FQFormer: A Fully Quantile Transformer for Time Series Forecasting

Shayan Jawed
shayan@ismll.uni-hildesheim.de
Information Systems and Machine Learning Lab
University of Hildesheim
Hildesheim, Germany

Lars Schmidt-Thieme
schmidt-thieme@ismll.uni-hildesheim.de
Information Systems and Machine Learning Lab
University of Hildesheim
Hildesheim, Germany

ABSTRACT

We propose FQFormer, a novel method for quantile forecasting. We base our method in the Implicit Quantile Network (IQN) learning framework, where random samples from the $\mathcal{U}(0, 1)$ distribution are reparameterized to quantile values of the target distribution. However, in IQNs, each univariate quantile forecast estimation corresponding to an input quantile level can only be considered a marginal distribution and correlations among quantile estimations are only modeled in the shared network parameters. Whereas, probabilistic metrics are based on the joint distribution of (quantile) samples, for example, the CRPS metric is computed as the Integral over differencing the predictive CDF with the heavyside function based on the forecast ground truth. In this paper, we propose to learn optimal quantile levels conditioned on the input time series's history and IDs that replace random levels. To this end, we firstly train a Transformer model with the well-established IQN framework through randomly sampled quantile levels. In the second stage, we train another Transformer model based on the same input to output quantile levels that are fed to the first stage frozen model and updated by minimizing the empirical CRPS loss that encourages selection of quantile levels estimating the joint distribution optimally. We experimentally validate the superiority of our proposed two-stage method to state-of-the-art probabilistic forecasting baselines and ablations to the loss formulation.

KEYWORDS

Probabilistic Forecasting, Implicit Quantile Networks, Sparse Attention Transformer, Quantile Proposal Network

1 INTRODUCTION

Time series forecasting is an active area of research with significant applicability across various domains such as Energy Management, Urban planning, Retail business forecasting to name a few[14, 22]. Successful application of forecasting solutions to these domains additionally requires uncertainty quantification in the forecasts. As a result, many probabilistic forecasting approaches have been proposed in prior work. A substantial number of works have combined sequential modeling primitives with likelihood components, where parameters of a distribution specified apriori are linearly extrapolated from the sequential encoding of the time series input history. However, the task of specifying a distribution apriori across various data generating processes of different application domains can hinder the application of such methods. In this paper, we focus on Quantile regression [10], a well-understood statistical method that has been extensively researched for robustly modeling probabilistic outputs [2, 4] across several data generating processes and underlying distributions thereof.

However, several prior quantile forecasting methods [15, 27] require retraining with a different quantile loss parameterization to provide quantile estimations for quantile levels they are not trained with. On the other hand, Variational Autoencoder (VAEs)[7, 30], Generative Adversarial Networks (GANs) [11], Variational Flow [20] models that estimate the full probabilistic density could provide quantile estimates at any level. In this regard, we note two prominent streams, firstly, Multi-Quantile networks (MQ-RNN) [17, 26] that learn to output a discrete set of multiple quantile estimations corresponding to different levels jointly and with further post-processing interpolation between quantile levels solve the retraining issue to sufficient extent. Secondly, the approach of learning the full quantile function, as done in Implicit Quantile Networks (IQN)[2]. Merits of IQNs over GANs and VAE based methods include more stable optimization with piecewise linear quantile loss functions and arbitrary extension to other quantile levels.

However, both MQNs and IQNs only estimate univariate quantiles and do not model correlations between multiple quantile estimations directly, rather only via shared paramater spaces. On the other hand, the most widely reported metric, Continuous Ranked Probability Score (CRPS) is computed as the Integral over differencing the predictive CDF with the heavyside function based on the forecast horizon ground truth [5]. Hence, requiring that the joint distribution based on the samples is correctly estimated. This opens the question which quantile levels to select given a limited number of samples to estimate the empirical CRPS metric. In this paper, we propose to learn optimal quantile levels conditioned on the input time series's history and respective IDs by minimizing the empirical CRPS that leads to selection of quantile levels that estimate the joint distribution with limited samples more accurately.

We look into two distinct instantiations of the empirical approximation to the Integral based CRPS loss formulation. The first empirical instantation is based on the quantile loss formulation where a sum of piecewise linear quantile loss functions estimates

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

the CRPS. Several works have used this instantiation to approximate the CRPS as a test metric [4, 19, 20] and moreover learning on multiple quantile loss functions simultaneously as done in MQ-RNN and related approaches leads to learning on this approximated empirical CRPS loss. Additionally, we consider the Energy score based empirical CRPS loss instantiation [5] that models correlations between quantile estimations as well as requiring that individually each estimation is a sharp approximation to the ground truth, but collectively they are maximally spread apart to capture various underlying modes of the underlying forecast distribution.

Although the equivalence of these two instantiations has already been established, we propose learning on both the empirical approximations in order to cover inherent biases from both approximations to the integral. We propose a two-stage optimization framework, where firstly we train a Transformer method for quantile forecasting with the well-established IQN framework, and in the second stage we train another Transformer model that acts a proposal network and outputs quantile levels that are learned such that quantile estimates corresponding to these input levels from the first stage network can minimize the Energy score equivalent CRPS loss. It can be intuitively seen that an optimization over the selection of quantile levels can not be carried out with respect to the minimization of the discrete sum of quantile loss function approximation to the CRPS, since that is prone to degenerate optimization as the quantile loss functions are parameterized with the same quantile levels for which the corresponding estimations are made. To recap, our contributions are:

- We design a novel two-stage optimization framework to learn quantile levels by minimizing the empirical Energy score based CRPS loss that leads to an optimal estimate of the joint distribution of quantile samples.
- We perform extensive experiments to validate `FQFormer`'s performance compared to several probabilistic forecasting baselines on benchmark datasets.
- We provide an ablation study that validates the optimality of the two-stage optimization framework.

