QuantileFormer: Probabilistic Time Series Forecasting with a Pattern-Mixture Decomposed VAE Transformer

Yimiao Shao , Wenzhong Li* , Kang Xia , Kaijie Lin , Mingkai Lin , Sanglu Lu
Nanjing University
yimiao_shao@smail.nju.edu.cn, lwz@nju.edu.cn,
{xiakang, kaijie}@smail.nju.edu.cn, {mingkai, sanglu}@nju.edu.cn

Abstract

Probabilistic time series forecasting has attracted an increasing attention in machine learning community for its potential applications in the fields of renewable energy, traffic management, healthcare, etc. Previous research mainly focused on extracting long-range dependencies for point-wise prediction, which fail to capture complex temporal patterns and statistical characteristics for probabilistic analysis. In this paper, we propose a novel pattern-mixture decomposition method that decomposes long-term series into quantile drift, divergence patterns, and Gaussian mixture components, which can effectively capture the intricate temporal patterns and stochastic characteristics in time series. Based on pattern-mixture decomposition, we propose a novel Transformer-based model called QuantileFormer for probabilistic time series forecasting. It takes the the comprehensive drift-divergence mixture patterns as features, and designs a variational inference based fusion Transformer architecture to generate quantile prediction results. Extensive experiments show that the proposed method consistently boosts the baseline methods by a large margin and achieves state-of-the-art performance on six real-world benchmarks.

1 Introduction

Recently, with an enhanced understanding of uncertainty, probabilistic time series forecasting has garnered increasing attention for its potential applications in the areas of renewable energy [Zheng *et al.*, 2023; Huy *et al.*, 2022], traffic management [Zhang *et al.*, 2022; Jiang *et al.*, 2024], healthcare [Caldas and Soares, 2022], etc.

The primary objective of probabilistic time series forecasting is to provide probability distribution information regarding uncertainty for predicting values at future time points. Unlike traditional time series forecasting, probabilistic forecasting aims to comprehensively describe the potential range of future values, which is achieved by estimating various quantiles (including median and percentiles) to offer a range of potential outcomes, thereby enhancing decision-making under uncertainty.

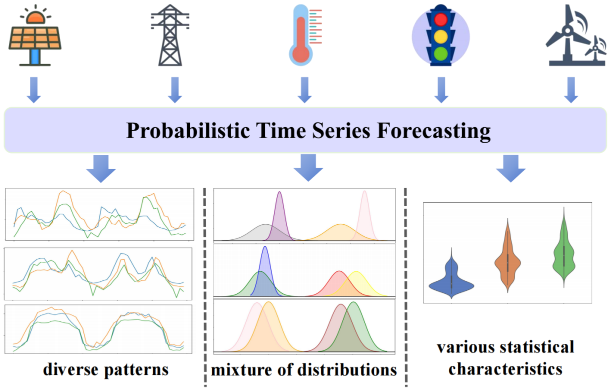

Probabilistic Time Series Forecasting
diverse patterns
mixture of distributions
various statistical characteristics

Figure 1: Illustration of mixture patterns in the Electricity dataset. It contains diverse patterns in different time period, with mixture distribution parameters and various statistical characteristics.

Deep neural networks (DNNs) [Lin *et al.*, 2022; Hong *et al.*, 2024; Hong *et al.*, 2025] have been increasingly used in probabilistic time series forecasting and demonstrated a promising performance. For instance, DeepAR [Salinas *et al.*, 2020] proposed to train autoregressive recurrent neural networks on a large set of correlated time series. MQ-R(C)N [Wen *et al.*, 2017] explored the non-parametric nature of quantile regression and introduced a framework for general probabilistic multi-step time series regression with direct multi-level predictions. Recently, Transformer-based models are introduced for probabilistic time series forecasting due to its great success in sequential learning. MQTransformer [Eisenach *et al.*, 2020] proposed a novel decoder-encoder attention for context-alignment to improve quantile forecasting accuracy. TFT [Lim *et al.*, 2019] used recurrent layers for local processing and self-attention layers to capture long-term dependencies for probabilistic forecasting.

However, these methods struggle to accurately capture the intricate temporal dynamics and statistical properties of long-term time series. Given the perpetual evolution of economic, social, and environmental factors, time series data are prone to concept drift, which can diminish the performance of previously effective models in new contexts. As depicted in Figure 1, real-world time series data commonly displays diverse

[*] Corresponding author

patterns, mixture of distributions, as well as various statistical characteristics, posing challenges for conventional methods to precisely capture such features. First, it is difficult to extract temporal patterns which are entangled and diversified. Second, the mixed distribution of data exacerbates the challenge of capturing probabilistic distribution information. Third, the diverse statistical properties of data complicate models' ability to simultaneously capture quantile information from multiple variates.

To tackle the aforementioned challenges, we propose a novel framework called *QuantileFormer* based on a pattern-mixture decomposition method for probabilistic time series forecasting. To depict the intricate temporal patterns and stochastic characteristics in time series data, we propose a *pattern-mixture decomposition* approach to decompose the long-term time series into different pattern components to facilitate analysis. Pattern-mixture decomposition is composed of two key components: drift-divergence and Gaussian mixture decomposition. The drift-divergence decomposition employs a quantile filter to break down the original time series into quantile drift and divergence pattern components, thereby enhancing the model's predictive power. Following this, Gaussian mixture decomposition is utilized to further decompose the divergence patterns into a blend of multiple Gaussian distributions. This step assigns probabilities to each data point, allowing the model to quantify the uncertainty associated with predictions by identifying which distribution each point belongs to. The Gaussian distribution components are then processed by a *variational inference network*, which extracts information about the overall statistical properties of the time series data. We further design a *fusion Transformer* architecture that integrates these pattern components holistically to form the final quantile predictions for probabilistic time series forecasting.

The contributions of this paper are summarized as follows.

- We propose a pattern-mixture decomposition method that decomposes long-term time series into quantile drift, divergence patterns, and Gaussian mixture components, which can effectively capture the intricate temporal patterns and stochastic characteristics in time series data.
- We propose a novel Transformer-based model called QuantileFormer for probabilistic time series forecasting. Based on pattern-mixture decomposition, the quantile drift part is proceeded by a Transformer encoder and the statistical patterns are captured by a Variational AutoEncoder (VAE) network, which are fed into a fusion Transformer to obtain the quantile prediction results.
- We conduct comprehensive experiments to rigorously assess the efficacy of our proposed method. In addition to employing conventional metrics, we introduce a new performance metric, *cpaw* (*Coverage Probability with Normalized Averaged Width*), specifically designed to quantify the precision of the predicted probabilistic intervals. Experimental results show that the proposed method consistently outperforms the baseline methods by a large margin and achieves state-of-the-art performance on six real-world benchmarks.

