. Wei, and J. Huang (2020). Adversarial Sparse Transformer for Time Series Forecasting. In *Advances in Neural Information Processing Systems*, Volume 33, pp. 17105–17115. Curran Associates, Inc.

Wu, Y., J. M. Hernandez-Lobato, and G. Zoubin (2013, May). Dynamic Covariance Models for Multivariate Financial Time Series. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 558–566. PMLR. ISSN: 1938-7228.

Wu, Y., Y. Qin, and M. Zhu (2020). High-dimensional covariance matrix estimation using a low-rank and diagonal decomposition. *Canadian Journal of Statistics 48*(2), 308–337. https://onlinelibrary.wiley.com/doi/pdf/10.1002/cjs.11532.

Wu, Y., Y. Yang, H. Nishiura, and M. Saitoh (2018, June). Deep Learning for Epidemiological Predictions. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, Ann Arbor MI USA, pp. 1085–1088. ACM.

Xie, Y., C. Li, M. Li, F. Liu, and M. Taukenova (2023, January). An overview of deterministic and probabilistic forecasting methods of wind energy. *iScience 26*(1).

Xu, Z., A. Zeng, and Q. Xu (2023, October). FITS: Modeling Time Series with \$10k\$ Parameters. In *Proceedings of the Twelfth International Conference on Learning Representations*.

Yan, T., H. Gong, H. Yongping, Y. Zhan, and Y. Xia (2024, July). Probabilistic Time Series Modeling with Decomposable Denoising Diffusion Model. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 55759–55777. PMLR. ISSN: 2640-3498.

Yang, Y., M. Jin, H. Wen, C. Zhang, Y. Liang, L. Ma, Y. Wang, C. Liu, B. Yang, Z. Xu, J. Bian, S. Pan, and Q. Wen (2024, June). A Survey on Diffusion Models for Time Series and Spatio-Temporal Data. arXiv:2404.18886.

Yeğin, M. N. and M. F. Amasyalı (2024, December). Generative diffusion models: A survey of current theoretical developments. *Neurocomputing 608*, 128373.

Yi, K., Q. Zhang, W. Fan, S. Wang, P. Wang, H. He, N. An, D. Lian, L. Cao, and Z. Niu (2023, December). Frequency-domain MLPs are More Effective Learners in Time Series Forecasting. In *Advances in Neural Information Processing Systems*, Volume 36, pp. 76656–76679. Curran Associates, Inc.

Yoon, J., D. Jarrett, and M. van der Schaar (2019). Time-series Generative Adversarial Networks. In *Advances in Neural Information Processing Systems*, Volume 32. Curran Associates, Inc.

Yuan, Y. and K. M. Kitani (2019, September). Diverse Trajectory Forecasting with Determinantal Point Processes. In *Proceedings of the Eighth International Conference on Learning Representations*.

Zeng, A., M. Chen, L. Zhang, and Q. Xu (2023, June). Are Transformers Effective for Time Series Forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 37, pp. 11121–11128.

Zhang, G., B. Eddy Patuwo, and M. Y. Hu (1998, March). Forecasting with artificial neural networks:: The state of the art. *International Journal of Forecasting 14*(1), 35–62.

Zhang, J., X. Wen, Z. Zhang, S. Zheng, J. Li, and J. Bian (2024). ProbTS: Benchmarking Point and Distributional Forecasting across Diverse Prediction Horizons. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), *Advances in Neural Information Processing Systems*, Volume 37, pp. 48045–48082. Curran Associates, Inc.

Zhang, T., Y. Zhang, W. Cao, J. Bian, X. Yi, S. Zheng, and J. Li (2022, July). Less Is More: Fast Multivariate Time Series Forecasting with Light Sampling-oriented MLP Structures. arXiv:2207.01186.

Zhang, Y., M. Li, S. Wang, S. Dai, L. Luo, E. Zhu, H. Xu, X. Zhu, C. Yao, and H. Zhou (2021, March). Gaussian Mixture Model Clustering with Incomplete Data. *ACM Trans. Multimedia Comput. Commun. Appl. 17*(1s). Place: New York, NY, USA Publisher: Association for Computing Machinery.

Zhang, Y. and J. Yan (2022, September). Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting. In *Proceedings of the Eleventh International Conference on Learning Representations*.

Zhao, D., S. Samsi, J. McDonald, B. Li, D. Bestor, M. Jones, D. Tiwari, and V. Gadepally (2023, October). Sustainable Supercomputing for AI: GPU Power Capping at HPC Scale. In *Proceedings of the 2023 ACM Symposium on Cloud Computing*, SoCC '23, pp. 588–596. ACM.

Zhao, P. and L. Lai (2021, July). On the Convergence Rates of KNN Density Estimation. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 2840–2845.

Zhao, P. and L. Lai (2022, December). Analysis of KNN Density Estimation. *IEEE Transactions on Information Theory 68*(12), 7971–7995.

Zheng, V. Z. and L. Sun (2024, December). Multivariate Probabilistic Time Series Forecasting with Correlated Errors. In *Advances in Neural Information Processing Systems*, Volume 37, pp. 54288–54329. Curran Associates, Inc.

Zhou, C., C. Che, P. Wang, and Q. Zhang (2024, August). SCAT: A Time Series Forecasting with Spectral Central Alternating Transformers. In *Proceedings of the*

Thirty-Third International Joint Conference on Artificial Intelligence, Volume 6, pp. 5626–5634. ISSN: 1045-0823.

Zhou, H., S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang (2021, May). Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 35, pp. 11106–11115.

Zhou, T., Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin (2022, July). FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato (Eds.), *Proceedings of the 39th International Conference on Machine Learning*, Volume 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286. PMLR.

Zhou, Y., Z. Chu, Y. Ruan, G. Jin, Y. Huang, and S. Li (2023, August). pTSE: A Multi-model Ensemble Method for Probabilistic Time Series Forecasting. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, Volume 4, pp. 4684–4692. ISSN: 1045-0823.

Zhou, Z., G. Lyu, Y. Huang, Z. Wang, Z. Jia, and Z. Yang (2024, August). SDformer: Transformer with Spectral Filter and Dynamic Attention for Multivariate Time Series Long-term Forecasting. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, Volume 6, pp. 5689–5697. ISSN: 1045-0823.

Ziel, F. and K. Berk (2019, October). Multivariate Forecasting Evaluation: On Sensitive and Strictly Proper Scoring Rules. arXiv:1910.07325.

A. Dataset Details

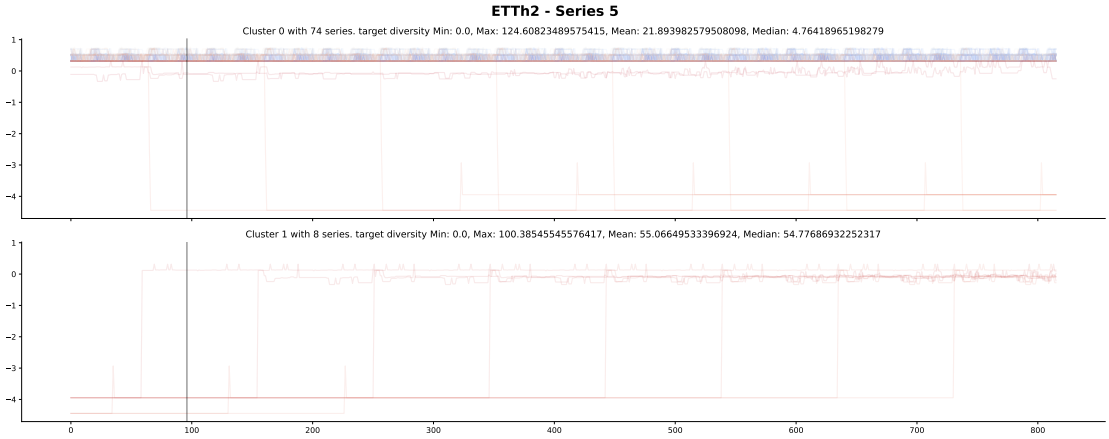


Figure A.1.: Clustering visualization for the ETTh2 dataset. Compared to ETTh1, this dataset exhibits less clearly separated forecast trajectories, with less interpretable multi-modal behavior in the forecasting horizon.