2 RELATED WORK

Probabilistic time series forecasting has been an active area of research, with many prior works being published, and a survey can be found in [1]. A series of works incorporate a likelihood component that outputs parameters for a distribution specified apriori [14, 18, 22]. `DeepAR` [22] is a probabilistic forecasting model that recursively unrolls the hidden state for each time step and a linear layer extrapolates from the hidden state to Gaussian likelihood parameters (μ, σ) for each time step autoregressively. We also note, Deep State Space Model (`DeepState`) [18] that forecasts through a linear Gaussian state-space model whose state and transition parameters are estimated via an underlying RNN component. Another prominent method, `LogTrans` is a Convolutional sparse Attention [14] based decoder only transformer that autoregressively decodes future forecasts one-step a time. Fundamentally, it is similar to prior listed works [22] where a linear layer extrapolates from the Attention representations to estimate Gaussian parameters.

Several probabilistic forecasting approaches are based on Variational inference as well. Conditional VAE (`CVAE`) [30] maps past trajectory and side information to latent codes, and a decoder maps these latent codes to future trajectory estimations. Interestingly, in [30] Determinantal Point Processes (DPP) are also used to sampled diverse trajectories, which shares the motivation with the Energy score CRPS definition taking the distance of the samples among each other into account. `STRIPE` [13] also uses a conditional VAE backbone and introduced new DPP processes that explicitly optimize for diverse future estimations based on time dilation and shape warping factors.

We also note a GAN based forecasting model in [11], which is designed as a single step probabilistic forecasting model with Generator and Discriminator networks being composed of RNNs. Adversarial Sparse Transformer (`AST`) [27] is a Sparse Attention model that autoregressively decodes for a single fixed quantile level. In order to mitigate the known error accumulation problem through single step decoding, the model is trained with the adversarial framework where a fully connected network subcomponent discriminates between generated and full forecast output. Orthogonally related is the work from [29] where interestingly a GAN framework is utilized for probabilistic estimation of time series data to sample augmentation data for downstream applications.

Notably, various probabilistic forecasting methods have been proposed for multivariate forecasting [19–21]. In [20], a sequential Transformer model is unrolled over multivariate time series data and a series of Invertible transformations are applied to derive a Normalizing Flow based density estimation of the multivariate observations. We also note, a Gaussian Copula based model [21] that models the joint multivariate distribution of high dimensional time series. To reduce the computational complexity, the authors do not model the entire covariance structure but randomly sample dimensions that are fed to a shared RNN to output the low-rank covariance matrix parameters. Lastly, Autoregressive Diffusion models have also been proposed for multivariate forecasting, where, following previous works, an RNN unrolls providing a time-indexed hidden state representation on to which a forward-backwards diffusion process is executed to arrive at a probabilistic estimation of the observation [19].

Early on [26] combined RNNs with quantile regression to directly estimate multiple quantiles for multiple future horizons jointly. This work was followed by [3] where the base sequential modeling was additionally extended with an Attention mechanism. Additionally, the focus was on increasing modeling capacity for event indicators and deriving rich Attention based interactions for same target forecasting horizons but modeled through variable sized history lengths. The model from [15] also uses a Transformer backbone trained with quantile loss functions for robust estimates of the 50^{th} and 90^{th} quantile outputs. The `SQF-RNN` model [4] is a distribution-free quantile regression method that models the quantile function of the forecast distribution through a monotonic spline based representation. Notably, the model is trained with an analytic CRPS loss formulation based on the spline-representation and pre-specified quantile levels. Quantile forecasting has already been explored with implicitly embedding the quantile levels [6, 25]. We note both works, `IQN-RNN` [6] where an RNN based encoding of the time series history is combined with the quantile level embeddeding and

the model parameters are optimized with the quantile loss parameterized with the same quantile levels. In the other work, [25], an RNN based encoder was combined with a Gaussian Copula that modeled the latent correlations between the marginal quantile estimations. Notably, quantile estimations from prior approaches [2, 3, 15, 26, 27] can be prone to *quantile crosssing* phenomena due to the lack of structural regularization and constraints in the output space and restricted capability to model correlations between quantile estimations only in the shared parameter spaces. The method in [17] solves quantile crossing by structuring successive quantile outputs corresponding to larger quantile level inputs to add on to preceeding ones and all outputs being constrained to be positive. Additionally, an analytic CRPS loss formulation was derived similar to `SQF-RNN` based on learnable spline components. Also related is the method in [9], where a multivariate quantile function based forecasting component is uniquely designed to fulfill monotonicity constraints with regard to the randomly sampled quantile levels. Interestingly, the Energy score based CRPS loss was also used to train the multivariate quantile estimations.

In summary, several works have been proposed for quantile forecasting, and recently implicit quantile modeling has also been explored for forecasting. Our work closely follows the same intuition as works in [4, 9, 17] that model explicit correlations between multiple quantile estimations. However, with `FQFormer` , we propose to learn quantile levels and validate that optimally choosing a limited set of quantile levels can improve modeling the joint distribution for the forecast horizons.

3 BACKGROUND

3.1 Problem Formulation

We consider N related univariate time series data $\mathbf{Y} \in \mathbb{R}^{T\times N}$ where each time series $Y^n \in \mathbb{R}^T$ is noted for a total of $t = [1, ..\tau, ..., T]$ timesteps[1] and τ partitions the observations in the input range and the multiple horizon indexes for forecasting. Additionally, we consider C many social time[2]covariates $X \in \mathbb{R}^{T\times C}$ that are observed in the entire range[3]. Our objective is to model the following conditional distribution:

$$p(Y^n_{\tau+1:T}|Y^n_{1:\tau}, X_{1:T}, \Theta) \tag{1}$$

This formulation in Eq. 1 explicitly models for multiple tasks jointly conditioned on the same input and model parameters Θ.