2 Related Work

2.1 Transformer-based Models

Transformer-based models were widely used in time series forecasting. Pyraformer [Liu *et al.*, 2022] proposed a novel pyramidal attention based Transformer to bridge the gap between capturing the long-range dependencies and reducing time and space complexity. PatchTST [Nie *et al.*, 2022] used patch to provide a longer sequence of inputs to extract meaningful temporal relationships and channel independence to predict multivariate time series. iTransformer [Liu *et al.*, 2023] encoded each variable as an independent token, and used the feedforward network to model the temporal correlation of variable variables to obtain better sequential temporal representation. However, the above mentioned works mainly focused on point-wise forecasting, and very few works adopted Transformer for probabilistic forecasting [Eisenach *et al.*, 2020; Lim *et al.*, 2019].

2.2 Decomposition of Time Series

In the realm of time series analysis, the standard methodology of time series decomposition [Cleveland *et al.*, 1990; Tukey, 1960; Hyndman and Khandakar, 2008; De Jong, 1980; McCullough and Renfro, 1990] dissects a temporal sequence into several components, each representing a more predictably discernible underlying pattern. When applied to forecasting tasks, decomposition serves as an essential preprocessing step for historical series prior to predicting future sequences. Examples include the application of trend-seasonality decomposition in models like Autoformer [Wu *et al.*, 2021] and FeDFormer [Zhou *et al.*, 2022], period decomposition in TimesNet [Wu *et al.*, 2022], basis expansion in N-BEATS [Oreshkin *et al.*, 2019], and matrix decomposition in DeepGLO [Sen *et al.*, 2019]. TS3Net [Ma *et al.*, 2024] expanded the time series into a 2D temporal-frequency distribution and decoupled a long-term series into trend-part, regular-part, and fluctuant-part. TimeMixer [Wang *et al.*, 2024] extracted past information and blended seasonal and trend components at different scales separately.

2.3 Probabilistic Time Series Forecasting Methods

To capture the parts of the sequence that reflect the probability distribution, several methods have been applied to probabilistic time series forecasting [Bontempi and Ben Taieb, 1999; Hyndman and Athanasopoulos, 2018; Bergmeir and Hyndman, 2015; Salinas *et al.*, 2018; Wang *et al.*, 2021]. P-TSE [Zhou *et al.*, 2023] proposed a multi-model distribution ensemble method which abstracts the transformation of the model into Hidden Markov Model. Conformalized quantile regression [Romano *et al.*, 2019] involved regressing the recentered influence function (RIF) of the quantile functional over input covariates to obtain unconditional quantile regression. TimeGrad [Rasul *et al.*, 2021] proposed an autoregressive model for multivariate probabilistic forecasting that leverages the exceptional performance of EBMs to learn from the distribution of the next time step. TMDM [Li *et al.*, 2024] took into account the covariate dependence of forward and reverse processes in the diffusion model to achieve highly accurate distribution estimation of future time series.

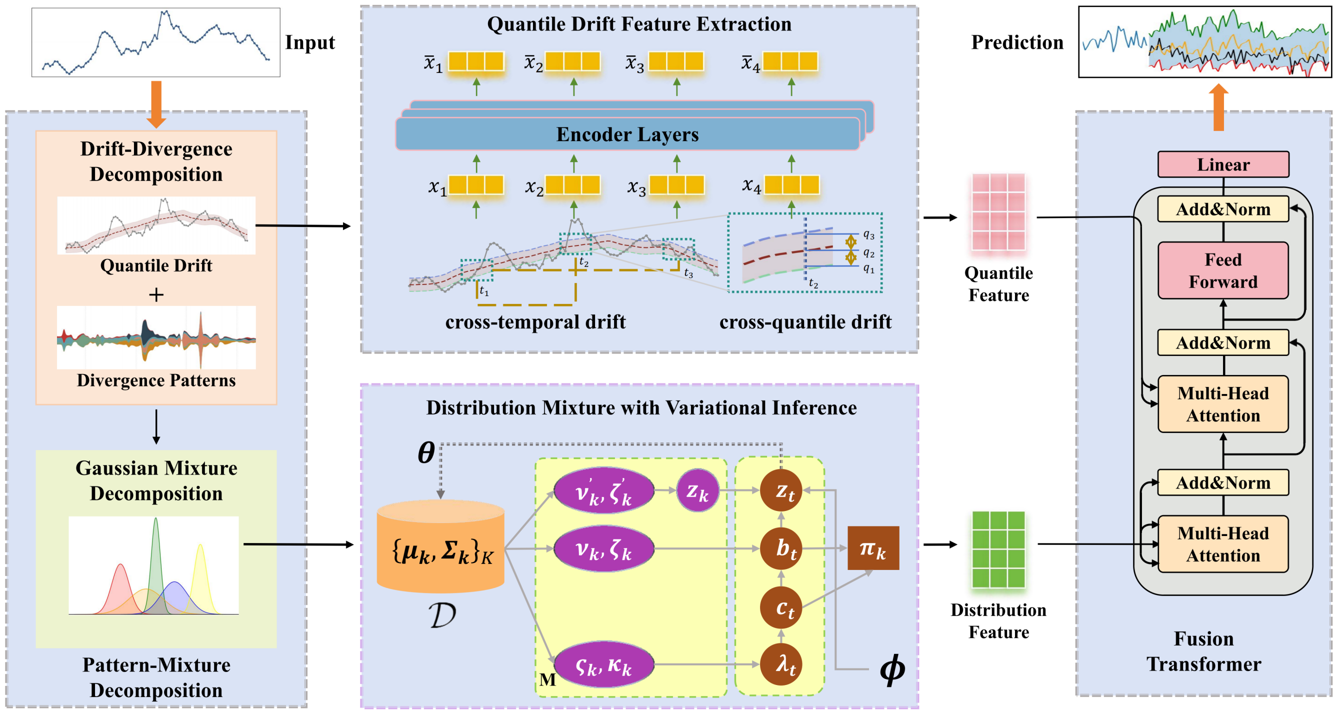


Figure 2: Architecture of QuantileFormer, it consists of a pattern-mixture decomposition, a quantile drift feature extraction, a variational inference and a fusion Transformer components.

To the best of our knowledge, we are the first to propose a pattern-mixture method that decomposes long-term series into a mixture of quantile patterns, and design a fusion Transformer architecture for probabilistic time series forecasting.

