GQFormer: A Multi-Quantile Generative Transformer for Time Series Forecasting

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

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

***Abstract*—We propose GQFormer, a probabilistic time series forecasting method that models the quantile function of the forecast distribution. Our methodology is rooted in the Implicit Quantile modeling approach, where samples from the Uniform distribution $\mathcal{U}(0, 1)$ are reparameterized to quantile values of the target distribution. This allows implicit generative quantile modeling without any prior assumptions on the data distribution like Gaussianity, common in prior works. Our work is distinguished from prior quantile forecasting methods by novel methodological advances that relate to directly modeling the correlations among multiple quantile estimations at each forecasting horizon. To this end, we firstly develop a parameter sharing architecture that implicitly models multiple quantile estimations efficiently and secondly regularize these through a novel multi-task loss function formulation that optimizes for quantile estimations to be sharper estimations individually and on the whole be spread maximally apart to capture the various modes of the underlying distribution. We experimentally validate the superiority of the method to state-of-the-art probabilistic forecasting baselines and ablations to the loss formulation.**

***Index Terms*—Probabilistic Forecasting, Implicit Quantile Networks, Sparse Attention Transformer, Multi-task Learning**

I. Introduction

For successful application of forecasting solutions, it is important to quantify uncertainty in the predictions. Hence, recent works focus on probabilistic forecasting to characterize forecast horizons either having low-variance or high noise around mean estimations. We note works [1]–[3] that combine sequential modeling primitives such as convolutions and recurrent hidden states with a likelihood component that outputs parameters for a distribution specified apriori (hence distribution-bound). For many real-world applications, however, such choice can be a limiting factor if at all not difficult to specify apriori [4]. In this paper, we focus on Quantile regression [5], a well-understood statistical method that has been extensively researched for robustly modeling probabilistic outputs [6], [7]. Particularly, in the time series forecasting domain, [8], [9] have combined the sequential models with quantile regression loss functions to generate the 50^{th} and 90^{th} percentile estimations to quantify the uncertainty unhindered by specification of likelihood choice across different underlying data generating distributions.

Despite the distribution-free modeling offered by Quantile regression, the retraining of quantile networks with a differently parameterized quantile loss for each quantile level [8], [9] can limit practical usability whereas other probabilistic models such as the Variational Autoencoder (VAEs) [10], [11], Generative Adversarial Networks (GANs) [12], Variational Flow [13] models can provide quantile estimates corresponding to any level by approximating the full probabilistic density. The Multi-Quantile networks (MQ-RNN) [4], [14] solve for this limitation by learning multiple quantile estimations jointly. On the other hand, Implicit Quantile Networks (IQN) can learn to model the full quantile function. In IQNs, a random $\mathcal{U}(0, 1)$ quantile level is embedded within the neural network through a dedicated embedding component and the corresponding loss function is parameterized with the same quantile level. By sampling multiple quantile levels randomly with stochastic gradient updates, the network can learn to estimate the full underlying distribution with a rather simpler piecewise linear loss function. Merits of IQNs over other approaches include arbitrary extension to many quantile levels, more stable optimization than GANs, and unhindered by structural limitations that arise to avoid intractability in VAEs and Flow based models.

Notably, existing approaches fall short in modeling desired structural artifacts between multiple quantile estimations, for e.g., the 50^{th} and the 90^{th} percentile estimations can share an encoder bottleneck as in MQ-RNN, however, those are still marginal probabilistic estimations. Hence, correlations between quantile estimations are only indirectly modeled through the shared parameters. Similarly, for IQN, a quantile level can be embedded in the input and a single quantile output can be generated corresponding to that level, but each univariate output can only be considered a marginal distribution and correlations between quantile estimations are indirectly modeled in the shared parameter space.

However, from a probabilistic prespective it is important to consider the joint distribution of samples. The Continuous Ranked Probability Score (CRPS) is an established probabilistic metric and is computed as an integral over differencing the predictive CDF with the heavyside function based on the forecast horizon ground truth [15]. Similar to prior work [7], [16] we exploit the Quantile loss equivalent formulation of the CRPS metric. These works repeatedly sample forecast trajectories through RNNs, corresponding to different quantile levels and compute the CRPS metric as an approximation to

978-1-6654-8045-1/22/$31.00 ©2022 IEEE

the integral based on the discrete quantile outputs[1]. Instead, in GQFormer, we utilize an attention based time series history and multivariate quantile representation in the Encoder and a shared fully connected layer over the encoding to generate multiple quantile estimations directly without repeated sampling across forecasting horizons, essentially extending the IQN framework to estimate multiple quantiles. Additionally, to model the correlations between the distribution of the quantile estimations, we rely on another approximation of the CRPS metric based on the Energy score [15]. Specifically, we combine the quantile loss functions with this Energy score based approximation of the CRPS loss in a novel multi-task loss formulation. Although, the equivalence of the Energy score based CRPS and its counterpart quantile loss based approximation has been known, we show combining the two approximations in a novel multi-task loss function learns on a richer gradient signal that covers the inherent biases from both approximations to the integral. For a smooth combination of these loss functions, we estimate the Energy score based CRPS loss function with quantile estimates. Additionally, the Energy score based CRPS loss component regularizes for an explicit structure among these quantile estimates such 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 data generating distribution. In summary,

- We propose a novel forecasting method that generates multiple quantile estimates per forecast horizon by combining the quantile level embeddings and the input time series Attention based embeddings efficiently by exploiting shared parameters extending the IQN framework for multiple outputs.
- We design a novel multi-task loss function for multiple quantile estimates and structural regularization among quantile estimates to capture various modes of the data generating distribution, which goes beyond the scope of heteroscedastic Gaussian distribution based uncertainty quantification.
- We perform extensive experiments to validate the performance of GQFormer compared to several probabilistic forecasting baselines on benchmark datasets.
- We provide a thorough ablation study grounded on rigorously validating the effect of separate building components of the proposed method. Ultimately, proving the method on whole is well-founded.