3.2 Quantile Regression

In many real-world applications, a distribution of the future is required instead of a single point estimate. Hence, we consider modeling the cumulative distribution (CDF) of the random variable $Y^n_{\tau+1} \in \mathbb{R}$ following prior work [4, 10, 17]. Let us denote the CDF by $F_Y(y)$, then the quantile estimate at quantile level $\alpha \in (0, 1)$ is:

$$Q_Y(\alpha) := F_Y^{-1}(\alpha) = \inf\{y \in \mathbb{R} : \alpha \le F_Y(y)\} \tag{2}$$

Where the function, Q_Y is called the quantile function or equivalently the inverse CDF function. Intuitively, $\alpha \in (0, 1)$ is the probability that Y is less than $Q_Y(\alpha)$. We can write the α quantile estimate as follows:

$$q^n_{\alpha,\tau+1} = Q_Y(\alpha|Y_{1:\tau}, X_{1:T}, \Theta) \tag{3}$$

We can model the α quantile estimate by minimizing expected quantile loss[4],

$$\arg\min_{\Theta\in\mathbb{R}} \mathbb{E}_{Y\sim F_Y} \rho_\alpha(Y, q_\alpha) \tag{4}$$

The loss function, $\rho_\alpha(Y, q_\alpha)$ can be expressed as follows:

$$\begin{aligned}\rho_\alpha(Y, q_\alpha) &= (Y - q_\alpha)(\alpha - \mathbb{I}_{(Y\le q_\alpha)}) \\ &= \begin{cases} \alpha(Y - q_\alpha), & \text{if } Y \ge q_\alpha, \\ (\alpha - 1)(Y - q_\alpha), & \text{if } Y < q_\alpha, \end{cases}\end{aligned} \tag{5}$$

Intuitively, the quantile loss is the asymmetric generalization of the mean absolute error. When $\alpha_i = 0.5$ we can estimate the median, otherwise, if $\alpha_i \to 0$ the loss penalizes more for overestimation and less for underestimation. Conversely, when $\alpha_i \to 1$ the model is penalized for underestimation significantly more than for overestimation. Choosing a sufficiently large and diverse set of $\alpha_{1:M} \in \mathcal{U}(0, 1)$ can lead to modeling a corresponding conditional quantile distribution. Notably, Quantile regression offers the flexibility to model data without any arbitrary assumptions on the data generating processes such as assuming Gaussianity and has shown to be an effective choice for modeling across a large class of data generating distributions in practice [6, 17].

3.3 Continuous Ranked Probability Score

We now extend the discussion towards *proper scoring* metrics for evaluating probabilistic forecast distributions. The CRPS is a proper scoring metric for evaluating probabilistic forecast distributions [5]. A proven proper scoring metric entails the important result that the score is the least when predicted distribution is equivalent to data distribution [4, 5, 16] [5]. The CRPS metric can be computed as:

$$\text{CRPS}(F_Y(y), Y) = \int_{\mathbb{R}} (F_Y(y) - \mathbb{I}\{Y \le y\})^2 dy \tag{6}$$

Where $\mathbb{I}\{Y \le y\}$ denotes the indicator function [5]. Analytic forms of the integral in Eq.6 for popular distributions such as Gaussian and Negative-binomial likelihood also exist [5, 8] but for various real-world applications such assumptions can be overly simplistic and limit modeling capability. Besides, motivated by the simple quantile regression formulation, prior works [6] have utilized an equivalent CRPS formulation for learning based on quantiles [12, 23][6]:

$$\text{CRPS}(F_Y(y), Y) = 2\int_0^1 \rho_\alpha(Y, q_\alpha) d\alpha \approx \sum_{i=1}^{M} \rho_{\alpha_i}(Y, q_{\alpha_i}) \tag{7}$$

On the other hand, in this paper we also consider another known sampled approximation of the CRPS metric based on the Energy score [5][9]. However, we uniquely model this approximation via M many *quantile estimates*,

$$\text{CRPS}(\hat{F}_Y(y), Y) = \frac{1}{M}\sum_{i=1}^{M} \left|q_{\alpha_i} - Y\right| - \frac{1}{2M^2}\sum_{i=1}^{M}\sum_{j=1}^{M} \left|q_{\alpha_i} - q_{\alpha_j}\right| \tag{8}$$

[1] t is relative, can correspond to different time across time series
[2] time-of-the-day, week-of-the-month etc
[3] We transform covariates from the natural domain to real domain via normalization
[4] In following, we simplify the notation, by dropping the indexes n and τ since the same loss is applied for all time series and horizons
[5] Proper scoring metrics are negatively oriented, lower scores are better
[6] A proof of equivalence is given in these works

4 METHOD

In this section, we provide a compartmentalized description of our model, `FQFormer` for time-series forecasting. The model borrows ideas from the recent work on Quantile conditional modeling [2], Sparse attention mechanisms [14], Quantile Proposal Networks [28] and multi-task learning [26].

4.1 Sparse Attention Encoder

We now describe the encoding of the time series observations. Sequential encoding primitives such as Convolutions and Recursive hidden states have been extensively researched for forecasting objectives. However, networks based on these primitives are unable to model long-range interactions prevalent in time-series. Convolutional receptive fields can become bottlenecks hindering learning of long-range interactions in successive layers and Recurrent neural networks suffer from catastrophic gradient vanishing for initial hidden states. In light of these challenges, recently, Transformer networks [14, 27, 31] have been proposed for time series forecasting. Transformer networks compute pairwise attention between inputs at all timesteps. This enhances modeling long-range relations since input at each timestep can attend to all other input timesteps regardless of the distance in-between. As a potential downside, this raises the complexity to $O(T^2)$. Hence, we adopt the Sparse Attention model proposed in *Log Sparse Transformer* [14] as our base encoding component. The sparse attention mechanism calculates only $O(logT)$ dot products for each timestep in each layer, effectively allowing each timestep representation only to attend to previous observations representations with an exponential step size and itself. Hence, the complexity is reduced from $O(T^2)$ to $O(T \log T)$. We donate this attention computation scheme as LogAttention and compute encoding as:

$$\xi_{\tilde{y}} = [Y_{1:T} \dashv\vdash X_{1:T} \dashv\vdash (\xi_{pos} + \xi_{ID})] \tag{9}$$

$$\xi_{\tilde{y}} = \text{LogAttention}(\xi_{\tilde{y}}, \xi_{\tilde{y}}, \xi_{\tilde{y}}) \tag{10}$$