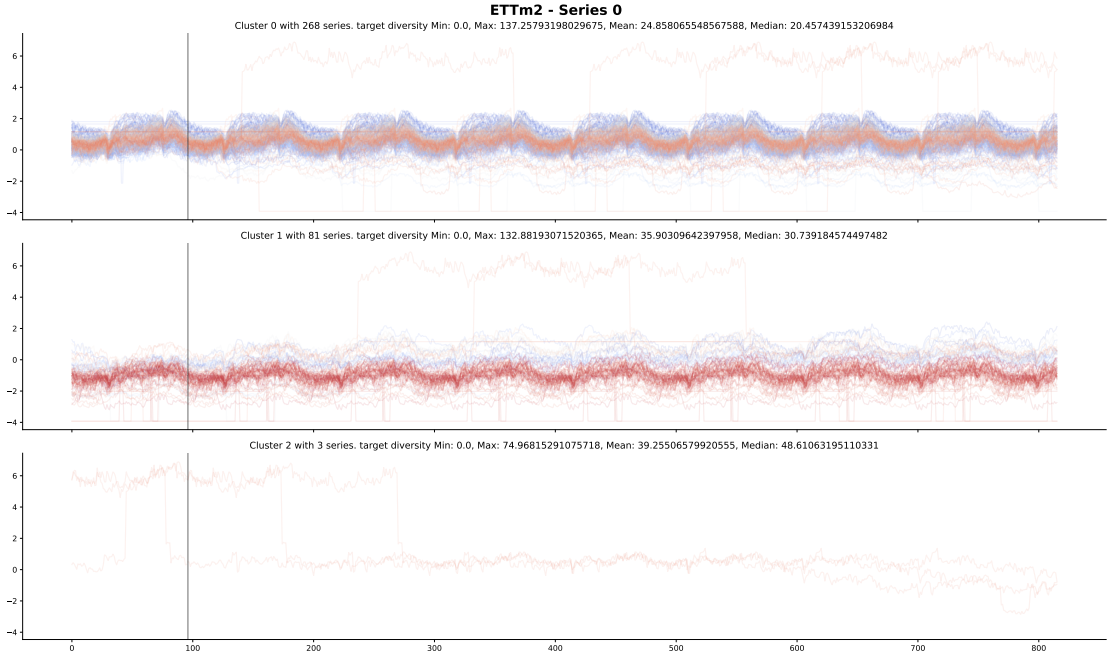


Figure A.2.: Clustering visualization for the ETTm2 dataset. While the diversity score is high, most of it is driven by isolated outliers, such as the single divergent trajectory within cluster 0, limiting the interpretability of multi-modal structures.

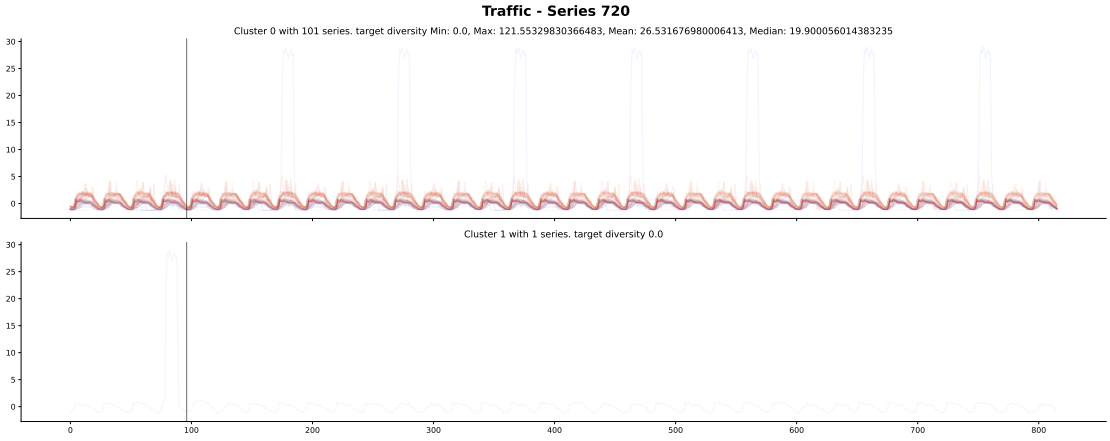


Figure A.3.: Clustering visualization for the Traffic dataset. Although it ranks high in maximum diversity, much of this stems from rare, extreme fluctuations rather than meaningful, sustained multi-modal patterns.

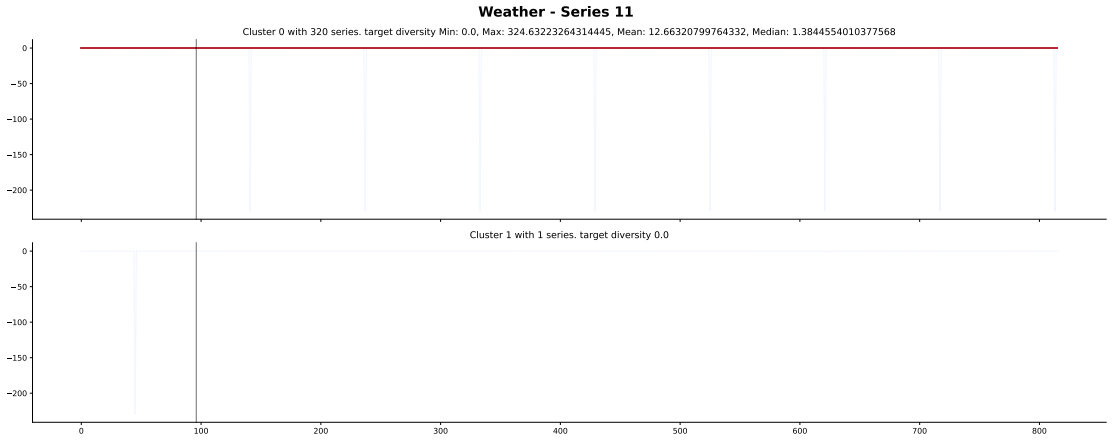


Figure A.4.: Clustering visualization for the Weather dataset. Despite having the highest diversity score, its clustering is highly imbalanced, with nearly all time series grouped into a single cluster, reducing its utility for studying *multi-world* dynamics.