II. Related Work

Several prominent approaches exist in the literature for probabilistic forecasting, where the distribution of future values is modeled. We note probabilistic models that incorporate a likelihood component that outputs parameters for a distribution specified apriori [1]–[3]. Several of these works serve as baselines, and elaborate more details in the experiments section.

Variational models include a conditional VAE (cVAE) that maps past trajectory and side information to latent codes, which are subsequently decoded to future trajectory estimations [11]. However, since random sampling of future trajectories would be biased towards underlying modes with high likelihood, Determinantal Point Processes (DPP) were also used to sample diverse samples [11]. Another model, STRIPE [17], uses a conditional VAE backbone and new DPP processes to sample diverse future trajectories. GAN based forecasting models include [9], [12]. In [12], a single step probabilistic forecasting model is designed where the Generator and Discriminator networks are composed of RNNs. Additionally, several probabilistic baselines [13], [18] are dedicated to solving the multivariate forecasting problem, where the focus is to model the correlations between the time series channels that are observed at the same time indexes. We note Autoregressive Flow based models [13], where a sequential model component is unrolled over multivariate time series and a series of invertible transformations are applied to derive a density estimation of the multivariate observations offset one step ahead. Inspired by Diffusion models ability to model high-dimensional distributions, an autoregressive extension for multivariate forecasting was also proposed [18].

Among quantile forecasting methods, the model in [19] extended the base RNN component in MQ-RNN with Attention mechanism and further increased the modeling capacity for event indicators. The SQF-RNN model [7] uses an RNN model for estimating the conditional quantile function through regression splines, thereby removing the need to specify a parametric form of the output distribution beforehand. Interestingly, it is trained with an analytic CRPS loss function based on spline-based quantile function representation [7]. Another Transformer model [8] was trained with quantile loss functions for robust estimates of the 10^{th}, 50^{th} and 90^{th} percentile outputs, but interestingly it differs from prior work [14], [19] by use of autoregressive training. Additionally, autoregressive transformer architecture (AST) [9] mitigated the error accumulation problem inherent to autoregressive decoding with a discriminator network that classified the ground truth and generated outputs in an adversarial framework. Moreover, AST was trained separately to estimate the 50^{th} and 90^{th} quantile outputs, requiring extensive retraining for additional quantile level estimations. We also note both prior forecasting approaches where an implicit quantile level was embedded within the network [16], [20] following the IQN framework [6]. In IQN-RNN [16], quantile outputs were modeled by combining RNN representations of the time series and the implicit quantile embedding. Whereas, in [20] an approach to generative quantile forecasting was developed extending the base MQ-RNN model. This involved a Copula component that learned to allow the possibility of capturing latent interactions between singular quantile outputs across forecast horizons. Another work [4] focuses on solving the *quantile crossing* problem that arises when quantile estimations are made separately for a particular forecasting horizon as done in [6], [8], [9], [14], [19]. The fundamental idea to solve quantile crossing is to structure the output such that successive quantile estimations add on to preceding ones and all quantile outputs are constrained to be positive with

[1]Practically, the real forecast distribution at a forecast horizon is unkown

commonly available activation functions. Similar to SQF-RNN, an analytical CRPS loss estimation was derived based on pre-specified quantile levels and fixed, or learnable spline based inter/extrapolations beyond those. A multivariate quantile function based forecasting method has also been recently proposed [21]. This network generates monotonic quantiles estimations and also uses the Energy score based CRPS optimization.

In summary, many prior works do not explicitly model the correlations among several quantile estimations per horizon [6], [14], [16], [19], [22]. Prior quantile methods model the correlations [4], [7], however, chose restrictive spline representations. We optimize a multi-task loss combining quantile loss and the Energy score loss functions. The Energy score based CRPS loss function parameterized through quantile estimations leads to modeling these correlations directly. This also differentiates our work to [21] which neither embeds implicit quantile levels nor takes advantage of the robust quantile loss functions. Besides, for multiple quantile estimations, we extend the implicit quantile forecasting work with a more powerful sparse attention model and exploit parameter sharing for efficient multi-task estimation.

III. Background

A. 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[2]. The variable τ is used to indicate the partitioning of the conditioning and the forecasting ranges. In addition to the real-world time series we also consider C many social time[3]covariates $X \in \mathbb{R}^{T\times C}$ that are observed in the entire range. We aim 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. This is in contrast to other works that reduce the problem complexity by formulating a simpler single step forecasting task $p(Y^n_{\tau+1}|Y^n_{1:\tau}, X_{1:\tau+1}, \Theta)$[4]. Note that our formulation and following background is similar to [23].

B. Quantile Regression

In order to learn a distribution of the future possible outcomes, we can consider modeling the cumulative distribution (CDF) of the random variable $Y^n_{\tau+1} \in \mathbb{R}$ [4], [5], [7]. Let us denote the CDF by $F_Y(y)$, then the $\alpha \in (0, 1)$ quantile can be given as:

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

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:

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

We can model the α quantile estimate by minimizing expected quantile loss[5],

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

The loss function, $\rho_\alpha(Y, q_\alpha)$ is given as:

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

Intuitively, the quantile loss can be considered as a weighted generalization of the L_1 loss. Hence, with rather simple quantile level $\alpha_{1:M} \in \mathcal{U}(0, 1)$ parameterized piecewise linear loss functions, we can model the conditional quantile distribution.