Where, $\dashv\vdash$ operator denotes concatenation. It is important to note that we state the computational complexity and the encoding with respect to the entire range of time series T since we utilize the known future covariate values for the forecast range. Practically, we fill the unknown future horizons $t = [\tau + 1, ..., T]$ with 0s for $Y_{\tau+1:T}$ for concatenation on the time axis similar to [31]. Moreover, we also learn distinct ID embeddings corresponding to various time series that can help learn richer latent representations [14, 22]. These ξ_{ID} are repeated along the time axis and added to learned positional embeddings ξ_{pos}. For completeness, ξ_{ID} and ξ_{pos} are embedded with latent dimensionality d_{model}.

4.2 Implicit Quantile Embedding

The Implicit quantile embedding forms the second part of the Encoding stage. We sample multiple $\alpha_{1:M} \in \mathcal{U}(0, 1)$ levels randomly in the first stage optimization or later as we see proposed by the Quantile Proposed Network and embed these with an Attention based embedding component as follows,

$$\xi_{\alpha} = \text{ReLU}(\alpha W_{\alpha} + b_{\alpha}) \tag{11}$$

$$\xi_{\alpha} = \text{Attention}(\xi_{\alpha}, \xi_{\alpha}, \xi_{\alpha}) \tag{12}$$

The parameters $W_{\alpha} \in \mathbb{R}^{M \times d_{model}}$ and $b_{\alpha} \in \mathbb{R}^{d_{model}}$ indicate a ReLU activated feed-forward layer applied to each quantile level in a position-wise manner and can be thought of as 1d convolution with a kernel size 1 [24]. Next, we use an Attention based embedding to learn a powerful latent representation of the quantile levels. It is worth noting that we sample multiple $\alpha_{1:M}$ values per mini-batch, and structure the decoding for quantile estimations corresponding to these levels per input series in the mini-batch. Moreover, the same M many quantile estimations are made across the forecast horizons. This is much more efficient than sampling multiple quantile levels differently for each time series and forecast horizon.

4.3 Decoder

The decoder combines the quantile level encodings and the time series encoding in an efficient manner exploiting shared parameters for multiple quantile forecasts.

$$\xi^{Flat}_{\tilde{y}} = \text{Flatten}(\xi_{\tilde{y}}) \tag{13}$$

$$\xi_{\tilde{y},1:M} = \text{Repeat}(\xi^{Flat}_{\tilde{y}}, M) \tag{14}$$

$$q_{\alpha_i,\tau+1:} = [\xi_{\tilde{y},i} \dashv\vdash \xi_{\alpha_i}]\, W_{MTL} + b_{MTL} \quad \forall i = [1, ..., M] \tag{15}$$

In the above equations, we first flatten the encodings of the time series to one feature axis ($d_{model} \times \text{len}(1:\tau)$), and repeat these M many times to combine these with the quantile embeddings in Eq. 11. Finally, a shared fully connected layer, given by parameters $W_{MTL} \in \mathbb{R}^{(d_{model} \times \text{len}(1:\tau)) \times \text{len}(\tau+1:T)}$,$b_{MTL} \in \mathbb{R}^{\text{len}(\tau+1:T)}$ is learned to produce a quantile forecast based on the concatenated repeated representation of the time series and the embeddings of the implicit quantile levels.

4.4 Quantile Proposal Network

The Proposal network is inspired by the work in [28], with the goal of learning optimal quantile levels. It is structured similar to the Encoder-Decoder components as we have introduced in the preceding subsections. The aim is to condition the quantile level outputs on time series values in the input range, their resp. IDs, and all covariate information. Therefore, in the second stage optimization, we initialize a new Encoder-Decoder and task it to predict M many quantile levels:

$$\xi'^{Flat}_{\tilde{y}} = \text{Flatten}(\xi'_{\tilde{y}}) \tag{16}$$

$$\alpha'_{1:M} = \text{Sigmoid}(\xi'^{Flat}_{\tilde{y}}\, W_{Prop} + b_{Prop}) \tag{17}$$

Where W_{Prop} and b_{Prop} are linear layer parameters of the same input dimensionality as W_{MTL} and b_{MTL}, however predicting the M many quantile levels. Lastly, we also differentiate the learned quantile levels as α', the structurally similar new second-stage encoding ξ', and the set of collective parameters of the Quantile Proposal network as Θ'.

4.5 Two-Stage Optimization

We propose a two-stage optimization process where the Encoder-Decoder as introduced above are first optimized to output quantile estimations corresponding to input randomly sampled quantile levels. We use the sum of quantile loss functions approximating the

CRPS loss to train this first stage network,

$$\arg\min_{\Theta \in \mathbb{R}} \sum_{n=1}^{N} \sum_{t=\tau+1}^{T} \sum_{i=1}^{M} \rho_{\alpha_i}(Y_t^n, q_{\alpha_i,t}^n) \tag{18}$$

Notably, we use the same quantile loss parameterized by quantile level α_i for all time series and forecast horizons considered in the mini-batch. On the other hand, we minimize the empirical CRPS loss based on the Energy score to estimate the parameters of the Quantile Proposal Network,

$$\arg\min_{\Theta' \in \mathbb{R}} \sum_{n=1}^{N} \sum_{t=\tau+1}^{T} \left(\frac{1}{M} \sum_{i=1}^{M} \left| q_{\alpha'_i,t}^n - Y_t^n \right| - \frac{1}{2M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \left| q_{\alpha'_i,t}^n - q_{\alpha'_j,t}^n \right| \right) \tag{19}$$

The above equation is differentiated from Eq. 18 with regard to the learned quantile levels α'. As we noted before, although the equivalence of these two instantiations has already been established, learning on both the empirical approximations stands to cover inherent biases from both approximations to the integral. From a practical standpoint, once the Encoder-Decoder are trained to estimate the quantile function of the forecast distribution, we move to optimize the quantile levels in the second stage with the energy score based CRPS loss (Eq. 19) that models the correlations among the quantile estimations directly as we can see with the pairwise component of the loss. The pairwise component of the loss intuitively maximizes the distance between the quantile estimations to capture various modes of the underlying data distribution. Hence, by learning and selecting quantile levels, and subsequent estimations based on these that serve this criteria and are individually sharper quantile estimations nevertheless, as optimized for through the first component of the loss, we aim to optimally estimate the joint distribution of the quantile estimations.