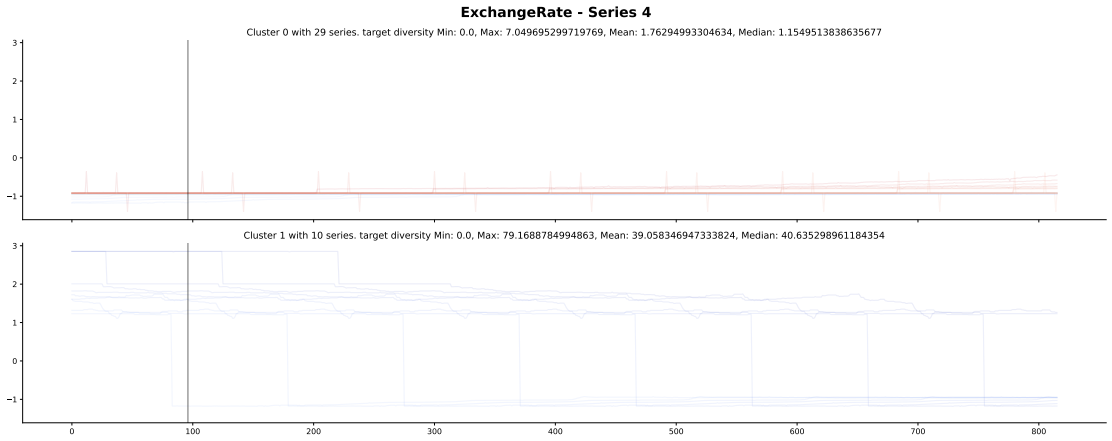


Figure A.5.: Clustering visualization for the ExchangeRate dataset. The high diversity observed in Table 5.1 is primarily due to extreme, isolated anomalies rather than robust, interpretable cluster structures.

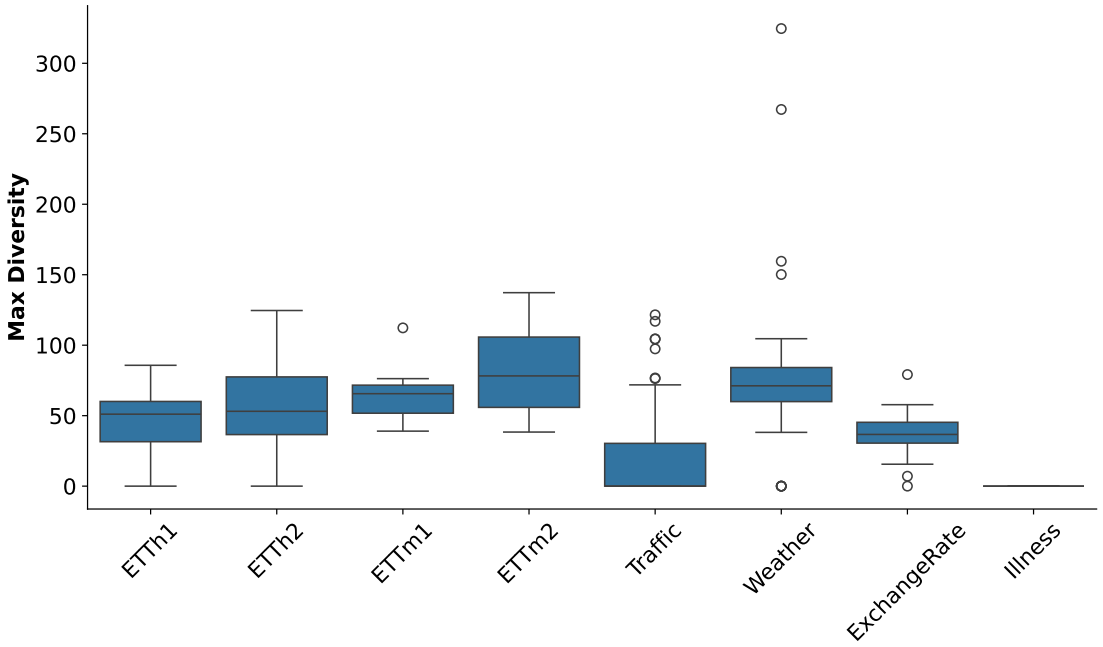


Figure A.6.: Box plot showing the distribution of maximum diversity values across individual time series in each dataset. Weather and Traffic exhibit high outliers.

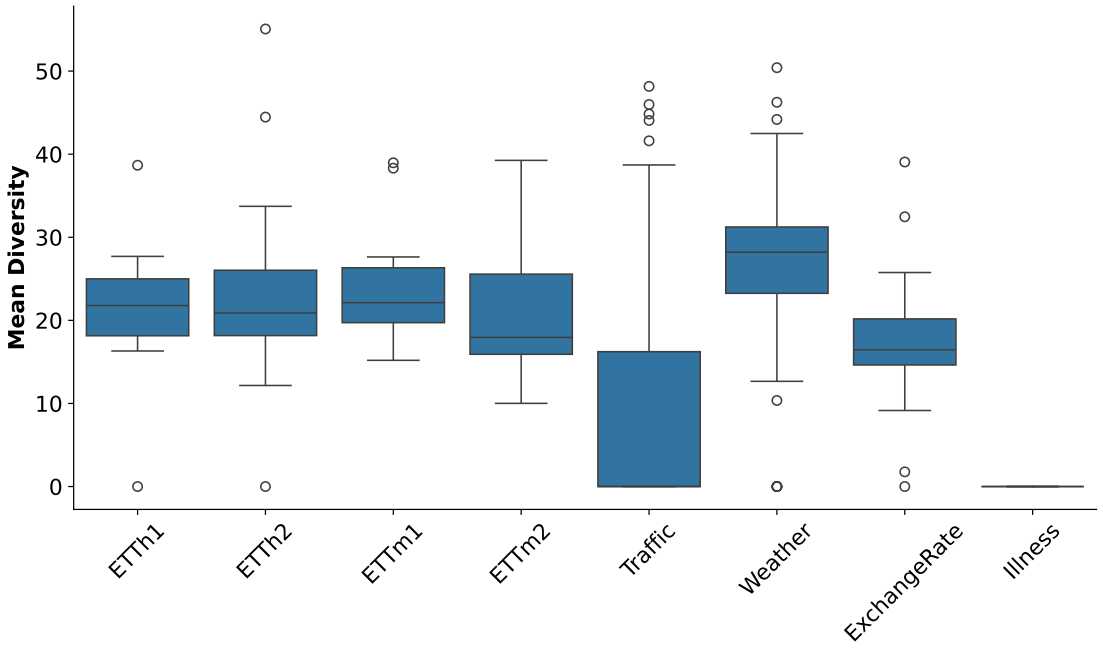


Figure A.7.: Box plot showing the distribution of mean diversity values across individual time series in each dataset.