3 Problem Formulation

Probabilistic time series forecasting focuses on predicting the quantiles for each time point of a time series. Quantiles are essential statistical measures that provide insights into the distributional properties of a random variable. In the context of probability distributions, a quantile represents a critical value below which a specified proportion of the distribution lies. For a given probability p, the p-quantile is defined as the value x_p such that $P(X \leq x_p) = p$, where X is a random variable and P denotes the probability measure. The quantile function, often denoted as $Q(p)$, maps a probability p to the corresponding quantile x_p, which is expressed as:

$$Q(p) = inf\{x : P(X \leq x_p) \geq p\}. \tag{1}$$

Considering the rolling forecasting setting with a fixed size window, we have the observations at T time points, represented by $X = \{x_i | i = 1, 2, ...T\}$. The objective is to perform quantile regression for time series analysis, i.e., estimating the conditional quantiles of the response variable y at different percentiles τ, which can be expressed as:

$$Q_\tau(y_t|X_t) = X_t\beta_\tau, \tag{2}$$

where X_t denotes the vector of observed variables at time t; y_t is the predicted quantile at time t; β_t is the vector of coefficients for the τ-th quantile; and $Q_\tau(y_t|X_t)$ represents the τ-th conditional quantile of y_t given X_t.

The quantile regression problem can be formulated as the following optimization problem:

$$min_{\beta_t} \sum_{t=1}^{T} \rho_\tau(y_t - X_t\beta_\tau), \tag{3}$$

where $u = y_t - X_t\beta_\tau$ is the residual for the t-th observation; $\rho_\tau(u)$ is the loss function that penalizes the residuals with respect to the τ-th quantile.

4 QuantileFormer Model

We propose a framework called QuantileFormer for probabilistic time series forecasting, which is illustrated in Figure 2. It begins with a *pattern-mixture decomposition* method, which meticulously breaks down long-term time series data into distinct pattern components to capture essential temporal dynamics. This decomposition process is composed of two main elements: a Drift-Divergence decomposition that isolates quantile drift and divergence patterns, and a Gaussian mixture decomposition that characterizes the global statistical properties of the data. These components provide a detailed description of the temporal and statistical features inherent in the time series. To further refine the analysis, we incorporate a *quantile drift feature extraction* module that adeptly extracts meaningful features from the quantile drift. Concurrently, a *variational inference network* is employed to deduce broader distributional insights from the divergence patterns. Finally, to synthesize the insights gained, we propose a *fusion Transformer* module, which fuses the pattern components to form the final quantile prediction results, thereby offering a comprehensive and robust approach for probabilistic time series forecasting.

4.1 Pattern-Mixture Decomposition

The pattern-mixture decomposition consists of two submodules: a drift-divergence decomposition and a Gaussian mixture decomposition, which are introduced as follows.

Drift-Divergence Decomposition

Conventional methods for sequence decomposition roughly separate the series into the trend-cyclical and seasonal parts. However, these two parts contribute little to quantile regression due to their nature of statistics. To this end, we propose a drift-divergence decomposition method to capture different quantile levels and complex composite patterns within the sequence, referring to as *Quantile Drift* and *Divergence Patterns*. For each quantile q in the quantile set $\mathcal{Q}$, we extract the drift component χ^q of the original series using a sliding window. We use $\chi^{\mathcal{Q}}$ to represent the set containing all the drift components, i.e., $\chi^{\mathcal{Q}} = \{\chi^q\}_{q\in\mathcal{Q}}$. Specifically, for each q, we calculate the drift component χ^q and the divergence component χ^d as follow:

$$\begin{aligned}\chi^q &= QuantileFilt(Padding(\chi), q),\\ \chi^d &= \chi - \chi^{0.5},\end{aligned} \tag{4}$$

where χ^d denotes the divergence patterns obtained by subtracting the median $\chi^{0.5}$ from the original series χ; and $QuantileFilt(\cdot, q)$ represents the moving q-th quantiles with padding operation to keep the series length unchanged.

Gaussian Mixture Decomposition

After drift-divergence decomposition, the quantile drift $\chi^{\mathcal{Q}}$ represents smooth components of the time series, and the divergence component χ^d contains complex periodic patterns and distribution characteristics. We propose a Gaussian mixture decomposition method to further capture statistical patterns from the divergence component χ^d.

Gaussian Mixture Models (GMM) [Ng and Jordan, 2001] is a probabilistic model that represents a mixture of multiple Gaussian distributions. The probability density function for a single Gaussian distribution is:

$$f(x|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}, \tag{5}$$

where x is the data vector; μ is the mean vector; Σ is the covariance matrix; and d is the dimensionality of the data. We hope to find a set of $\Theta = \{\mu_k, \Sigma_k\}_k$ based on which the Gaussian distribution ensemble can optimally fit the divergence patterns χ^d. The probability of the set Θ given the divergence patterns χ^d can be denoted as:

$$\mathcal{L}(\Theta|\chi^d) = \Pi_{i=1}^{N} P(x_i; \Theta). \tag{6}$$

GMM decomposition aims to maximize the above likelihood function, which can be achieved by an iterative optimization algorithm such as Expectation-Maximization. We use $GauDe(\cdot)$ to summarize the above operations. Thus, we have

$$\mathcal{D} = GauDe(\chi^d), \tag{7}$$

where $\mathcal{D} = \{\mu_k, \Sigma_k\}_{k=1}^{K}$ represents the K Gaussian components which optimally fits χ^d.

4.2 Distribution Mixture Inference with Variational AutoEncoder (VAE)

Due to the intricate nature of data distribution, the local distributions do not linearly constitute the global distribution in a straightforward manner, thereby complicating the derivation of the target distribution. The Gaussian components $\mathcal{D} = \{\mu_k, \Sigma_k\}_{k=1}^{K}$ conveys essential information about the local distribution of the time series. We use $\hat{\mathcal{D}}$ to denote the target global distribution which has K components:

$$\hat{\mathcal{D}} = \sum_{k=1}^{K} \pi_k \mathcal{D}_k, \tag{8}$$

where $\mathcal{D}_k$ is the kth Gaussian component and $\pi_k \geq 0, \sum_{k=1}^{K} \pi_k = 1$. Note that each time step data can be allocated into some of the K components [Ioffe and Szegedy, 2015]. We further introduce the following notations to describe the model.