It is worth reiterating that the quantile loss based formulation cannot be used to optimize the quantile level selection. Moreover, directly learning on the energy score loss is also not feasible given the computational complexity with regard to the pairwise distances and considering the competition among the two sub loss components requiring sharp estimations while simultaneously being spread apart. Therefore, we first propose to take advantage of the well-established IQN framework and optimize for the quantile estimations by randomly sampling quantile levels and then freeze the parameters of the first network, when training the proposal network on the energy score based CRPS loss in the second stage.

5 EXPERIMENTS[7]

The experiments are based on 2 real-world datasets on 4 different forecasting tasks. We follow the experimental protocol from previous works [14, 22].

5.1 Dataset Statistics

1) `electricity` is an hourly Kilo Watts time series dataset of electricity consumption of 370 households from January 2011 to end of 2014

[7] github.com/super-shayan/fqformer; Baseline implementation details are given in the Appendix

	`electricity`	`traffic`
# time series, N	370	963
time granularity	hourly	hourly
domain	$\mathbb{R}^+$	$[0, 1]$
# training examples	500K	500K
# input length, $[1...\tau]$	168	168
# forecasting length, $[\tau, ..., T]$	24/168	24/168

Table 1: Time series dataset statistics including the number of training windows generated through the sliding window procedure

2) `traffic` dataset notes the hourly occupancy rates in the range$[0, 1)$ of 963 car lanes across different freeways in San Francisco during the first half of year 2008.

Following the preprocessing described in [14, 22] we exploit the sliding window procedure to generate training samples from the datasets. Multiple windows are sampled for each dataset with fixed conditioning and forecasting length by sampling τ several times across the full time series ranges as noted in Table. 1. We also utilize the same weight sampling and scaling techniques as described in [14, 22] in order to consistently compare the results on the benchmark from [14].

5.2 Proposed Model and Ablations

We now describe the ablations to the proposed model, that validate the effect of the loss components and two-stage optimization.

1) `FQFormer-BASE` is the base version of `FQFormer` that is trained with only the quantile loss functions and randomly sampling the quantile levels following the IQN framework.
2) `A-FIX` is the Transformer based approach for *fixed* quantile modeling; it generates 99 discrete quantile level estimations with multi-task decoding. This model is also trained with quantile loss functions.
3) `A-DIR` is the same Transformer method as in `FQFormer-BASE` , however it is optimized directly with Energy score based CRPS loss without any Quantile loss functions.
4) `A-PRE-DIR` pretrains `FQFormer-BASE` with the quantile loss and in the second stage the model is optimized again with only Energy score CRPS loss for the same number of epochs.
5) `FQFormer` is the proposed model which uses the pretrained `FQFormer` model from the first stage and optimizes the quantile proposal network in the second stage to minimize the Energy score based CRPS loss.

5.3 Results

We report the comparison of our proposed `FQFormer` with several baselines in Tables. 2,3. We did extensive hyperparameter search for all baselines, ablations and proposed methods separately for each task with fair computational budgets. Each hyperparameter configuration was run for the first 3 integer seeds, and we report the test performance corresponding to the least validation error across configs on held-out 10% validation set windows sampled randomly in the training range [14]. Whereas, the test range corresponds to the last 7 days. The evaluation metrics correspond to loss functions

Table 2: Comparison to forecasting baselines in terms of two QL metrics. Results are formatted with $\cdot 10^2$, columnar least is boldfaced, second-least is underlined. Results in brackets denote reimplementation results

		electricity$_{24}$		electricity$_{168}$		traffic$_{24}$		traffic$_{168}$	
		$\mathbf{QL_{0.5}}$	$\mathbf{QL_{0.9}}$	$\mathbf{QL_{0.5}}$	$\mathbf{QL_{0.9}}$	$\mathbf{QL_{0.5}}$	$\mathbf{QL_{0.9}}$	$\mathbf{QL_{0.5}}$	$\mathbf{QL_{0.9}}$
Reported in [14]	ARIMA	15.4	10.2	28.3	10.9	22.3	13.7	49.2	28
	ETS	10.1	7.7	12.1	10.1	23.6	14.8	50.9	52.9
	TRMF	8.4	–	8.7	–	18.6	–	20.2	–
	DeepState	8.3	5.6	8.5	5.2	16.7	11.3	16.8	11.4
	DeepAR	7.5(6.198)	4.00(5.448)	8.2(8.264)	5.3(6.554)	16.1(12.041)	9.9(9.697)	17.9(15.657)	10.5(12.301)
	LogTrans	5.9(**5.781**)	3.4(**2.972**)	**7**(7.614)	4.4(3.845)	12.2(12.27)	8.1(7.891)	13.9(14.014)	9.4/(**8.567**)
VAE	CVAE MSE	6.693	6.622	8.137	6.494	13.268	11.661	15.244	12.329
	CVAE DIL	7.110	5.584	16.424	20.061	38.266	33.855	17.544	17.668
	STRIPE DIL	12.141	8.978	13.046	10.455	38.471	32.648	28.000	53.498
Quantile	AST	7.380	4.636	8.887	5.726	20.534	14.047	38.816	19.545
	MQ-RNN	7.572	3.847	9.145	4.726	11.662	8.452	14.499	10.400
	SQF-RNN	6.952	5.098	7.977	4.982	11.792	9.603	15.206	10.521
	IQN-RNN	6.429	4.652	8.419	6.813	12.155	8.984	14.940	9.823
Prop	FQFormer	6.418	3.133	7.88	**3.729**	11.023	7.890	13.255	9.353
	FQFormer-BASE	6.315	3.133	7.876	3.745	**10.756**	**7.866**	13.758	9.676
Ablat.	A-FIX	6.252	3.344	7.359	4.192	12.168	10.226	14.959	12.725
	A-DIR	7.121	5.472	7.793	6.762	11.687	13.459	**12.578**	13.366
	A-PRE-DIR	6.798	3.325	8.712	4.057	11.301	8.060	14.335	9.435