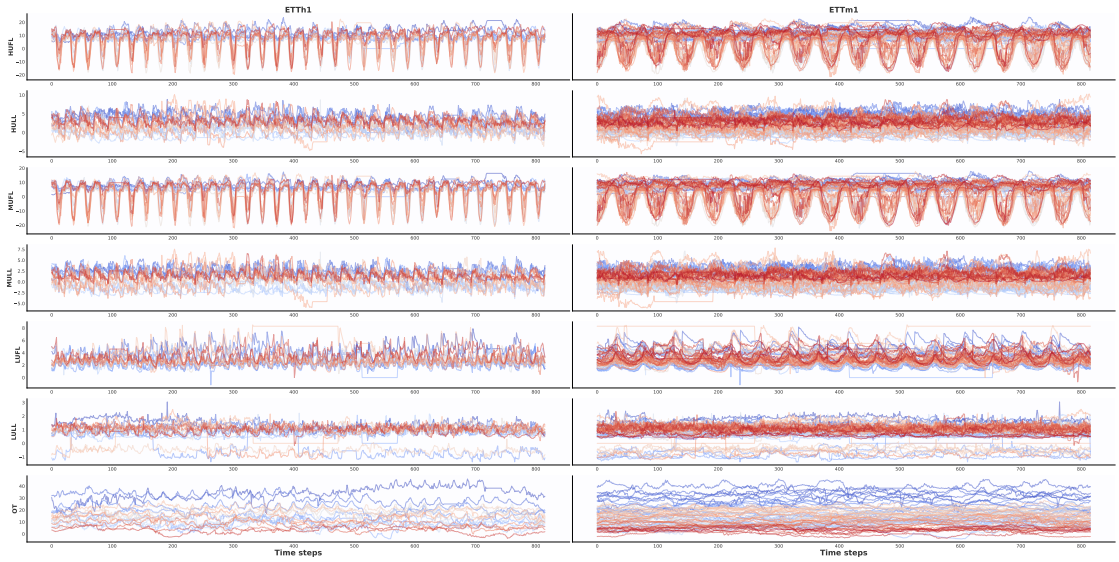


Figure A.8.: Illustration of all individual series of the ETTh1 and ETTm1 datasets, where each time series is displayed in non-overlapping windows of length $816 = 96 + 720$. A color gradient from blue (earlier segments) to red (later segments) illustrates the temporal progression across the dataset.

B. Hyperparameter Configurations

Table B.1.: Model-specific hyperparameters and their respective value ranges or distributions used during hyperparameter optimization. This table includes all tunable parameters considered for DeepAR, DLinear and PatchTST.

Model	Parameter	Range / Values
DeepAR	embedding_size	[8, 16, 32, 64, 128]
	hidden_size	[8, 16, 32, 64, 128, 256]
	num_layers	[2, 3, 4, 8]
	id_feat_size	[8, 16, 32, 64, 128]
	use_ts_id	[True, False]
DLinear	individual, prob_individual	[True, False]
PatchTST	encoder_layers	[2, 3, 4, 5, 7, 10]
	n_heads	[2, 4, 8]
	dim_model, dim_ff	[8, 16, 32, 64, 128]
	dropout, fc_dropout, head_dropout, attn_dropout	Uniform(0.01, 0.4)
	patch_len, stride	[2, 4, 8, 16, 32, 64, 128]
	padding_patch	['None', 'end']
	individual, revin, affin, subtract_last, decomposition	[0, 1]
	kernel_size	[3, 7, 13, 25, 49]
	positional_encoding	[normal, zeros, zero, uniform, lin1d, exp1d, lin2d, exp2d, sincos, None]
	norm	['LayerNorm', 'BatchNorm']
	activation	['gelu', 'relu']
	pre_norm	[True, False]

Table B.2.: Hyperparameter search space for each probabilistic forecasting head. This includes all tunable parameters and their respective value ranges or distributions used during optimization for univariate, multivariate, quantile and i_quantile heads.

Prob. Head	Parameter	Range / Values
Univariate	`distribution_type`	`[gaussian, laplace, student_t]`
Multivariate	`rank`	`[7, 36, 72, 110, 180, 360, 480]`
quantile	`quantiles`	`[0.005, 0.025, 0.165, 0.25, 0.5, 0.75, 0.835, 0.975, 0.995]`
i_quantile	`num_layers`	`[2, 3, 4, 5, 7, 10]`
	`quantile_embed_dim`, `cos_embedding_dim`	`[8, 16, 32, 64, 128]`
	`decoding`	`[concat, hadamard]`

C. Additional Experimental Results

C.1. Simple Multi-World Example

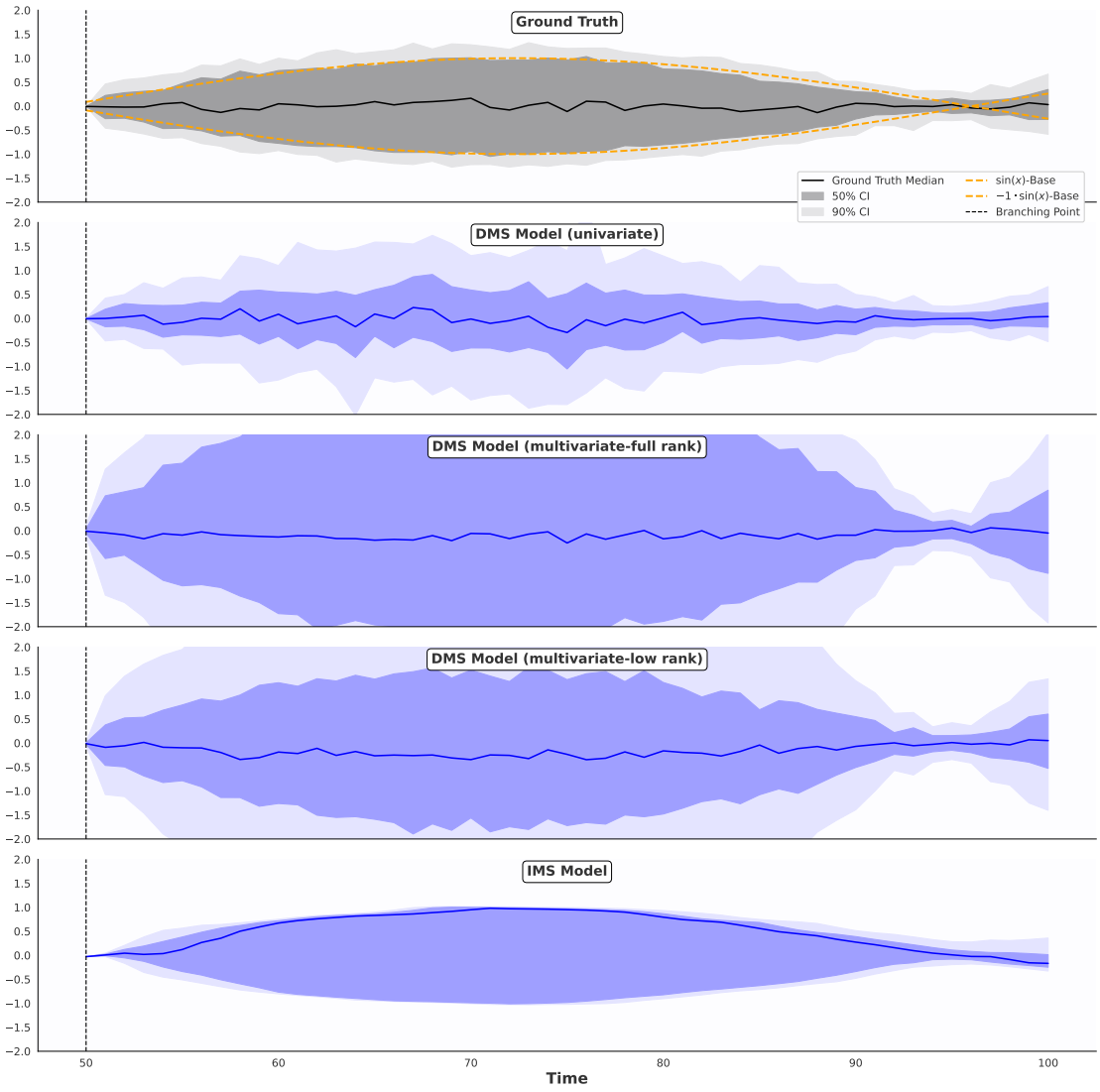


Figure C.1.: Visualization of central intervals. The ground truth intervals (top) are shown in black, while the forecasted intervals are shown in blue. Darker shades represent the 50% central intervals, and lighter, more transparent shades indicate the 90% intervals. Median lines are displayed in black for the ground truth and dark blue for the forecast. The visualization reveals that the central predictive intervals fail to capture the underlying multi-modal structure effectively.