C. Continuous Ranked Probability Score

We now extend the discussion towards metrics for evaluating probabilistic forecast distributions. Proper scoring metrics are minimized when the predictive distribution is equivalent to the data distribution. For example, for deterministic predictions, we can consider the mean absolute error. Importantly, proper scoring metrics are negatively oriented, and generally the optimization is cast to minimize the corresponding loss functions. The CRPS metric can be computed as:

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

Where $\mathbb{I}\{Y \leq y\}$ denotes the indicator function [15]. Closed-form expressions for popular distributions such as Gaussian and Negative-binomial likelihood exist for integral in Eq.6 [15], [24], but for modeling various real-world data generating processes such assumptions can be overly simplistic and restrictive. Therefore, prior works considered the following equivalent formulation based on quantile loss [25], [26][6],

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

Besides the above formulations, we can note another CRPS approximation on the empirical CDF $\hat{F}_Y(y)$, referred to as the Energy score based approximation [15], [24][6]. We consider modelling this approximation via M many *quantile estimations*,

$$\hat{F}_Y(y) = \frac{1}{M}\sum_{i=1}^{M} \mathbb{I}\{Y \leq q_{\alpha_i}\} \tag{8}$$

$$\begin{aligned}\text{CRPS}(\hat{F}_Y(y), Y) = \frac{1}{M}\sum_{i=1}^{M} \left|q^n_{\alpha'_i,t} - Y^n_t\right| - \\ \frac{1}{2M^2}\sum_{i=1}^{M}\sum_{j=1}^{M} \left|q^n_{\alpha'_i,t} - q^n_{\alpha'_j,t}\right|\end{aligned} \tag{9}$$

[2] t is relative, can correspond to different time across time series

[3] time-of-the-day, week-of-the-month etc

[4] Horizons $[\tau+2, ..., T]$ can be modeled autoregressively via past predictions

[5] 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

[6] Equivalence proofs have been derived in these works

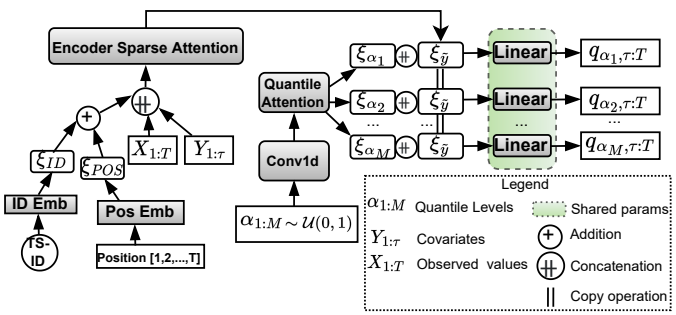

Fig. 1: architecture showcasing the estimations of various quantile levels for all horizons. Layer depth and loss components are not shown.

IV. Method

A. Position and Time-Series ID Embedding

Following previous work [1], [3] for positional encoding, we build another channel $(D+C+1)$ containing ordinal counting for the entire range considered $t = [1, ..., T]$. Each ordinal level in this channel is embedded with a learned embedding d_{model}. This embedding component allows the model to learn a relative positional information for the sequential time series input. Similarly, we can embed IDs of each time series. Each series $n \in [1, ..., N]$ is given a unique ordinal count based ID, which is embedded identically but independently to positional embedding. As we aim to learn a global model across multiple time series, this allows the modeling of individual latent patterns associated to individual time series. As the model observes more data from a series, it improves its ID embedding. $\xi_{pos} \in \mathbb{R}^{d_{model}}$ and, $\xi_{ID} \in \mathbb{R}^{d_{model}}$ denote the position and ID embeddings, respectively.

B. Implicit Quantile Level Embedding

We now describe the embedding of sampled α values that leads to modeling a full conditional quantile distribution [6]. The aim is to embed various $\alpha_{1:M} \in \mathcal{U}(0,1)$ through a dedicated embedding component to model the forecast distribution by parameterizing the quantile loss functions in Eq. 5 with the same α values. To fix ideas, we represent the learned embedding component as $\xi_\alpha \in \mathbb{R}^{d_{model}}$. In our work, we choose a self-attention based embedding,

$$\xi_\alpha = \max(0, \alpha W_1 + b_1) \tag{10}$$

$$\xi_\alpha = \text{Attention}(\xi_\alpha, \xi_\alpha, \xi_\alpha) \tag{11}$$

We sample multiple α values for an input batch which pass through an embedding stage. The embedding is composed of a non-linear ReLU based feed-forward network $(W_1 \in \mathbb{R}^{M \times d_{model}}, b_1 \in \mathbb{R}^{d_{model}})$ applied to each α level in a position-wise manner and can be thought of as 1d convolution [27]. Next a deep self-attention based embedding of the dimensionality $\mathbb{R}^{d_{model}}$ is learned where the Query, Key and Value are independent transformations of the same ξ_α. This embedding component hence ensures that the model can exploit latent embeddings for various α levels. We ensure that the model converges to minimizing the expected quantile loss for multiple α for multiple horizons, structuring the optimization based on stochastic batch updates by sampling M many $\alpha_{1:M} \sim \mathcal{U}([0,1])$ values in one batch update same for multiple horizons. This is considerably more efficient than sampling all possible α values for every time series sample and horizon individually, and allows for significant speedups through broadcasting the quantile loss functions.