introduced before, and exact definitions of metrics used are provided in the Appendix B. Q-AVG refers to the averaged quantile loss (Eq. 7) over 99 quantile estimations on the discrete grid $\alpha_{1:M} = [0.01, 0.02, ..., 0.99]$, and E-CRPS corresponds to the loss stated in Eq. (8) estimated through the same quantile estimation as in Q-AVG with the exception of FQFormer results, given for it the predicted quantile levels are used instead.

A number of interesting observations can be drawn from these results. We first discuss the results corresponding to the 50^{th} and the 90^{th} quantile loss functions, $QL_{0.5}$ and $QL_{0.9}$ resp. We can see that the wins across tasks are unevenly distributed and FQFormer performs better on the long-range forecasting relative to other baselines.

The LogTrans baseline appears to be the strongest contender, showcasing the importance of Attention mechanism for forecasting tasks relative to the predominant RNN based models. We hypothesize that autoregressive decoding combined with the extensive prior literature on modeling Electricity dataset with Gaussian likelihood is the reason for LogTrans outperforming other baselines on the electricity$_{24}$ task. On the other hand, we can note the comparisons with regard to ablations and other FQFormer variants. Given that these share the same underlying modeling primitives as the FQFormer the performances are more or less in the similar range.

We now move to discuss the results for the CRPS metrics, which are more representative probabilistic metrics for judging the performance of the forecasts. It is, specifically here, that we see wins for the FQFormer models against the ablations. Notably, when the hyperparameters are tuned fairly for ablations, it is possible to see one time lifts for ablations such as we see for A-FIX on the $QL_{0.5}$ metric for the electricity$_{168}$ task however on the more representative probabilistic metrics the ablation model is unable to outperform the FQFormer on this task. Similarly, comparing A-DIR and FQFormer for the traffic$_{168}$ task also supports this argument.

Another set of observations can be derived from benchmarking the performance of the Gaussian likelihood based DeepAR and quantile methods MQ-RNN , IQN-RNN and SQF-RNN . Given how these methods are based on RNNs, the underlying model complexity is comparable, however, we see that the quantile methods are able to outperform DeepAR showcasing the importance of the quantile modeling. Additionally, we can observe that IQN-RNN is able to outperform MQ-RNN with significant margins on the electricity forecasting datasets, and coming close in the other 2. It is therefore plausible that learning on several quantile levels provides additional auxiliary regularization that leads to a generalization effect on the test set. CVAE MSE and CVAE DIL [13, 30] models also approximate the true forecast distribution with the Gaussian distribution through variational learning and use less powerful RNNs as the base sequential encoders and decoders, which might be the reason for lack of competitive results. On the other hand, we can note that CVAE DIL does not outperform the mean squared counterpart variational model. The poor performance from CVAE DIL also affects the STRIPE DIL model performance. The reason is that STRIPE DIL is a two-stage method that uses a pretrained CVAE DIL model. We also found that the two stage optimization of the STRIPE DIL model is prone to unstable optimization despite using available official code. Similarly, AST involves the notable difficulties in training GANs, and we find that the baseline performs poorly for the traffic forecasting tasks on the $QL_{0.5}$ and $QL_{0.9}$ metrics. Moreover, retraining it independently for each quantile level to report the CRPS metrics is computationally intensive, especially considering hyperparameter configurations for each quantile level. Notably, despite how quantile loss for the 90^{th} quantile estimation is weighted less in the loss formulation than the 50^{th} quantile, and

Table 3: Comparison to forecasting baselines in terms of averaged quantile loss metric and the Energy score based CRPS metric. AST requires retraining per quantile level, so we do not report CRPS for it. Results are formatted like before. Noatbly, E-CRPS metric is computed with predicted quantile levels for the FQFormer

		electricity$_{24}$		electricity$_{168}$		traffic$_{24}$		traffic$_{168}$	
		Q-AVG	**E-CRPS**	**Q-AVG**	**E-CRPS**	**Q-AVG**	**E-CRPS**	**Q-AVG**	**E-CRPS**
Gaus	DeepAR	5.704	5.680	7.836	7.814	9.606	9.525	14.107	14.026
	LogTrans	**4.388**	**4.342**	6.168	6.107	9.254	9.173	10.719	10.604
VAE	CVAE MSE	6.588	6.578	8.014	8.005	12.688	12.65	14.847	14.829
	CVAE DIL	6.982	6.974	16.069	16.044	37.980	37.960	17.093	17.064
	STRIPE DIL	11.629	11.618	12.628	12.575	36.605	36.540	24.372	24.243
Quantile	AST	–	–	–	–	–	–	–	–
	MQ-RNN	5.927	5.859	7.542	7.474	9.417	9.320	12.112	11.989
	SQF-RNN	5.772	5.719	6.705	6.648	10.083	9.999	12.128	12.008
	IQN-RNN	5.411	5.370	6.798	6.741	9.971	9.872	12.157	12.039
Prop	FQFormer	4.853	4.8*	**5.952**	**5.888***	8.506	8.425*	**10.227**	**10.138***
	FQFormer-BASE	4.770	4.739	**5.952**	5.913	**8.327**	**8.267**	10.574	10.475
Ablat.	A-FIX	4.981	4.955	5.974	5.947	10.432	10.393	12.953	12.911
	A-DIR	7.121	7.121	7.793	7.793	11.687	11.687	12.578	12.578
	A-PRE-DIR	5.230	5.201	6.838	6.801	8.833	8.774	12.730	10.991

is generally an easier objective to learn, we see that some baselines are unable to provide spread apart quantiles and as a result the error metrics for the $50th$ quantile are lesser in magnitude.