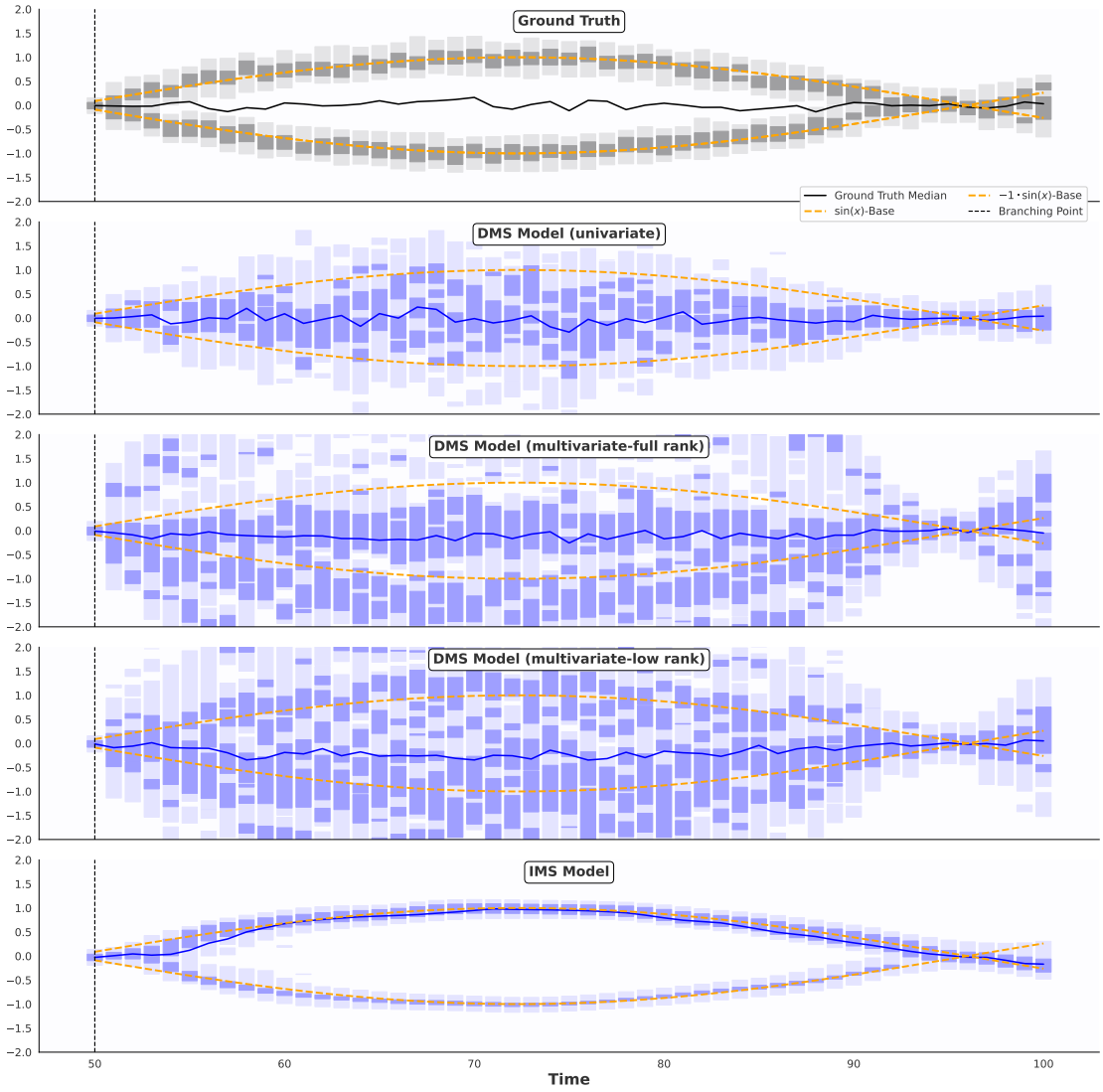


Figure C.2.: Visualization of KDE-approximated intervals. This better reflects the multi-modal nature of the data.