- $c_t \in \{0,1\}^K$ is a binary vector representing the distribution allocation, where $c_{tk} = 1$ represents the distribution of the tth time step is allocated to the $k$the Gaussian component.
- $b_t = \{b_{tk} \in [0,1] | k = 1, ...K\}$, subject to $\sum_{k=1}^{K} b_{tk}c_{tk} = 1$, represents the contribution of the tth time step which are hyperparameters in the proposed distribution inference network. Noted that the contribution $b_{tk} \neq 0$ only when the corresponding allocation component $c_{tk} = 1$.

Since the true distribution is unknown, we propose a Variational AutoEncoder (VAE) method to approximate the optimal parameters π_k, c_t and b_t of the global distribution.

Using the stick-breaking construction of the Indian Buffet Process (IBP) [Griffiths and Ghahramani, 2011], we infer that c_t is sampled from a Bernoulli distribution parameterized by $\lambda_t = \{\lambda_{tk} | k = 1, ...K\}$, where λ_t is sampled in i.i.d from a Beta distribution $Beta(\varsigma_k, \kappa_k)$ parameterized by ς_k, κ_k. Similarly, we infer that b_t is sampled from a Gaussian prior distribution $\mathcal{N}(\nu_k, \zeta_k)$ which is parameterized by ν_k and ζ_k.

We denote $z_t = \sum_{k=1}^{K} b_{tk}\dot{z}_t$, which is a latent variable used by a variational decoder θ to reconstruct the given $\mathcal{D}$. In the expression, z_t means the latent vector sampled from every allocated distribution of tth time step from Gaussian prior distribution $\mathcal{N}(\nu_k', \zeta_k')$, which can adaptively adjust the latent posterior to a suitable probabilistic distribution [Duan *et al.*, 2023].

As illustrated in Figure 2, the parameters of $Beta(\varsigma_k, \kappa_k)$, $\mathcal{N}(\nu_k, \zeta_k)$ and $\mathcal{N}(\nu_k', \zeta_k')$ can be inferred with a variational encoder ϕ based on the Gaussian components $\mathcal{D}$, i.e., $\{\nu_k, \zeta_k, \nu_k', \zeta_k', \varsigma_k, \kappa_k\} = \phi(\mathcal{D})$. Meanwhile, b_t and z_k contribute to the calculation of a latent variable z_t, which is then fed to a decoder θ.

In order to infer the latent vector z_t, we should derive the variational posterior $q_\phi(\lambda_t, c_t, b_t)$. From Figure 2 we know that variables in variational posterior are conditionally independent with the priori $p(\mathcal{D})$. So we can decouple the variables as: $q_\phi(\lambda, c, b) = \prod_{k=1}^{K}\prod_{m=1}^{M} q_\phi(b_{tk}) \cdot q_\phi(c_{tk}|\lambda_{tk}) \cdot$

$q_\phi(\lambda_{tk})$, where the variational posterior distributions can be derived as[Cote, 2016]:

$$b_t \sim \mathcal{N}(\nu_k, \zeta_k),\ \lambda_t \sim Beta(\varsigma_k, \kappa_k),\ c_t \sim Bernoulli(\prod_{k=1}^{K} \lambda_{tk}). \quad (9)$$

Given the model structure, the component weights π_k can be inferred from the allocation vector c_t and the contribution values b_t. To derive the component weight π_k, we use a softmax function[Dempster *et al.*, 1977] over the expected contribution of each component across all time steps:

$$\pi_k = \frac{exp(\frac{1}{K} S_k)}{Z},\ where\ S_k = \sum_{t=1}^{T} q_\phi(c_{tk}) \cdot b_{tk}, \quad (10)$$

and Z is a normalization constant:

$$Z = \sum_{k=1}^{K} exp(\frac{1}{K} \sum_{t=1}^{T} q_\phi(c_{tk}) \cdot b_{tk}). \quad (11)$$

By optimizing the parameters, the optimal b_t and c_t can be derived, which can be further used to derived the distribution weights π_k.

We then introduce the algorithm to optimize the VAE based on the derivation in the above section. For convenient, we omit the latent variables $\{b_t, c_t, \lambda_t\}$ and their priors in representing the encoder model ϕ.

The true posterior $p_\theta(z_t|\mathcal{D})$ is typically intractable, thus we approximate it with $q_\phi(z_t|\mathcal{D})$ .by minimizing their KL-divergence:

$$\phi^*, \theta^* = \arg\min_{\theta,\phi} \mathbb{D}_{KL}(q_\phi(z_t|\mathcal{D})||p_\theta(z_t|\mathcal{D}), \quad (12)$$

where

$$\mathbb{D}_{KL}(q_\phi(z_t|\mathcal{D})||p_\theta(z_t|\mathcal{D}) = \int q_\phi(z_t|\mathcal{D}) \log \frac{q_\phi(z_t|\mathcal{D})}{p_\theta(z_t|\mathcal{D})} dz_t. \quad (13)$$

Since directly computing $p_\theta(z_t|\mathcal{D})$ is difficult, we optimize the ELBO (Evidence Lower BOund) [201, 2012].

$$\begin{aligned} &\mathbb{E}_{q_\phi(z_t|\mathcal{D})}[\log \frac{p_\theta(z_t, \mathcal{D})}{q_\phi(z_t|\mathcal{D})}] = \\ &\mathbb{E}_{q_\phi(z_t|\mathcal{D})}[\log \frac{p(z_t)}{q_\phi(z_t|\mathcal{D})}] + \mathbb{E}_{q_\phi(z_t|\mathcal{D})}[\log p_\theta(\mathcal{D}|z_t)]. \end{aligned} \quad (14)$$

According to the theory of variational inference [Kingma, 2013], the above problem can be solved with the SGD method using a nonlinear deep neural network to optimize the mean squared error loss function.

We summarize the above operation as $VAE(\cdot, \cdot)$. Thus, we can obtain indications of the global distribution by

$$\chi^d_{out} = VAE(\chi^d, \mathcal{D}), \quad (15)$$

which linearly projects each time point in χ^d onto a Gaussian distribution in $\mathcal{D}$. The output χ^d_{out} contains rich global distribution information providing insights into the shape, spread, and central tendency of the time series, which can facilitate the subsequent probabilistic time series forecasting task.

4.3 Quantile Drift Feature Extraction

The decomposed drift component $\chi^{\mathcal{Q}}$ encapsulates a wealth of drift information within the sequence. Recognizing the heterogeneity in these drifts, we incorporate both cross-time drift and cross-quantile drift into our considerations. Cross-time drift captures temporal interactions between data at different time steps, while cross-quantile drift reveals disparities in trends across quantile levels.