C. Sparse Self-Attention Encoder

The Implicit Quantile Level Embedding methodology [6] is architecturally compatible with several existing time series forecasting methods, although with varying levels of modifications. In this paper, we focus on the Transformer architecture, which has been recently shown to excel for probabilistic forecasting tasks [3], [13]. Transformer architectures [27] model pairwise interactions among all input tokens that leads to a $O(T^2)$ complexity. This allows the model to capture long-range temporal dependencies, however, raises practical issues regarding compute and memory requirements. A natural choice for the Sparse Attention model is the *Log Sparse Transformer* [3] which calculates $O(logT)$ dot products for each timestep in each layer by restricting the attention representations to be computed only causally with an exponential step size. Hence, the complexity could be reduced from $O(T^2)$ to $O(T \log T)$. We denote this attention computation scheme as LogAttention and compute encoding:

$$\xi_{\tilde{y}} = [Y_{1:T} + X_{1:T} + (\xi_{pos} + \xi_{ID})] \tag{12}$$

$$\xi_{\tilde{y}} = \text{LogAttention}(\xi_{\tilde{y}}, \xi_{\tilde{y}}, \xi_{\tilde{y}}) \tag{13}$$

Where, $+$ operator denotes concatenation. 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 [28]. Additionally, the ξ_{ID} are repeated along the time axis for addition. We also note that the LogAttention layers are initialized with the dimensionality d_{model} plus the dimension of the multivariate time series input.

D. Decoder

In the forecasting literature, for short-range rolling forecasting application, autoregressive decoding that models correlations between sequential outputs is preferred [1], [3], whereas for long-range forecasting application a multi-task decoding scheme that avoids error accumulation inherent to autoregressive decoding is used [14], [28]. Our proposed model is built as a multi-task decoder over the base sparse attention mechanism encoding from [3]. However, we derive an autoregressive quantile forecasting decoder counterpart to our proposed model and describe it further in the experiments section as a baseline.

1) Multi-task Decoding: The fundamental idea is to combine the quantile id embeddings and the time series embeddings in

cost-effective manner exploiting shared parameters for multiple quantile forecasts.

$$\xi_{\tilde{y}}^{Flat} = \text{Flatten}(\xi_{\tilde{y}}) \tag{14}$$

$$\xi_{\tilde{y}}^{1:M} = \text{Repeat}(\xi_{\tilde{y}}^{Flat}, M) \tag{15}$$

$$q_{\alpha_i,\tau:} = [\xi_{\tilde{y}}^{i} + \xi_{\alpha_i}] W_{MTL} + b_{MTL} \quad \forall i = [1, ..., M] \tag{16}$$

In the above equations, we first flatten the embedding of the time series to one feature axis, this results into the embedding size: $(d_{model} \times \text{len}(1:\tau))$, where d_{model} indicates the embedded dimensionality of each timestep input. Next we repeat these M many times to combine these with the quantile embeddings in Eq. 10. Observe that each of the $[1, ...M]$ quantile embedding is different, but the time series embedding $\xi_{\tilde{y}}^{Flat}$ remains the same. Finally, a shared fully connected layer, given by parameters $W_{MTL} \in \mathbb{R}^{(d_{model} \times \text{len}(1:\tau) + d_{model}) \times \text{len}(\tau+1:T)}$,$b_{MTL} \in \mathbb{R}^{\text{len}(\tau+1:T)}$ 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. By repeatedly calling the layer, $[1, ..., M]$ many times, each time with different quantile embedding, we can generate the M quantile forecasts for each forecast horizon $[\tau + 1, ..., T]$. In summary, we utilize the same *direct forecasting* [4], [14], [28], [29] for all horizons with a Linear layer and flattened time series embedding, however, we repeatedly call the same layer to generate the respective $[1, ..., M]$ quantile forecasts.

Given our focus is on generating multiple quantile forecasts simultaneously, we can contrast the above motivated parameter sharing approach with that of a compute and memory intensive strategy of learning multiple quantile forecasts by stacking M many fully connected layers $(W_{2:M}, b_{2:M})$ on the flattened representation of the time series embedding. Additionally, since we sample quantile levels randomly, by sharing the layer across the quantile level embeddings we allow for the possibility for faster convergence. On the other hand, multiple fully connected layers would only update parameters corresponding to only one quantile level embedding for each gradient update.

2) Quantile Transposed Attention: We also motivate another hierarchical representation to be learned across the quantile forecasts. The intuition is to learn pairwise interactions between the quantile forecasts similar to the pairwise interactions learned for time steps as follows:

$$q_{\alpha_{1:M},\tau:} = \text{Attention}(\text{Tr}(q_{\alpha_{1:M},\tau:}), \text{Tr}(q_{\alpha_{1:M},\tau:}), \text{Tr}(q_{\alpha_{1:M},\tau:})) \tag{17}$$

Where the function $\text{Tr}()$ computes the transpose and expresses the necessary change in the input dimension in order to compute the $\text{Attention}()$ representations for the quantile forecasts instead of on the embedded time series features. This equips the model to directly learn representations that contrast all quantile forecasts to each other, which the earlier components do not directly learn. For example, attention representations between the 50^{th} and 90^{th} estimations could be computed with the sequential information kept as latent features. Feature transposition based learning has been utilized before in [30], although not in the context of Attention.

3) Auxiliary Reconstruction with Multi-Task Decoding: We can observe that the multi-task decoder motivated above does not reconstruct the time series given in the input range and only outputs the forecasts required in the prediction range $t = [\tau, ..., T]$, whereas the autoregressive training through causal masking and offsetting of targets by being always one step ahead inherently reconstructs the time series for the entire range since the input is already given as a concatenation of the input and the prediction range both [1], [3], [7]. In order to ensure more supervision is granted and the multi-task decoder can learn on correlations between more time steps, we increase the dimensionality of the fully connected layer in Eq. 16 to also reconstruct the time series up to a certain limit tuned as a hyperparameter $\tau - r$. Note that prior direct forecasting approaches (i.e. multi-task decoding) [4], [14], [28], [29] do not reconstruct input. Notably, both autoregressive and multi-task decoding approaches use future covariate information but reconstruct and forecast only the quantiles of the target time series channel.