We can also compare the performance of FQFormer to ablations dealing with architectural and optimization alternatives. Firstly, we can see that FQFormer is able to outperform FQFormer-BASE on the long-range forecasting task. However, the FQFormer-BASE performs better on the short-range rolling forecasting task. This showcases the importance of the quantile level selection through the two-stage optimization is most beneficial for the long-range forecasting. This is also intuitive in terms of model complexity given that for the rolling forecasting tasks, the multi-task estimation covers 24 forecast horizons vs. 168 in the direct forecasting scenario, which makes the latter a more complex task where additional parameters and quantile level optimization through the Energy score CRPS loss is more helpful. On the other hand, comparing FQFormer to the A-FIX baseline showcases the important result of learning the quantile function implicitly compared to fixed discrete quantile levels. Moreover, regarding optimization, comparing FQFormer to A-DIR we can see that optimization alone through the Energy score based CRPS loss is suboptimal compared to the proposed two-stage optimization. Additionally, we explored sequential two-stage optimization in A-PRE-DIR , in order to ensure that the performance gain is not observed through additional training epochs as required in the proposed two-stage optimization, however as we observe selection of quantile levels outperforms this ablation as well.

6 CONCLUSION

In this work, we developed a quantile proposal network that outputs learned quantile levels conditioned on input time series's history and context features. We trained it with a novel optimization process by minimizing the Energy score based CRPS loss. We showcased that learning quantile levels and modeling correlations between the estimations based on these levels leads to an optimal estimate of the forecast distribution with limited samples, as validated through extensive empirical evaluation on real-world datasets. In future work, we shall research multivariate forecasting extensions that model correlations among quantile functions of covariate channels.

ACKNOWLEDGEMENTS

This work is co-funded by the industry project Data-driven Mobility Services of ISMLL and Volkswagen Financial Services. We also acknowledge funding by the Federal Ministry for Economic Affairs and Climate Action (BMWK), Germany, within the framework of the IIP-Ecosphere project.

REFERENCES

[1] Konstantinos Benidis, Syama Sundar Rangapuram, Valentin Flunkert, Bernie Wang, Danielle Maddix, Caner Turkmen, Jan Gasthaus, Michael Bohlke-Schneider, David Salinas, Lorenzo Stella, et al. 2020. Neural forecasting: Introduction and literature overview. *arXiv preprint arXiv:2004.10240* (2020).

[2] Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. 2018. Implicit quantile networks for distributional reinforcement learning. In *International conference on machine learning*. PMLR, 1096–1105.

[3] Carson Eisenach, Yagna Patel, and Dhruv Madeka. 2020. Mqtransformer: Multi-horizon forecasts with context dependent and feedback-aware attention. *arXiv preprint* (2020).

[4] Jan Gasthaus, Konstantinos Benidis, Yuyang Wang, Syama Sundar Rangapuram, David Salinas, Valentin Flunkert, and Tim Januschowski. 2019. Probabilistic forecasting with spline quantile function RNNs. In *International conference on Artificial Intelligence and Statistics*. PMLR.

[5] Tilmann Gneiting and Adrian E Raftery. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association* 102, 477 (2007), 359–378.

[6] Adele Gouttes, Kashif Rasul, Mateusz Koren, Johannes Stephan, and Tofigh Naghibi. 2021. Probabilistic Time Series Forecasting with Implicit Quantile Networks. *arXiv* (2021).

[7] Vincent Le Guen and Nicolas Thome. 2020. Probabilistic time series forecasting with structured shape and temporal diversity. *arXiv preprint arXiv:2010.07349* (2020).

[8] Alexander Jordan, Fabian Krüger, and Sebastian Lerch. 2017. Evaluating probabilistic forecasts with scoringRules. *arXiv preprint arXiv:1709.04743* (2017).

[9] Kelvin Kan, François-Xavier Aubet, Tim Januschowski, Youngsuk Park, Konstantinos Benidis, Lars Ruthotto, and Jan Gasthaus. 2022. Multivariate Quantile Function Forecaster. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 10603–10621.