We apply Transformer encoder on $\chi^{\mathcal{Q}}$ to capture the drift features. The Transformer encoder consists of multiple identical layers (typically 6 layers). Each encoder layer consists of two sub-layers: a Multi-Head Self-Attention Layer and a Fully Connected Feedforward Layer. Residual connections and layer normalization are also included between these two sublayers. We use $\chi_{eout} = Encoder(\chi)$ to represent the Transformer encoder. Thus we have $\chi^{\mathcal{Q}}_{eout} = \{Encoder(\chi^q)\}_{q \in \mathcal{Q}}$ to denote the encoder output of the quantile drift feature.

4.4 Fusion Transformer with Cross-Attention

To fuse the output features from different components to form the final prediction, we design a Fusion Transformer with cross-attention to establish a soft correspondence between the drift-divergence (i.e., $\chi^{\mathcal{Q}}_{eout}$ and χ^d_{out}) , as illustrated in the right part of Figure 2. We first align χ^d_{out} with $\chi^{\mathcal{Q}}_{eout}$ using a linear projection W^a. Then we adopt three linear projections $\mathbf{W}^K$, $\mathbf{W}^Q$, $\mathbf{W}^V$ to generate the Query-Key-Value triples as follows.

$$\mathbf{Q} = \chi^d_{out} \cdot \mathbf{W}^a \cdot \mathbf{W}^Q,\ \ \mathbf{K} = \chi^{\mathcal{Q}}_{eout} \cdot \mathbf{W}^K,\ \ \mathbf{V} = \chi^{\mathcal{Q}}_{eout} \cdot \mathbf{W}^V. \quad (16)$$

We then apply cross-attention among $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ and following by a FeedForward Network (FFN) to enhance the expressive capability of the model. We highlight the core calculation process as:

$$\begin{aligned} Fusion = LayerNorm(&SelfAtt(\mathbf{Q}, \mathbf{Q}, \mathbf{Q}) \\ &+ CrossAtt(Input, \mathbf{K}, \mathbf{V}) \\ &+ FFN(Input)), \end{aligned} \quad (17)$$

where $Att(\cdot, \cdot, \cdot)$ is the multi-head attention module.

The residual connections allow the network to retain the original Gaussian mathematical implications, which contain quantile drift and Gaussian components information to enrich the final predictions. Thus the final output of prediction result is

$$\hat{y} = \mathbf{W}(Fusion), \quad (18)$$

where $\mathbf{W}$ is the parameter of the linear prediction head.

4.5 Loss Function

In order to synthesize the information of the context vectors, we train our model by combining the losses of three parts, and each part of the loss is measured by a quantile loss function. In line with previous works [Wen *et al.*, 2017; Lim *et al.*, 2019; Zhou *et al.*, 2023], we use a jointly quantile loss which sums across all quantile outputs for horizons in the future, i.e., $\tau \in 1..., \tau_{max}$, to train our model:

$$\mathcal{L}(\Omega, \mathbf{W}) = \textstyle\sum_{y_t \in \Omega} \sum_{q \in \mathcal{Q}} \sum_{\tau=1}^{\tau_{max}} \frac{q(y-\hat{y})_+ + (1-q)(\hat{y}-y)_+}{M\tau_{max}}, \quad (19)$$

where Ω is the domain of training data containing M samples, $\mathcal{Q}$ is the set of output quantiles, and $(\cdot)_+ = max(0, \cdot)$.

	Electricity					Wind				
	0.5	0.6	0.7	0.8	0.9	0.5	0.6	0.7	0.8	0.9
DeepAR	1.0002	1.1177	1.9544	1.2077	1.0830	1.0205	0.9987	0.7805	1.0182	1.4419
MQRNN	1.1648	1.5772	1.6336	1.8193	0.8273	2.1937	4.4670	5.5987	5.9560	1.8574
TFT	1.5547	1.0037	1.0440	0.8772	0.7618	0.9526	**0.8611**	0.7978	0.6568	0.4658
Transformer	1.3703	0.8873	1.0098	0.9005	0.9439	1.0011	1.0585	0.9898	0.9006	0.9750
Autoformer	1.0584	0.9191	1.0301	0.8786	0.6420	1.4353	1.6054	1.3345	0.9921	0.6361
FEDformer	1.9429	1.0447	0.9669	3.0007	1.0618	1.1361	1.0831	1.2615	0.6544	0.3876
PatchTST	1.8354	1.3134	1.0657	0.8800	0.7567	1.4666	0.9831	1.1394	0.9008	0.3667
iTransformer	1.3430	1.0348	1.2174	0.9072	1.2742	1.5983	1.0314	0.8091	0.6814	0.9900
QuantileFormer	**0.7469**	**0.8136**	**0.3330**	**0.4340**	**0.5121**	**0.8403**	0.9105	**0.7346**	**0.5842**	**0.3369**
	ETTm1					ETTh1				
	0.5	0.6	0.7	0.8	0.9	0.5	0.6	0.7	0.8	0.9
DeepAR	1.2026	1.1749	0.7901	1.0616	0.5388	2.3414	0.7631	1.2217	1.0815	1.9889
MQRNN	16.5845	21.9918	17.9190	12.0559	3.6909	1.4757	1.6722	1.0317	1.1949	1.2239
TFT	0.4930	0.7829	0.6769	0.4976	0.3513	1.4639	1.0443	0.9283	0.7382	0.3662
Transformer	1.0397	0.8740	0.7372	0.4998	0.3618	1.1989	0.8805	0.7284	0.4868	0.5546
Autoformer	1.8463	1.3424	1.1008	0.8392	0.4774	1.7221	1.2556	1.1977	0.9091	0.4569
FEDformer	0.6619	0.8673	0.4927	0.5491	0.3865	0.9480	0.8875	0.8328	0.7208	0.4582
PatchTST	1.4268	1.3088	1.0240	0.5100	0.2816	1.4719	1.4558	1.1307	0.4275	0.3166
iTransformer	0.7514	0.4112	0.8834	0.5824	0.1228	0.8850	0.9508	0.8607	0.4721	**0.3129**
QuantileFormer	**0.1536**	**0.1642**	**0.2689**	**0.4340**	**0.0596**	**0.3007**	**0.6130**	**0.2912**	**0.4273**	0.3388
	Solar					Traffic				
	0.5	0.6	0.7	0.8	0.9	0.5	0.6	0.7	0.8	0.9
DeepAR	**0.8666**	1.1173	1.2854	1.4512	1.6117	1.0502	0.8813	1.2484	0.9394	1.1539
MQRNN	0.8994	1.3492	1.0459	1.1921	1.7157	1.8146	2.2111	2.5796	2.9482	0.9940
TFT	1.0039	**1.1082**	1.2493	1.3740	1.0015	1.1494	0.8900	0.8500	**0.5862**	1.0570
Transformer	1.0391	1.1617	1.1381	1.0794	1.0777	0.9664	0.9325	1.0574	0.8679	0.9247
Autoformer	1.1641	1.2367	1.2088	1.0030	0.6167	0.9908	1.1109	0.8686	0.6064	0.4970
FEDformer	1.0363	1.1708	**1.0261**	1.5427	0.6414	2.4497	0.9188	2.3784	1.7356	0.8770
PatchTST	1.0806	1.1242	1.2547	1.1935	0.5950	0.9775	1.6937	1.1269	0.5962	1.1450
iTransformer	1.0705	1.1843	1.1845	1.3705	1.6083	1.8998	1.3545	1.1941	0.8247	1.5621
QuantileFormer	1.0641	1.0480	1.1832	**1.0008**	**0.5883**	**0.8489**	**0.8291**	**0.8489**	0.5998	**0.4688**