E. Optimization

As previously motivated, to model the correlations between the distribution of the quantile estimations, we combine the quantile loss functions with the Energy score based approximation of the CRPS loss in a novel multi-task loss formulation. We reiterate the important consideration of approximating the Energy score loss component with quantile estimates that correspond to the quantile level embeddings in input.

$$\underset{\Theta \in \mathbb{R}}{\arg\min} \sum_{n=1}^{N} \sum_{t=\tau-r}^{T} \left(\sum_{i=1}^{M} \rho_{\alpha_i}(Y, q_{\alpha_i}) + \frac{1}{M} \sum_{i=1}^{M} \left| q_{\alpha_i',t}^{n} - Y_t^n \right| - \frac{1}{2M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \left| q_{\alpha_i',t}^{n} - q_{\alpha_j',t}^{n} \right| \right) \tag{18}$$

The above multi-task loss function combines sub-objectives which are three separate granular instantiations of the L_1 loss functions. We can have a look at the first sub-objective that minimizes a weighted specific parameterization of the L_1 loss that corresponds to the quantile loss and enables modeling a quantile level of the underlying distribution. The second sub-objective optimizes for a sharper forecast sample, regardless of which quantile level is being modeled. The third sub-objective optimizes for variability in the output forecast distribution since different quantile estimation although are required to be sharper; jointly are optimized to be spread further apart such that the pairwise distances among the estimates are maximum. It can be observed that the sub-objectives can compete to optimize the shared model parameters in different directions, however, recent work [31] has shown that with sufficient model capacity and weighing of sub-objectives, this issue can be resolved and does not impede direct optimization of the multi-task loss, and we also further experimentally validate this in the experiment section. Finally, we note that despite the equivalence of the CRPS formulations through Quantile losses or the Energy score based formulation, the multi-task loss in

Table I: Summary of dataset statistics. Multiple training examples are sampled with the sliding window procedure.

	electricity$_{24}$	traffic$_{24}$	electricity$_{168}$	traffic$_{168}$	wind	solar	m4-hourly
# time series, N	370	963	370	963	28	137	414
time granularity	hourly	hourly	hourly	hourly	daily	hourly	hourly
domain	$\mathbb{R}^+$	$[0,1]$	$\mathbb{R}^+$	$[0,1]$	$\mathbb{R}^+$	$\mathbb{R}^+$	$\mathbb{R}^+$
# training examples	500K	500K	500K	500K	10K	50K	50K
# input length, $[1...\tau]$	168	168	168	168	168	90	168
# forecasting length, $[\tau,...,T]$	24	24	168	168	24	30	48

Eq. 18. learns on approximations where intuitively biases from both approximations can be exploited in learning. However, the major gradient signal is still derived from the quantile losses, since we use the Energy score loss component as auxiliary regularization through weighted averaging. Our weighting strategy can be explained with respect to the normalization of the loss in Eq.18. We specifically normalize the loss with the factor $(N+(T-r)+(M+1))$ such that the Energy score loss component is weighted as a single quantile loss component.

V. Experiments[7]

Our primary set of experiments is based on two real-world datasets on four different forecasting tasks. We follow the experimental protocol from previous works [1], [3]. Additionally, we also report results on three other smaller datasets, as highlighted by the number of windows sampled for training and other dataset statistics stated in Table. I. Note that the experiments are followed and extended from [23].

A. Dataset Statistics

Our experiments are based on well-established benchmarks for the following datasets:

1) `electricity` dataset composes of hourly Kilo Watts electricity consumption of 370 houses from 2011-2014.
2) `traffic` dataset composes of hourly occupancy rates in the range$[0,1)$ of 963 car lanes in 2008.
3) `wind` dataset where the daily energy potential is noted across 28 regions from 1986 to 2015.
4) `solar` describes hourly sampled solar power generation records from 2006 from 137 Photovoltaics plants.
5) `m4-hourly` has 414 time series from various sources.

B. Baselines

1) `ARIMA` [3] models forecasts as linear combination of past time series values.
2) `ETS` [3] computes forecasts as weighted averages of past observations, with the weights exponentially decaying for past observations.
3) `TRMF` [32] factorizes the matrix of time series observations global latent features and autoregressive temporally regularized features per time step.
4) `DeepState` [2] forecasts through a linear Gaussian state-space model whose state and transition parameters are predicted via an underlying RNN.
5) `DeepAR` [1] recursively unrolls the hidden state for each time step and a linear layer extrapolates from hidden state to (μ,σ) for forecasts autoregressively.
6) `LogTrans` [3] is a Gaussian likelihood based sparse transformer that autoregressively decodes future values. Extrapolating (μ,σ) is similar to [1]. This model is also used as the base attention module in `GQFormer`.
7) `CVAE MSE` [11], [17] is a conditional VAE trained with mean squared error and quantiles can be computed on samples from the learned sampling layers.
8) `CVAE DIL` [17] is similar to the cVAE model from [11] trained with shape and temporal loss to generate sharper samples.
9) `MCVAE MSE` [17] is our proposed extension to `CVAE MSE` that learns on multivariate covariates available for past and future since `CVAE MSE` only uses target channel. We simply appended the covariate information in the channel space in the base sequential RNN models in the VAE, similar to `DeepAR` incorporates.
10) `STRIPE DIL` [17] can generate sharper future trajectory samples through learning DPP processes finetuned on first stage `CVAE DIL`.
11) `AST` [9] autoregressively decodes for a single fixed quantile level, requires retraining for other quantile levels.
12) `MQ-RNN` [14] estimates multiple quantiles per forecasting horizon through parameter sharing but does not model any correlations among quantiles.
13) `SQF-RNN` [7] is an RNN that estimates the Quantile function through isotonic splines and minimizes an analytical CRPS based on the spline representation.
14) `IQN-RNN` [16] is an RNN based on IQN framework with independent sampling runs to estimate multiple quantiles