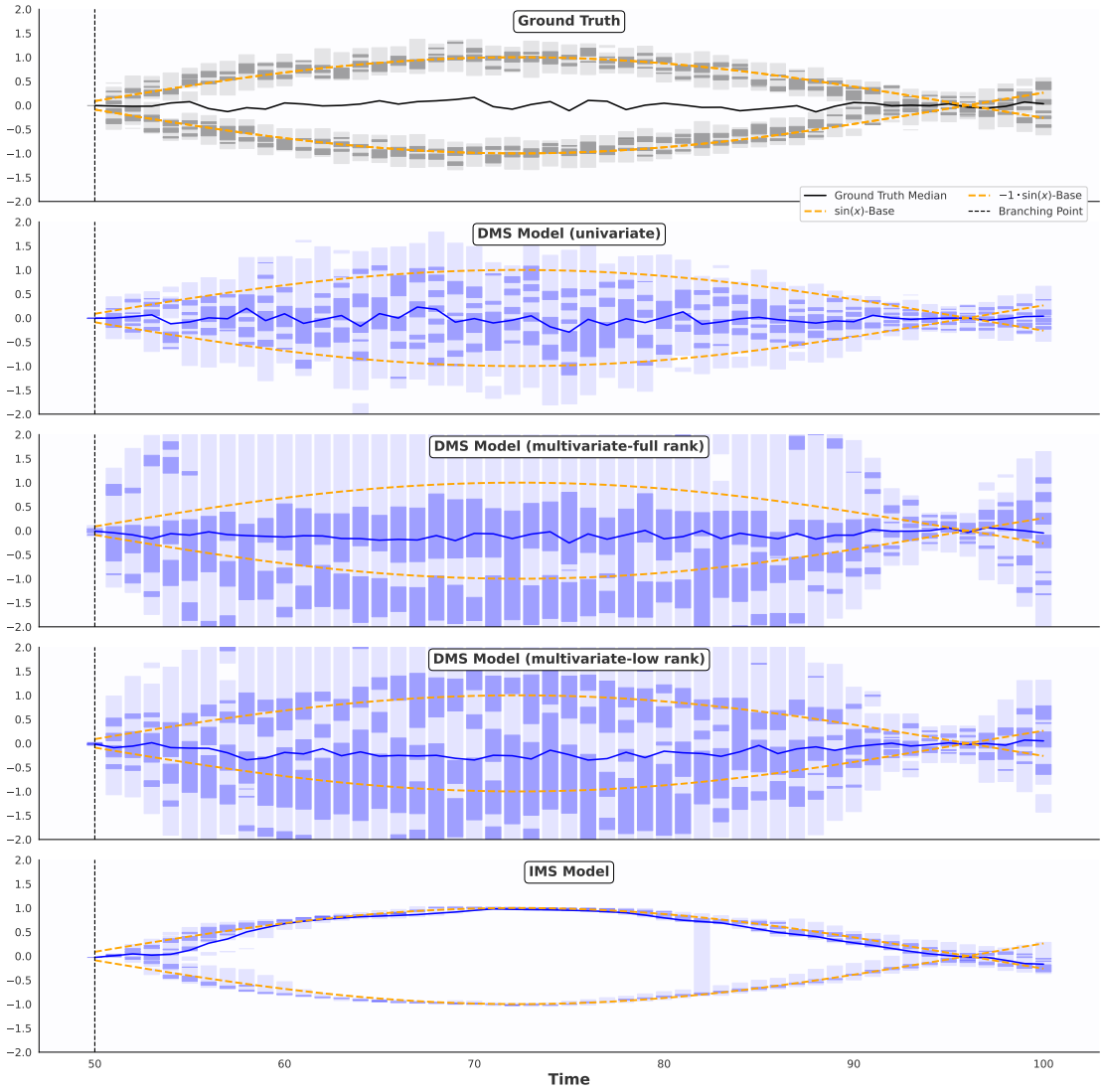


Figure C.3.: Visualization of kNN-approximated intervals. This better reflects the multimodal nature of the data.

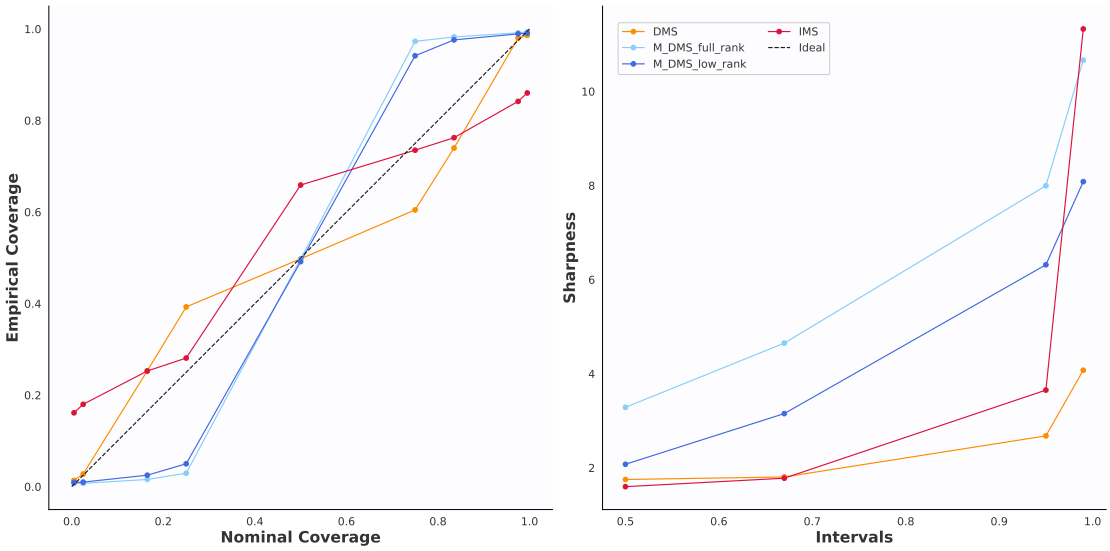


Figure C.4.: Calibration (left) and sharpness (right) curves for our four models based on the central intervals of 100 samples.

C.2. Probabilistic LTSF

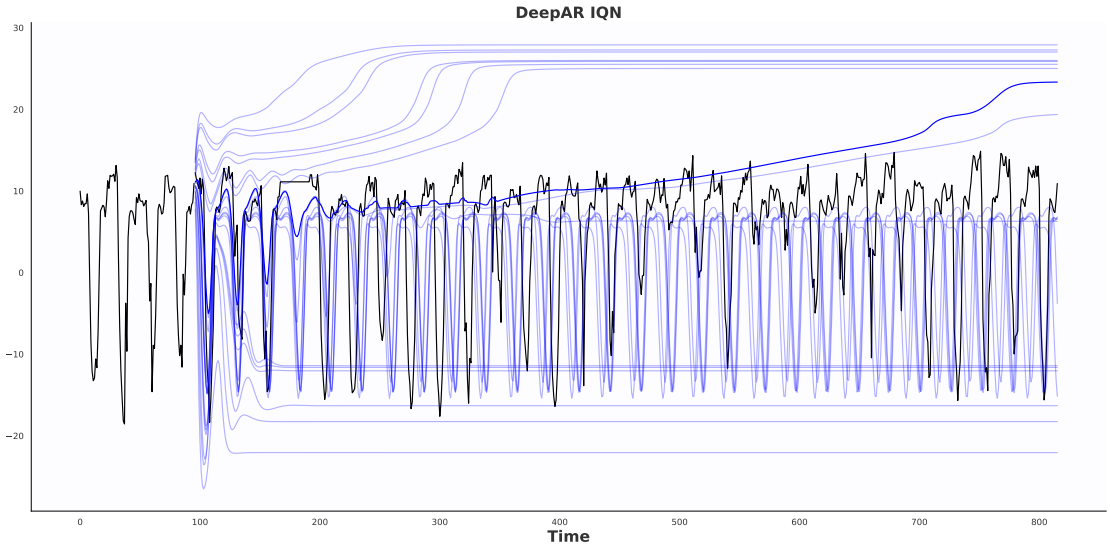


Figure C.5.: Visualization of the predictions of the IQN DeepAR model on ETTh1 with a constant quantile level for each trajectory. It becomes clear that the constant quantile level causes the model to diverge into unrealistic regions on many occasions.