Table 1: Time series probabilistic forecasting results evaluated by q-risk (the lower the value, the better the performance), where predicted quantiles of 0.5, 0.6, 0.7, 0.8, and 0.9 are illustrated.

Dataset	Range	Freq	Samples	Features
Electricity	2016/7/1 to 2019/7/1	1h	26304	321
ETTm1	2016/7/1 to 2018/6/26	15min	69680	7
ETTh1	2016/7/1 to 2018/6/26	1h	17420	7
Wind	2020/7/1 to 2023/2/28	15min	93412	3
Traffic	2016/7/1 to 2018/7/2	1h	17544	861
Solar	2020/1/1 to 2023/1/31	15min	108192	5

Table 2: Dataset Information.

	Elec.	Wind	ETTm1	ETTh1	Traffic	Solar
deepar	5.2890	5.4470	3.8999	8.6446	4.8742	11.2021
mqrnn	3.8166	2.8071	8.4531	5.2274	1.6137	5.6390
tft	2.0002	2.4662	2.6199	**2.1166**	3.0367	1.7246
Transformer	-	-	**0.8988**	-	-	2.3645
Autoformer	3.2389	3.2790	1.8055	1.8830	2.3327	4.2420
FEDformer	2.3841	2.1214	3.7312	1.1557	2.8512	2.1066
QuantileFormer	**1.9902**	**1.8435**	5.0815	4.4471	**1.5858**	**0.8335**

Table 3: Time series probabilistic forecasting results evaluated by cpaw (the lower the better).

5 Experiments

Datasets: We evaluate the performance of our proposed model based on the following open datasets: (1) **Electricity**, (2) **ETT** [Zhou *et al.*, 2020], which contains four subsets **ETTm1**, **ETTm2**, and **ETTh1**, and **ETTh2**, (3) **Traffic**, (4) **Solar**, (5) **Wind**. Information are shown in Table 2.

Baselines We compare the proposed **QuantileFormer** with three state-of-the-art probabilistic forecasting models, which include **TemporalFusionTransformer (TFT)** [Lim *et al.*, 2019], **DeepAR** [Salinas *et al.*, 2020] and **MQRNN** [Wen *et al.*, 2017]. We also compare our model with other Transformer-based models, such as **PatchTST** [Nie *et al.*, 2022], **iTransformer** [Liu *et al.*, 2023], **Autofomer** [Wu *et al.*, 2021], **FeDformer** [Zhou *et al.*, 2022] and **Transformer** [Vaswani *et al.*, 2017]. Note that the later five Transformer-based models were designed for point-wise forecasting and we adapt them to quantile prediction by training them with the proposed quantile loss.

Performance Metrics: Previous works [Salinas *et al.*, 2020] widely used the *q-risk* to quantify the accuracy of a q-th quantile of the predictive distribution, which is defined as:

$$\textit{q-risk} = \frac{2\sum_{y_t\in\hat{\Omega}}\sum_{\tau=1}^{\tau_{max}}(q(y-\hat{y})_+ + (1-q)(\hat{y}-y)_+)}{\sum_{y_t\in\hat{\Omega}}\sum_{\tau=1}^{\tau_{max}}|y_t|}, \quad (20)$$

where $(\cdot)_+ = max(0,\cdot)$.

Since q-risk only considers the accuracy of quantiles, it is lack of consideration to measure the probabilistic interval (PI). To this end, we propose a new performance metric to measure how the true value interact with the predicted probabilistic interval. We combine the coverage probability with normalized averaged width to form the metric called *cpaw*, which is formulated by the PI coverage probability (PICP) and PI normalized averaged width (PINAW) as

$$cpaw = PINAW(1 + \gamma \cdot e^{-(PICP-\mu)}), \quad (21)$$

where $PICP = \frac{1}{n}\sum_{i=1}^{n} I(y_i \in [\hat{q}_{i,l}, \hat{q}_{i,u}])$ calculates the average probability of whether an observation is within the prediction interval of corresponding quantile across all samples; $PINAW = \frac{1}{n}|\hat{q}_{i,u} - \hat{q}_{i,l}|$ calculates the average percentage of the prediction interval width relative to the range

	Electricity			Wind		
	0.5	0.7	0.9	0.5	0.7	0.9
w/o D-D Decomp.	0.7629	0.8890	0.6738	1.0746	1.2476	1.7182
w/o GMM Decomp.	0.9890	0.9125	0.5570	0.9782	0.9575	0.4451
w/o Fusion Transformer.	0.9389	0.9104	0.9885	0.8954	0.8861	1.0460
QuantileFormer	**0.7546**	**0.3330**	**0.5121**	**0.8403**	**0.7346**	**0.3369**
	Solar			Traffic		
	0.5	0.7	0.9	0.5	0.7	0.9
w/o D-D Decomp.	1.3440	1.2463	0.6142	0.9626	1.3814	0.5497
w/o GMM Decomp.	1.0831	1.1991	0.7914	1.3995	0.8849	0.5837
w/o Fusion Transformer.	1.0708	1.1930	0.7289	1.5161	1.1245	0.8275
QuantileFormer	**1.0641**	**1.1832**	**0.5883**	**0.8489**	**0.8489**	**0.4688**

Table 4: Ablation on the model architecture (evaluated by q-risk).