C. Proposed Model and Ablations

1) `GQFormer` is our proposed model with multi-task decoding Eq. 16, quantile transposed attention Eq. 17, auxiliary reconstruction Sec. IV-D3 and optimized via Eq.18.
2) `GQFormer-BASE` is `GQFormer` without quantile transposed attention and Energy score based loss components in optimization, geared towards short range multi-task forecasting with fewer parameters.
3) `A-SEQ` is the sequential autoregressive decoding model similar to `LogTrans` that generates multiple quantile estimates and trained with quantile loss. The 50^{th} quantile is used for autoregressive decoding.
4) `A-FIX` does *fixed* quantile modeling by decoding 99 discrete quantile level estimations. It does not reconstruct the input, nor uses quantile attention and is trained with Quantile loss functions only.
5) `A-REC` is `GQFormer-BASE` without reconstruction.
6) `A-CRPS` is the `GQFormer-BASE` with quantile transposed attention without the Energy score based CRPS

[7] github.com/super-shayan/gqformer. Appendix A provides hyperparameter tuning and implementation details

Table II: Performance comparison in terms of individual quantile metrics. Results are formatted with $\cdot 10^2$, columnar least is boldfaced, second-least is underlined. Reimplementation results are stated in brackets.

		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 [3]	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
	MCVAE MSE	11.232	9.508	13.439	11.451	22.606	20.908	24.670	23.591
	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	GQFormer	6.604	3.358	7.416	**3.697**	12.055	8.651	13.215	9.154
	GQFormer-BASE	6.315	3.133	7.876	3.745	10.756	**7.866**	13.758	9.676
Abalations	A-SEQ	6.587	7.046	7.881	9.078	**10.377**	10.654	13.277	14.274
	A-FIX	6.252	3.344	7.359	4.192	12.168	10.226	14.959	12.725
	A-REC	6.258	3.669	7.484	4.257	12.559	12.161	**12.541**	11.267
	A-QATTN	6.502	3.104	7.856	3.784	12.022	9.441	15.670	11.050
	A-CRPS	6.689	3.453	8.137	3.772	11.960	8.484	14.474	9.488
	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
	A-FQFormer	6.418	3.133	7.880	3.729	11.023	7.890	13.255	9.353

loss. As a result, we can therefore judge the importance of the quantile transposed attention.

7) A-QATTN is the GQFormer-BASE with Energy score based CRPS without quantile transposed attention.
8) A-DIR is the GQFormer-BASE optimized directly with Energy score based CRPS loss without the main Quantile loss functions. Hence, it is similar to [21] where only the Energy score can be used for optimization.
9) A-PRE-DIR pretrains GQFormer-BASE with the quantile loss and in the second stage the model is optimized again with only Energy score based CRPS for same epochs. This ablation therefore studies the impact of the joint multi-task training as opposed to a two-stage optimization.
10) A-FQFormer can be considered an ablation [23], it predicts the α quantile levels for a pretrained and frozen GQFormer-BASE and optimizes forecasts from it for the Energy score based CRPS loss function.

D. Evaluation Protocol

Firstly, we describe the preprocessing which follows prior work [1], [3]. A sampling window procedure is used to generate multiple training samples by sampling τ multiple times within the total time series ranges. This leads to fixed length time series input and targets for the learning algorithms. Notably, we also utilize similar weighted sampling procedure as prior work [1], [3], which leads to sampling windows proportional to scale. The preprocessing therefore ensures direct comparability to prior work [3], [23]

We summarize the results comparing our proposed models with baselines and several ablations in the Tables. II,III. We note that for each method, independent hyperparameter tuning was carried out for each forecasting dataset and task combination with fair computational budgets [23]. The test set error result stated corresponds to the forecasting last 7 days for each dataset, either in a rolling or direct forecasting manner [1], [3]. The test set error corresponds to each method's best performance on a separate held-out validation set across 3 random seeds. The validation set is derived from within the training range, covering data before the testing range's last 7 days. We study short horizon forecasting and long-range forecasting as separate tasks for both these datasets. Hence, forecasting for a short-range for only 24 horizons in a rolling manner, and forecasting for a long-range for all 168 horizons directly stands to evaluate the model and baseline performances from two different standpoints. Lastly, Q-AVG denotes the averaged quantile loss (Eq. 7) over 99 quantile estimations on the discrete grid $\alpha_{1:M} = [0.01, 0.02, ..., 0.99]$ whereas E-CRPS corresponds to the loss stated in Eq. 9 estimated through the same quantile estimation as in Q-AVG.

E. Results

As our first result, we compare the performance of the models GQFormer and GQFormer-BASE to various Classical, Gaussian likelihood based, VAE based, Quantile loss based forecasting baselines on individual quantile forecasting metrics $\text{QL}_{0.5}$ and $\text{QL}_{0.9}$ respectively. We can see that the proposed models on average perform better than various baselines, however do not always lead to the least error across the different tasks and metrics. It is worth emphasizing that the LogTrans baseline performs exceptionally better than other Recurrent Neural Network based baselines given the efficacy of the Attention mechanism in learning richer latent

Table III: Performance comparison in terms of probabilistic metrics averaged over the discrete quantile level grid. Formatting is similar to before. AST requires retraining, rendering it inapplicable here. * marks the E-CRPS computed w.r.t quantile proposals.