[10] Roger Koenker and Kevin F Hallock. 2001. Quantile regression. *Journal of economic perspectives* 15, 4 (2001), 143–156.
[11] Alireza Koochali, Andreas Dengel, and Sheraz Ahmed. 2021. If You Like It, GAN It—Probabilistic Multivariate Times Series Forecast with GAN. In *Engineering Proceedings*.
[12] Francesco Laio and Stefania Tamea. 2007. Verification tools for probabilistic forecasts of continuous hydrological variables. *Hydrology and Earth System Sciences* (2007).
[13] Vincent Le Guen and Nicolas Thome. 2020. Probabilistic time series forecasting with shape and temporal diversity. *Advances in Neural Information Processing Systems* 33 (2020).
[14] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. 2019. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in Neural Information Processing Systems (NeurIPS)* (2019).
[15] Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. [n. d.]. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting* ([n. d.]).
[16] James E Matheson and Robert L Winkler. 1976. Scoring rules for continuous probability distributions. *Management science* 22, 10 (1976), 1087–1096.
[17] Youngsuk Park, Danielle Maddix, François-Xavier Aubet, Kelvin Kan, Jan Gasthaus, and Yuyang Wang. 2021. Learning Quantile Functions without Quantile Crossing for Distribution-free Time Series Forecasting. *arXiv preprint arXiv:2111.06581* (2021).
[18] Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. 2018. Deep state space models for time series forecasting. *NeurIPS* (2018).
[19] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. 2021. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*. PMLR, 8857–8868.
[20] Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs Bergmann, and Roland Vollgraf. 2020. Multivariate probabilistic time series forecasting via conditioned normalizing flows. *arXiv* (2020).
[21] David Salinas, Michael Bohlke-Schneider, Laurent Callot, Roberto Medico, and Jan Gasthaus. 2019. High-dimensional multivariate forecasting with low-rank gaussian copula processes. *Advances in neural information processing systems* 32 (2019).
[22] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting* 36, 3 (2020), 1181–1191.
[23] Phillip Si, Allan Bishop, and Volodymyr Kuleshov. 2021. Autoregressive Quantile Flows for Predictive Uncertainty Estimation. *arXiv preprint arXiv:2112.04643* (2021).
[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.
[25] Ruofeng Wen and Kari Torkkola. 2019. Deep generative quantile-copula models for probabilistic forecasting. *arXiv preprint arXiv:1907.10697* (2019).
[26] Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. 2017. A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053* (2017).
[27] Sifan Wu, Xi Xiao, Qianggang Ding, Peilin Zhao, Ying Wei, and Junzhou Huang. 2020. Adversarial sparse transformer for time series forecasting. *NeurIPS* (2020).
[28] Derek Yang, Li Zhao, Zichuan Lin, Tao Qin, Jiang Bian, and Tie-Yan Liu. 2019. Fully parameterized quantile function for distributional reinforcement learning. *NeurIPS* 32 (2019).
[29] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. 2019. Time-series generative adversarial networks. *Advances in Neural Information Processing Systems* 32 (2019).
[30] Ye Yuan and Kris Kitani. 2019. Diverse trajectory forecasting with determinantal point processes. *arXiv preprint arXiv:1907.04967* (2019).
[31] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of AAAI*.

A HYPERPARAMETER TUNING

We describe here the hyperparameter tuning done for our models and baselines. All hyperparameter tuning was done on the validation split. For our proposed models, `FQFormer` , `FQFormer-BASE` and ablations, we tuned the hyperparameter number of Sparse Attention Layers in the grid $[1, 2, .., 10]$ for each dataset separately [27], and we sampled the learning rate on log-scale uniformly at random in the range $[10^{-4}, 10^{-1}]$ for each initial seed value. In total, we ran 30 configurations per dataset, each for a total of 20 epochs as in [14]. All other hyperparameters were kept constant as the base sparse attention model [14]. Once, the best configuration was found for the `FQFormer-BASE` we froze its parameters and combined it with a 1 layer Transformer proposal network and trained for another 20 epochs. The results stated in Tables 2,3 correspond to the best found checkpoint (on validation split) found on the second stage optimization for the `FQFormer` model.

We also note that for all experiments in our paper, we used the fixed batch size 64 and sampled number of windows as indicated in the dataset statistics (Table. 1), including proposed model, ablations and baseline experiments.

On the other hand, for the Transformer baselines, `LogTrans` and `AST` we used tuned the same hyperparameters as done for our model on the same range and ran the hyperparameter configurations for 20 epochs as well. We used the official code provided by authors[8]. For the RNN baselines, `DeepAR` [9], `SQF-RNN` [9], `IQN-RNN` [10] and `MQ-RNN` [9] we tuned the hyperparameters number of RNN layers in the grid $[4, 8]$, and the cell size in $[256, 512]$ and ran these configurations for the same seeds and sampled learning rate in the log-space range as described above. We used the official code for these models, and ran these models for double the number of epochs 40 per configuration; ensuring fair computational budgets relative to proposed model.

For the Variational autoencoder based baselines, we recall that the `CVAE MSE` , `CVAE DIL` are based on RNNs and use the Mean Squared Error and the DILATE loss [7] and we tuned the cell size and number of layers of RNN similar to noted above for other RNN based models. Additionally, we tuned the fully connected layer dimensionality in the grid $[512, 1024]$ that extrapolates for the forecast in the decoder. We used the best found checkpoint for the `CVAE DIL` model to run the second stage optimization `STRIPE DIL` model. All hyperparameter configurations for `CVAE MSE` , `CVAE DIL` were trained for 40 epochs. We trained 1 epoch each for the second stage optimization routines stripe shape loss and subsequently time loss based optimization. For these experiments, we also used the author's provided official code[11].

B EVALUATION METRICS

We report here the exact definition of the normalized metrics we used to report the results for the metrics, $\text{QL}_{0.5}$, $\text{QL}_{0.9}$, Q-AVG and E-CRPS. The metrics are summed over all time series, i.e., $n = [1, ...N]$, and over the whole prediction range, i.e., $t = [\tau + 1, ..., T]$ [14, 22]. The quantile loss metrics can be stated w.r.t α as follows:

$$QL_{\alpha_i} = \frac{\sum_{n,t} \rho(Y_t^n, q_{\alpha_i,t}^n)}{\sum_{n,t} |Y_t^n|} \tag{20}$$

The Q-AVG corresponds to the average of the 99 quantile loss functions shown above, for $\alpha = [0.01, 0.02, ..., 0.99]$. Similarly, we can state the E-CRPS metric as:

$$E\text{-}CRPS = \frac{\sum_{n,t} \left(\frac{1}{M} \sum_{i=1}^{M} \left| q_{\alpha_i,t}^n - Y_t^n \right| - \frac{1}{2M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \left| q_{\alpha_i,t}^n - q_{\alpha_j,t}^n \right| \right)}{\sum_{n,t} |Y_t^n|} \tag{21}$$

[8] https://github.com/hihihihiwsf/AST
[9] https://github.com/awslabs/gluon-ts
[10] https://github.com/zalandoresearch/pytorch-ts/tree/master/pts
[11] https://github.com/vincent-leguen/STRIPE