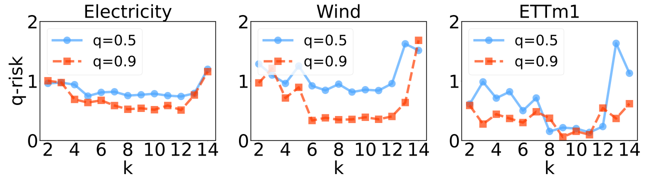
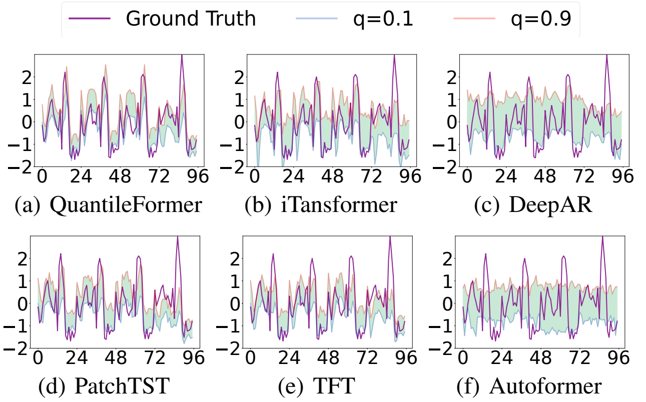


Figure 3: Hyperparameter analysis on Gaussian components k is performed to investigate its impact on q-risk for 0.5 and 0.9 quantile on Electricity, Wind and ETTm1 dataset.



(a) QuantileFormer (b) iTansformer (c) DeepAR

(d) PatchTST (e) TFT (f) Autoformer

Figure 4: Visualization of the probabilistic forecasting results of different models on the Electricity dataset. The dark lines stand for the ground truth and the light shadow stand for the predicted probabilistic intervals. The gray line represents the prediction upper bound, and the yellow line represents the prediction lower bound. We set the upper and lower bound quantile as 0.1 and 0.9.

of observations across all samples; μ represents the difference between the upper and lower quantiles; and γ is an indicative function reflecting whether the PICP exceeds μ.

5.1 Main Results

Comprehensive forecasting results evaluated by q-risk are listed in Table 1 with the best in bold. We compare the experimental results at various quantile levels. The results show that QuantileFormer achieves the best performance in most cases, with an average q-risk decrease of 24% for 0.5 quantile, decrease of 15% for 0.6 quantile, decrease of 27%, 14%, and 22% for 0.7, 0.8, and 0.9 quantile respectively, compared to the second-place algorithm.

Performance evaluated by cpaw is shown in Table 3. By analyzing the table, we make the following discussions and conclusions. 1) Compared with methods which are based on Transformer (i.e., TFT, Transformer, Autoformer, FeDformer, PatchTST and iTransformer), our method achieves 20% and 51% improvement on Wind and Traffic dataset, respectively. 2) Compared with methods which are based on RNN (i.e., DeepAR, MQRNN), our method improves by 55%, 50% and 88% on Electricity, Wind and Traffic datasets over other baselines, respectively.

5.2 Ablation Study

To explore the role of each module in our proposed framework, we compare the prediction results obtained by different sections as shown in Table 4. In the results, removing each component results in performance drop in different levels, showcasing the effectiveness of the proposed framework. We conduct experiments without the drift-divergence decomposition, the Gaussian mixture model decomposition and the fusion Transformer module respectively.

5.3 Hyperparameters Analysis

We analysis the impact of hyperparameters. i.e., selection of the number of Gaussian components k on the model's final performance, and the results are illustrated in Figure 3. According to the figure, if k is too small (e.g., $k \leq 4$), the performance is relative poor due to no enough Gaussian components to describe the mixture distribution. If k is too large (e.g., $k \geq 12$), the performance also degrade, probably due to overfitting. A suitable k is within [8,10] for Electricity, within [6,10] for Wind, and within [8,11] for ETTm1.

5.4 Visualization

We visualize the probabilistic forecasting results of different models as in Figure 4 (the Electricity dataset). These visualizations offer insights into how different models perform in capturing the underlying uncertainty and predictive trends within each respective dataset. It demonstrates that the QuantileFormer is more in line with the ground truth, with a much narrower probabilistic interval (PI) and a lower q-risk. This verify the effectiveness of the pattern-mixture decomposed Transformer model.

6 Conclusion

This paper introduced QuantileFormer, a novel Transformer-based model that revolutionizes probabilistic time series forecasting through a meticulous pattern-mixture decomposition approach. It decomposed complex time series data into quantile drift and divergence patterns, capturing the nuanced temporal dynamics and stochastic features. The quantile drift was captured by an encoder, while the divergence patterns were broken down into Gaussian mixture components. A Variational distribution inference network was introduced to extract the global statistical properties. These decomposed elements were then merged by a fusion Transformer, which synthesizes the information to produce accurate quantile predictions. Through extensive experimentation, the paper demonstrated the efficacy of the proposed model, confirming its effectiveness in probabilistic time series forecasting.

Acknowledgments

This work was partially supported by the Natural Science Foundation of Jiangsu Province (Grant No.BK20222003), the Collaborative Innovation Center of Novel Software Technology and Industrialization, and the Sino-German Institutes of Social Computing. The corresponding author is Wenzhong Li (lwz@nju.edu.cn).

References

[201, 2012] A tutorial on variational bayesian inference. *Artificial Intelligence Review*, 38:85–95, 2012.

[Bergmeir and Hyndman, 2015] Christoph Bergmeir and Rob J Hyndman. Probabilistic time series forecasting with boosted additive models. *International Journal of Forecasting*, 31(1):49–61, 2015.

[Bontempi and Ben Taieb, 1999] Gianluca Bontempi and Souhaib Ben Taieb. Modelling time series with neural networks: A comparative study. *Journal of Applied Sciences*, 25(9):1457–1469, 1999.

[Caldas and Soares, 2022] Francisco M Caldas and Cláudia Soares. A temporal fusion transformer for long-term explainable prediction of emergency department overcrowding. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 71–88. Springer, 2022.

[Cleveland *et al.*, 1990] Rb Cleveland, William S. Cleveland, Jean E. McRae, and Irma J. Terpenning. Stl: A seasonal-trend decomposition procedure based on loess (with discussion). 1990.

[Cote, 2016] Marco Cote. Stick-breaking variational autoencoders. *arXiv: Machine Learning*, 2016.

[De Jong, 1980] Piet De Jong. A seasonal-trend decomposition procedure based on loess. *Journal of Time Series Analysis*, 1(4):363–376, 1980.

[Dempster *et al.*, 1977] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the em - algorithm plus discussions on the paper. 1977.

[Duan *et al.*, 2023] Jian-Hui Duan, Wenzhong Li, Derun Zou, Ruichen Li, and Sanglu Lu. Federated learning with data-agnostic distribution fusion. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8074–8083, 2023.