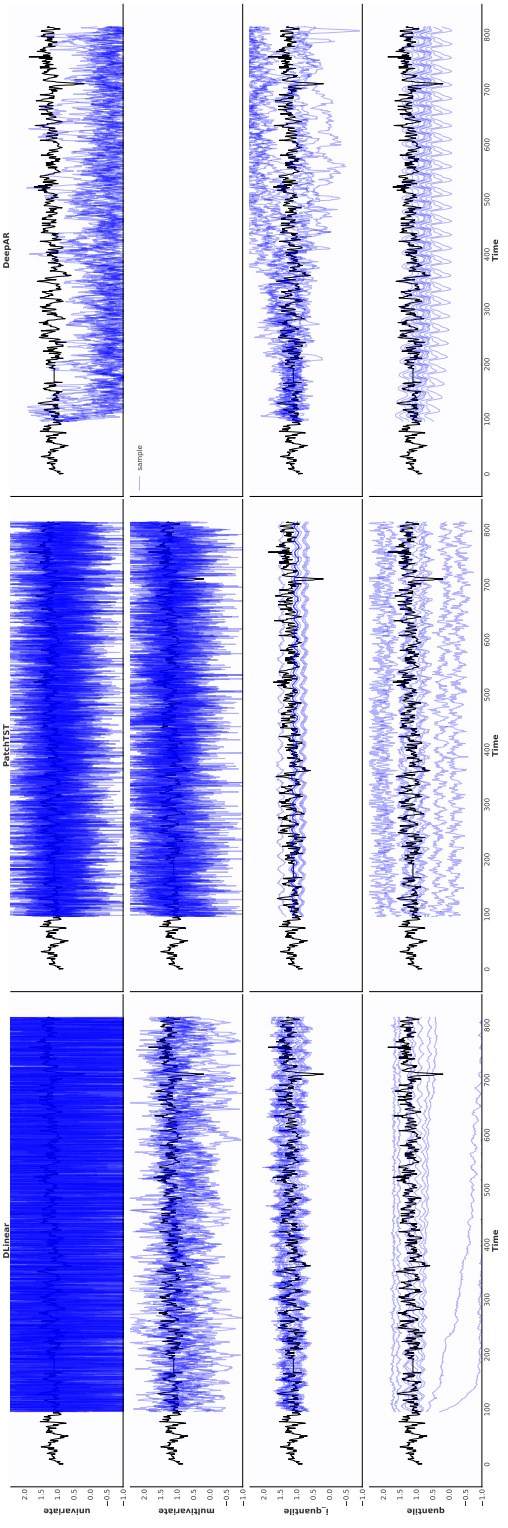


Figure C.6.: 20 sample forecasts per model for an instance of the LULL test time series, part of the ETTh1 dataset.

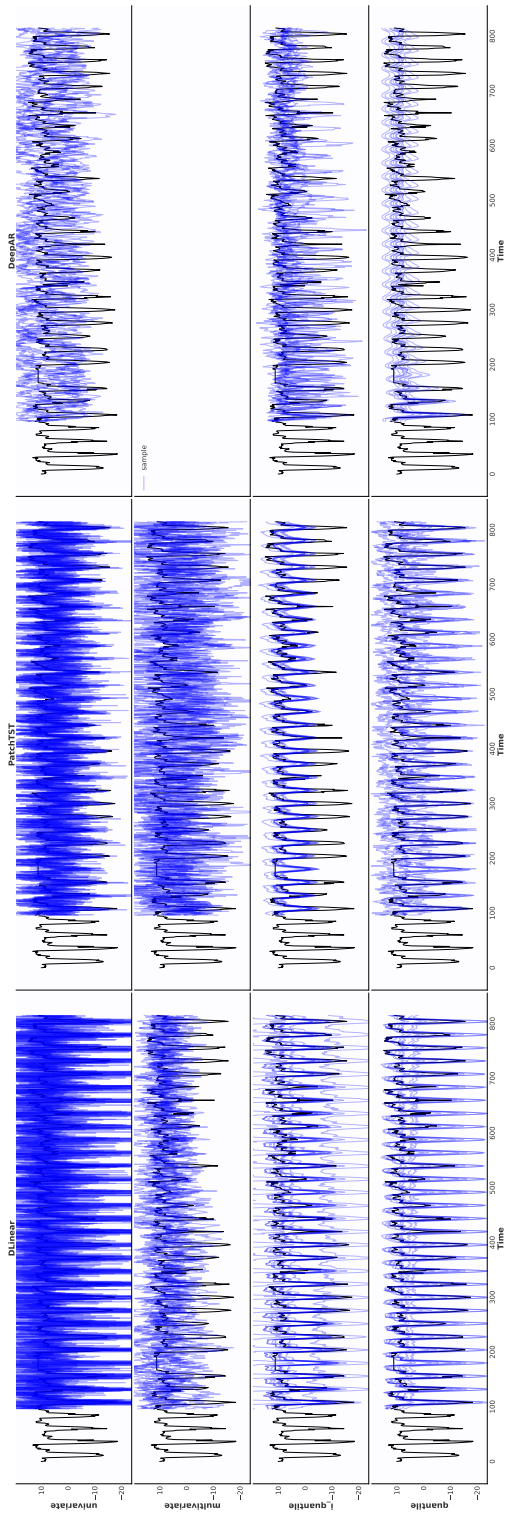


Figure C.7.: 20 sample forecasts for an instance of the HUFL test time series.

Figure C.8.: Sample forecasts for two ETTh1 test time series. Each panel shows 20 forecast samples in blue and the corresponding ground truth in black. For the quantile-based methods, the "samples" shown correspond to the nine predicted quantile trajectories, as no sampling is performed in these cases.

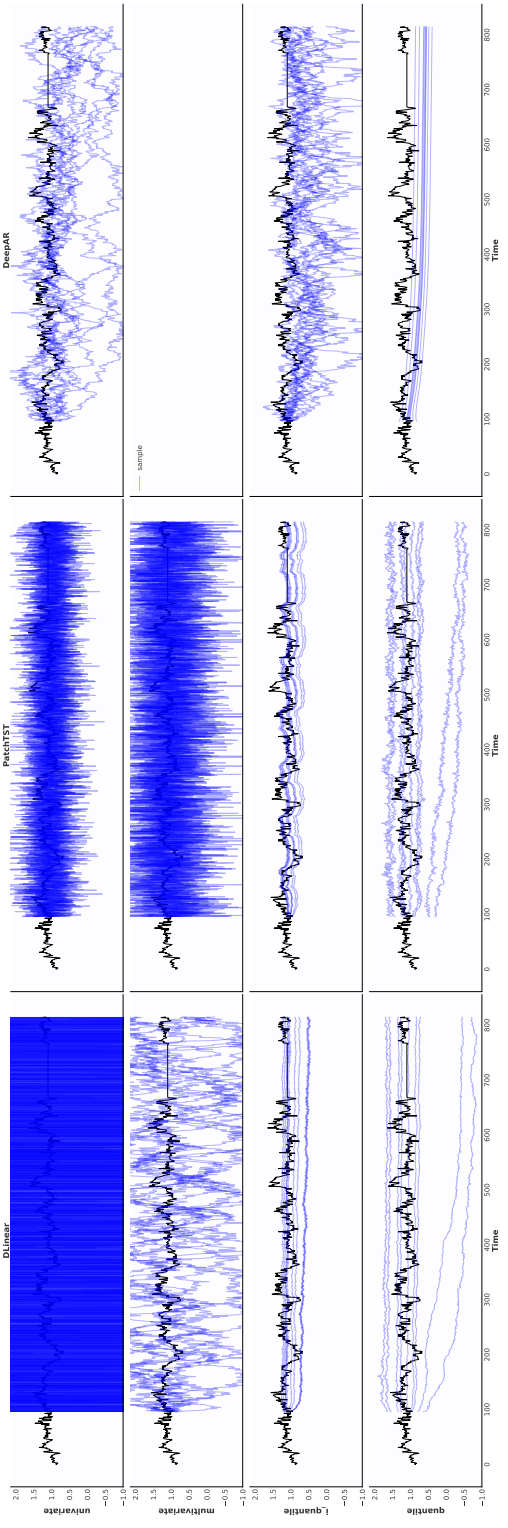


Figure C.9.: 20 sample forecasts per method for an instance of the LULL test time series, part of the ETTm1 dataset.