		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
	MCVAE MSE	10.962	10.944	13.305	13.295	22.233	22.209	24.377	24.354
	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	GQFormer	4.956	4.924	**5.646**	**5.605**	9.333	9.254	**10.185**	**10.088**
	GQFormer-BASE	4.770	4.739	5.952	5.913	**8.327**	**8.267**	10.574	10.475
Abalations	A-SEQ	6.60	6.382	7.897	7.687	10.422	10.098	13.354	12.938
	A-FIX	4.981	4.955	5.974	5.947	10.432	10.393	12.953	12.911
	A-REC	4.959	4.933	6.084	6.058	11.075	11.035	10.802	10.760
	A-QATTN	4.918	4.883	5.993	5.953	9.360	9.279	12.267	12.156
	A-CRPS	5.055	5.024	6.093	6.046	9.189	9.109	11.160	11.073
	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
	A-FQFormer	4.853	4.800*	5.952	5.888*	8.506	8.425*	10.227	10.138*

representations. We can also compare the performance to the ablation methods. Given their similarity to the GQFormer model with learnable encoding modules, whereas differences only relate to loss function and decoding modules dedicated to multiple quantile modeling, we can see that on the simpler individual quantile metrics the performances are comparable. Interestingly, A-SEQ is able to clearly outperform all baselines on the rolling forecasting of traffic dataset (traffic$_{24}$), which is intuitive given autoregressive decoding is the default choice for rolling forecasting tasks in existing literature. Moreover, despite offering generative quantile estimates, we use the 50^{th} percentile estimate to autoregressively decode in inference and the reason why the QL$_{0.9}$ result might not be optimal in comparison. Partly, this also motivates the autoregressive decoding based LogTrans baseline's better performance on the electricity$_{24}$ task. Moreover, given the consensus from recent works on modeling the Electricity dataset with Gaussian likelihood further motivates the better performance of the baseline on this task but at the same time long-range direct forecasts are best made with the multi-task decoding as we show for GQFormer. Our main set of results are stated in Table III, in terms of CRPS metrics that evaluate probabilistic forecasts more comprehensively across several quantile levels. Here, we can clearly see that the proposed GQFormer models are able to outperform all ablations. Given that we tuned the hyperparameters of these ablation methods independently for each forecasting task, it is reasonable to expect better one-off performances, however, on representative probabilistic metrics GQFormer performs better. Comparing the performance of the Gaussian likelihood based DeepAR and quantile methods MQ-RNN, IQN-RNN and SQF-RNN, we can note that despite sharing the same RNN based learnable modules and learnable parameters, the quantile methods fair better, showcasing the advantages of quantile forecasting. The variational models, CVAE MSE and CVAE DIL [11], [17] also approximate the true forecast distribution as a Gaussian. This combined with the RNN based learnable modules for learning time series representations leads to less competitive results. Moreover, we can observe that the CVAE DIL model performed subpar. Since STRIPE DIL is a two-stage optimization based method, that relies on a pretrained CVAE DIL method, its performance was also degraded. Given the lack of modeling capacity for covariate information such as social time based features in the original formulation of the variational models, we designed the extension MCVAE MSE by simply appending the covariate information in the channel space in the base sequential RNN models in the VAE similar to how other RNN based methods in the experiments incorporated covariate information. Nevertheless, MCVAE MSE did not benefit from the additional covariate information. We posit that more advanced architectural modifications might be required to model the covariate information. On the other hand, we found that AST is prone to degenerate optimization prone to GAN based frameworks and that lead to poor performance on the traffic forecasting tasks. Given that AST required retraining for each quantile level, training it on many quantile levels was rendered inapplicable. Comparing the performance of GQFormer to GQFormer-BASE we can note that the Energy score based optimization and the transposed quantile attention benefit the long-range forecasting task the most. Given long-range forecasting is inherently a more difficult forecasting task, the additional transposed quantile attention module and the regularization from the Energy score loss is advantageous. Additionally, better performance achieved by the proposed GQFormer models compared to learning on fixed quantile levels as done with A-FIX showcases the advantage of learning multiple quantiles implicitly. Notably, we see that this difference is more pronounced than comparing IQN-RNN and MQ-RNN where the main difference also lies in learning quantile estimations implicitly versus fixed. However, IQN-RNN does not learn multiple quantiles as

Table IV: Comparison on additional datasets in terms of individual quantile metrics

		wind$_{24}$		solar$_{30}$		m4-hourly$_{48}$	
		QL$_{0.5}$	**QL$_{0.9}$**	**QL$_{0.5}$**	**QL$_{0.9}$**	**QL$_{0.5}$**	**QL$_{0.9}$**
Rep [3]	TRMF	31.1	–	24.1	–	–	–
	DeepAR	28.6(29.136)	11.6(12.665)	22.2(22.952)	9.3(8.666)	9.0(4.053)	3.0(4.523)
	LogTrans	**28.4**(28.677)	**10.8**(11.669)	**21.0**(21.437)	**8.2**(8.861)	6.7(4.724)	2.5(4.28)
Prop	GQFormer	29.108	11.126	21.935	8.256	4.731	3.566
	GQFormer-BASE	29.389	11.661	22.769	8.919	**4.023**	**2.1**

Table V: Comparison on additional datasets in terms of CRPS metrics

		wind$_{24}$		solar$_{30}$		m4-hourly$_{48}$	
		Q-AVG	**E-CRPS**	**Q-AVG**	**E-CRPS**	**Q-AVG**	**E-CRPS**
Gaus	DeepAR	21.038	20.787	16.332	16.154	3.621	3.597
	LogTrans	20.361	20.127	**15.803**	**15.619**	3.932	3.897
Prop	GQFormer	**20.254**	**19.949**	15.882	15.899	3.706	3.689
	GQFormer-BASE	20.508	20.335	16.784	16.684	**2.999**	**2.979**

the GQFormer models. Moreover, comparing GQFormer and GQFormer-BASE to A-REC shows the importance of the auxiliary reconstruction of the time series from the history. The results for the ablations A-QATTN and A-CRPS validate that GQFormer with both Quantile Attention Eq. 17 and optimized with the joint multi-task loss performs best for the long-range forecasting. Moreover, regarding optimization, comparing GQFormer to A-DIR we can see that optimization alone on the Energy score based CRPS is suboptimal compared to the joint multi-task loss and weighting the loss weight less in the multi-task loss is also justified in Eq. 18 since GQFormer-BASE optimized only with the quantile loss functions outperforms it significantly. We can also infer that learning the quantile levels contextually in A-FQFormer through a second stage optimization procedure over the Energy score loss provides a minimal lift over GQFormer-BASE on the long-range forecasting tasks, however, GQFormer still outperforms it. Additionally, we explored sequential two-stage optimization in A-PRE-DIR, however the multi-task optimization outperforms it. Lastly, we can observe that our proposed models fair well on additional small datasets, however, LogTrans leads in terms of individual quantile metrics in Table. IV. We hypothesize this is due to the short range of the two datasets solar$_{30}$ and wind$_{24}$ where recursive forecasting can be worthwhile. Nevertheless, in Table. V, in terms of representative probabilistic CRPS metrics, proposed models perform better.