[Eisenach *et al.*, 2020] Carson Eisenach, Yagna Patel, and Dhruv Madeka. Mqtransformer: Multi-horizon forecasts with context dependent and feedback-aware attention. *ArXiv*, abs/2009.14799, 2020.

[Griffiths and Ghahramani, 2011] Thomas Griffiths and Zoubin Ghahramani. The indian buffet process: An introduction and review. *Journal of Machine Learning Research*, 12:1185–1224, 04 2011.

[Hong *et al.*, 2024] Xiaobin Hong, Jiangyi Hu, Taishan Xu, Xiancheng Ren, Feng Wu, Xiangkai Ma, and Wenzhong Li. Magnet: Multilevel dynamic wavelet graph neural network for multivariate time series classification. *ACM Transactions on Knowledge Discovery from Data*, 19(1):1–22, 2024.

[Hong *et al.*, 2025] Xiaobin Hong, Jiawen Zhang, Wenzhong Li, Sanglu Lu, and Jia Li. Unify and anchor: A context-aware transformer for cross-domain time series forecasting. *arXiv preprint arXiv:2503.01157*, 2025.

[Huy *et al.*, 2022] Pham Canh Huy, Nguyen Quoc Minh, Nguyen Dang Tien, and Tao Thi Quynh Anh. Short-term electricity load forecasting based on temporal fusion transformer model. *IEEE Access*, 10:106296–106304, 2022.

[Hyndman and Athanasopoulos, 2018] Rob J Hyndman and George Athanasopoulos. Forecasting: principles and practice. *OTexts*, 1(2):1–456, 2018.

[Hyndman and Khandakar, 2008] Rob J Hyndman and Yeasmin Khandakar. Automatic time series forecasting: The forecast package for r. *Journal of Statistical Software*, 27(3):1–22, 2008.

[Ioffe and Szegedy, 2015] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ArXiv*, abs/1502.03167, 2015.

[Jiang *et al.*, 2024] Yue Jiang, Xiucheng Li, Yile Chen, Shuai Liu, Weilong Kong, Antonis F. Lentzakis, and Gao Cong. Sagdfn: A scalable adaptive graph diffusion forecasting network for multivariate time series forecasting. *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 1255–1268, 2024.

[Kingma, 2013] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[Li *et al.*, 2024] Yuxin Li, Wenchao Chen, Xinyue Hu, Bo Chen, baolin sun, and Mingyuan Zhou. Transformer-modulated diffusion models for probabilistic multivariate time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

[Lim *et al.*, 2019] Bryan Lim, Sercan Ö. Arik, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *ArXiv*, abs/1912.09363, 2019.

[Lin *et al.*, 2022] Mingkai Lin, Wenzhong Li, Ding Li, Yizhou Chen, and Sanglu Lu. Resource-efficient training for large graph convolutional networks with label-centric cumulative sampling. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 1170–1180, New York, NY, USA, 2022. Association for Computing Machinery.

[Liu *et al.*, 2022] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X. Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International Conference on Learning Representations*, 2022.

[Liu *et al.*, 2023] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long.

itransformer: Inverted transformers are effective for time series forecasting. *ArXiv*, abs/2310.06625, 2023.

[Ma *et al.*, 2024] Xiangkai Ma, Xiaobin Hong, Sanglu Lu, and Wenzhong Li. Ts3net: Triple decomposition with spectrum gradient for long-term time series analysis. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 887–900, 2024.

[McCullough and Renfro, 1990] B D McCullough and Charles G Renfro. Seasonal adjustment by signal extraction. *Journal of Business & Economic Statistics*, 8(1):27–36, 1990.

[Ng and Jordan, 2001] A. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Neural Information Processing Systems*, 2001.

[Nie *et al.*, 2022] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *ArXiv*, abs/2211.14730, 2022.

[Oreshkin *et al.*, 2019] Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *ArXiv*, abs/1905.10437, 2019.

[Rasul *et al.*, 2021] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, 2021.

[Romano *et al.*, 2019] Yaniv Romano, Evan Patterson, and Emmanuel J. Candès. Conformalized quantile regression. In *Neural Information Processing Systems*, 2019.

[Salinas *et al.*, 2018] Jorge Salinas, Velimir Čorić, and Pierre Pinson. A comparison of probabilistic methods for uncertainty quantification in short-term wind speed forecasting. *International Journal of Forecasting*, 34(4):785–798, 2018.

[Salinas *et al.*, 2020] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

[Sen *et al.*, 2019] Rajat Sen, Hsiang-Fu Yu, and Inderjit S. Dhillon. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. *ArXiv*, abs/1905.03806, 2019.

[Tukey, 1960] John W Tukey. Time series: a biostatistical introduction. *Time series: a biostatistical introduction*, 2, 1960.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, 2017.

[Wang *et al.*, 2021] Jianing Wang, Yifan Xiong, Jing Yang, and Xi Zhang. A hierarchical bayesian neural network framework for probabilistic time series forecasting. *Journal of Computational and Graphical Statistics*, 30(1):131–142, 2021.

[Wang *et al.*, 2024] Shiyu Wang, Haixu Wu, Xiao Long Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. *ArXiv*, abs/2405.14616, 2024.

[Wen *et al.*, 2017] Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. A multi-horizon quantile recurrent forecaster. *arXiv: Machine Learning*, 2017.

[Wu *et al.*, 2021] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 22419–22430. Curran Associates, Inc., 2021.

[Wu *et al.*, 2022] Haixu Wu, Teng Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *ArXiv*, abs/2210.02186, 2022.

[Zhang *et al.*, 2022] Hao Zhang, Yajie Zou, Xiaoxue Yang, and Hang Yang. A temporal fusion transformer for short-term freeway traffic speed multistep prediction. *Neurocomputing*, 500:329–340, 2022.

[Zheng *et al.*, 2023] Peijun Zheng, Heng Zhou, Jiang Liu, and Yosuke Nakanishi. Interpretable building energy consumption forecasting using spectral clustering algorithm and temporal fusion transformers architecture. *Applied Energy*, 349:121607, 2023.

[Zhou *et al.*, 2020] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wan Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI Conference on Artificial Intelligence*, 2020.

[Zhou *et al.*, 2022] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. *ArXiv*, abs/2201.12740, 2022.

[Zhou *et al.*, 2023] Yunyi Zhou, Zhixuan Chu, Yijia Ruan, Ge Jin, Yuchen Huang, and Sheng Li. ptse: A multi-model ensemble method for probabilistic time series forecasting. *ArXiv*, abs/2305.11304, 2023.