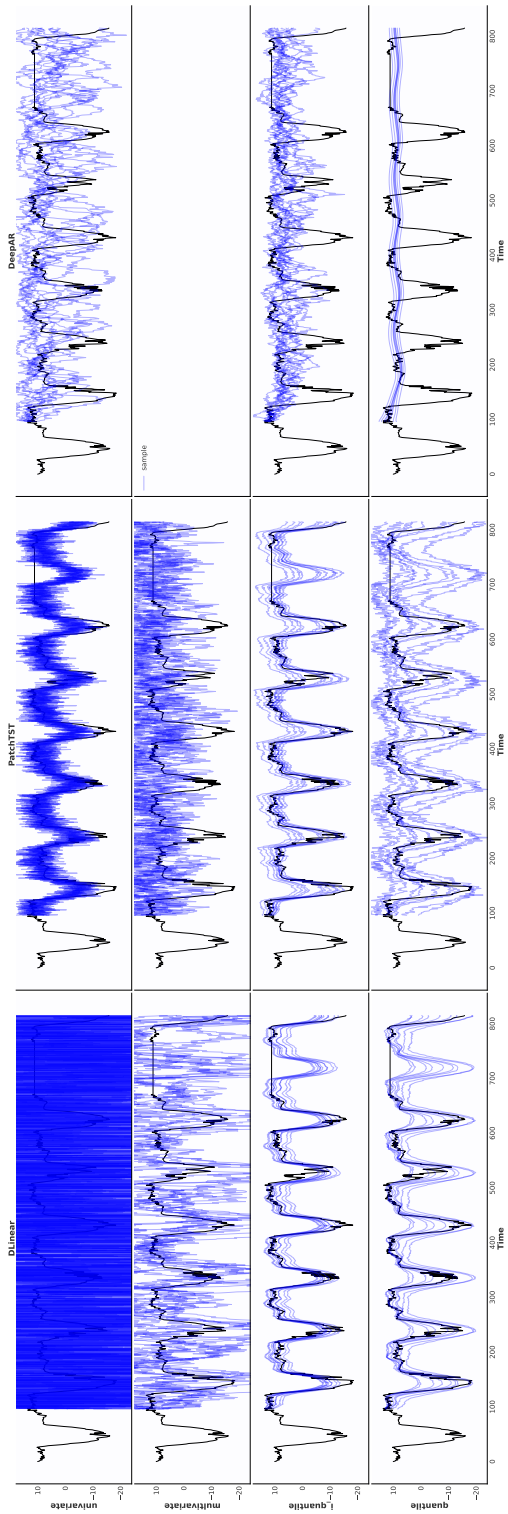


Figure C.10.: 20 sample forecasts for an instance of the HUFL test time series.

Figure C.11.: Sample forecasts for two ETTm1 test time series. Each panel shows 20 forecast samples in blue and the corresponding ground truth in black. For the quantile-based methods, the "samples" shown correspond to the nine predicted quantile trajectories, as no sampling is performed in these cases.

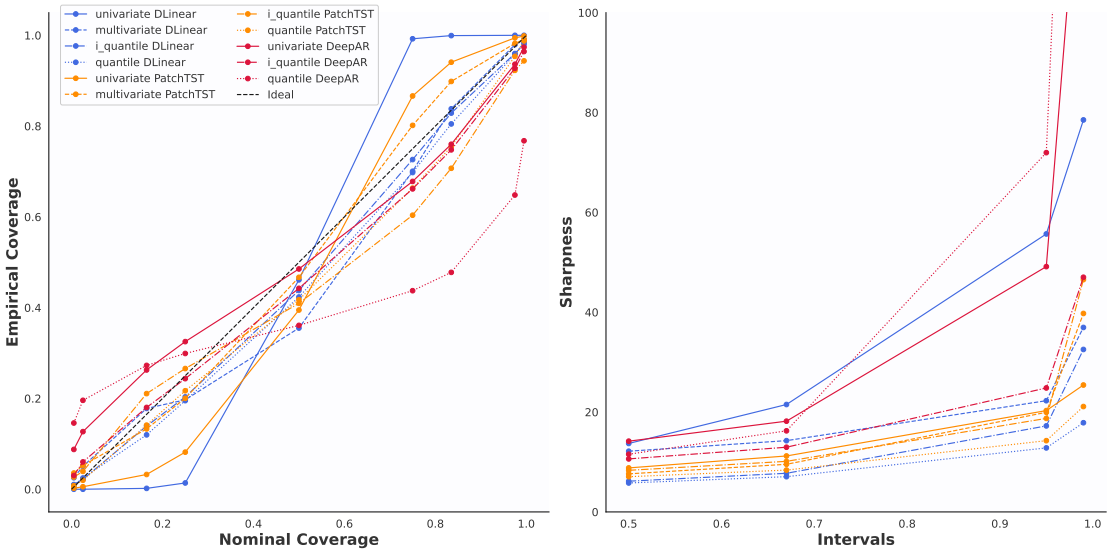


Figure C.12.: Calibration (left) and sharpness (right) curves for our probabilistic LTSF experiments on the ETTh1 data set, in which the single calibration plots aids in the comparison of inter-model differences.

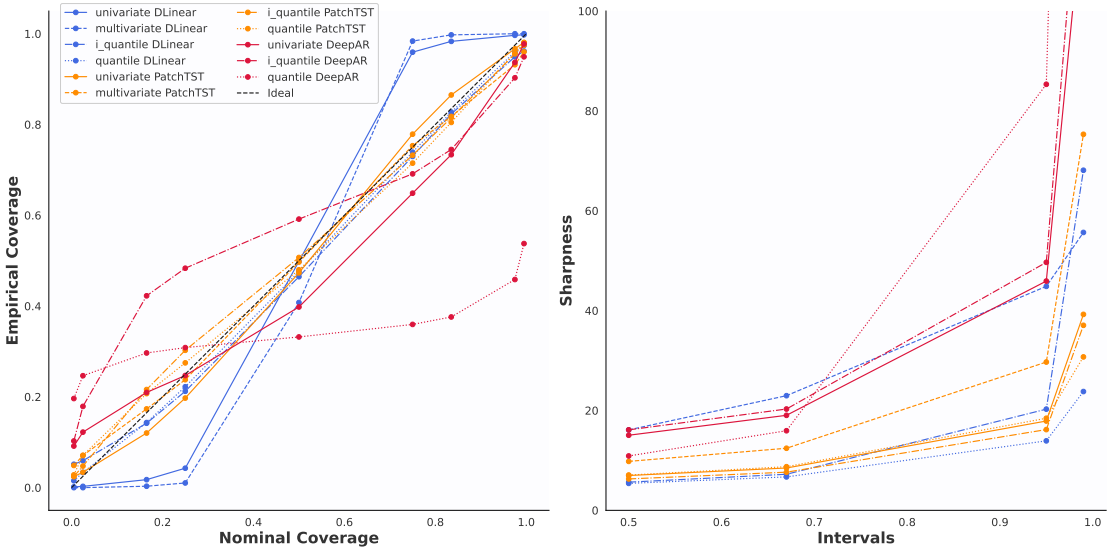


Figure C.13.: Calibration (left) and sharpness (right) curves for our probabilistic LTSF experiments on the ETTm1 data set, in which the single calibration plots aids in the comparison of inter-model differences.

Ehrenwörtliche Erklärung

Ich versichere, dass ich die beiliegende Bachelor-, Master-, Seminar-, oder Projektarbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und in der untenstehenden Tabelle angegebenen Hilfsmittel angefertigt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Declaration of Used AI Tools

Tool	Purpose	Where?	Useful?
ChatGPT	Rephrasing	Throughout	+
DeepSeek	Rephrasing	Throughout	+
GPT-4	Code generation	Codebase	+
Claude	Code generation	Codebase	+

Unterschrift
Mannheim, den 14.Juli 2025