VI. Conclusion

In this work, we proposed a probabilistic forecasting method that implicitly generates multiple quantile estimates per forecast horizon and models the correlations among those estimations through a novel multi-task loss formulation. Empirical evaluation showed the model outperformed several forecasting baselines. In addition, we showcased that the multi-task loss formulation and transposed quantile attention are key to modeling the long-range forecasts through an ablation study. In future work, we shall research novel multivariate forecasting extensions that model correlations among quantile functions of covariate channels in high-dimensional time series.

VII. Acknowledgements

This work is co-funded by the industry project Data-driven Mobility Services of ISMLL and Volkswagen Financial Services; also through BMWK Germany project IIP-Ecosphere: Next Level Ecosphere for Intelligent Industrial Production.

References

[1] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "Deepar: Probabilistic forecasting with autoregressive recurrent networks," *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[2] S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski, "Deep state space models for time series forecasting," *NeurIPS*, 2018.

[3] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[4] Y. Park, D. Maddix, F.-X. Aubet, K. Kan, J. Gasthaus, and Y. Wang, "Learning quantile functions without quantile crossing for distribution-free time series forecasting," *arXiv preprint arXiv:2111.06581*, 2021.

[5] R. Koenker and K. F. Hallock, "Quantile regression," *Journal of economic perspectives*, vol. 15, no. 4, pp. 143–156, 2001.

[6] W. Dabney, G. Ostrovski, D. Silver, and R. Munos, "Implicit quantile networks for distributional reinforcement learning," in *International conference on machine learning*. PMLR, 2018, pp. 1096–1105.

[7] J. Gasthaus, K. Benidis, Y. Wang, S. S. Rangapuram, D. Salinas, V. Flunkert, and T. Januschowski, "Probabilistic forecasting with spline quantile function rnns," in *International conference on Artificial Intelligence and Statistics*. PMLR, 2019.

[8] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *International Journal of Forecasting*.

[9] S. Wu, X. Xiao, Q. Ding, P. Zhao, Y. Wei, and J. Huang, "Adversarial sparse transformer for time series forecasting," *NeurIPS*, 2020.

[10] V. L. Guen and N. Thome, "Probabilistic time series forecasting with structured shape and temporal diversity," *arXiv preprint arXiv:2010.07349*, 2020.

[11] Y. Yuan and K. Kitani, "Diverse trajectory forecasting with determinantal point processes," *arXiv preprint arXiv:1907.04967*, 2019.

[12] A. Koochali, A. Dengel, and S. Ahmed, "If you like it, gan it—probabilistic multivariate times series forecast with gan," in *Engineering Proceedings*, 2021.

[13] K. Rasul, A.-S. Sheikh, I. Schuster, U. Bergmann, and R. Vollgraf, "Multivariate probabilistic time series forecasting via conditioned normalizing flows," *arXiv*, 2020.

[14] R. Wen, K. Torkkola, B. Narayanaswamy, and D. Madeka, "A multi-horizon quantile recurrent forecaster," *arXiv:1711.11053*, 2017.

[15] T. Gneiting and A. E. Raftery, "Strictly proper scoring rules, prediction, and estimation," *Journal of the American statistical Association*, vol. 102, no. 477, pp. 359–378, 2007.

[16] A. Gouttes, K. Rasul, M. Koren, J. Stephan, and T. Naghibi, "Probabilistic time series forecasting with implicit quantile networks," *arXiv*, 2021.
[17] V. Le Guen and N. Thome, "Probabilistic time series forecasting with shape and temporal diversity," *Advances in Neural Information Processing Systems*, vol. 33, 2020.
[18] K. Rasul, C. Seward, I. Schuster, and R. Vollgraf, "Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8857–8868.
[19] C. Eisenach, Y. Patel, and D. Madeka, "Mqtransformer: Multi-horizon forecasts with context dependent and feedback-aware attention," *arXiv preprint*, 2020.
[20] R. Wen and K. Torkkola, "Deep generative quantile-copula models for probabilistic forecasting," *arXiv preprint arXiv:1907.10697*, 2019.
[21] K. Kan, F.-X. Aubet, T. Januschowski, Y. Park, K. Benidis, L. Ruthotto, and J. Gasthaus, "Multivariate quantile function forecaster," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 10 603–10 621.
[22] T. FORECASTS, "Mecats: Mixture-of-experts for probabilis-tic forecasts of aggregated time series."
[23] S. Jawed and L. Schmidt-Thieme, "Fqformer: A fully quantile transformer for time series forecasting," *8th SIGKDD International Workshop on Mining and Learning from Time Series–Deep Forecasting: Models, Interpretability, and Applications*, 2022.
[24] A. Jordan, F. Krüger, and S. Lerch, "Evaluating probabilistic forecasts with scoringrules," *arXiv preprint arXiv:1709.04743*, 2017.
[25] F. Laio and S. Tamea, "Verification tools for probabilistic forecasts of continuous hydrological variables," *Hydrology and Earth System Sciences*, 2007.
[26] P. Si, A. Bishop, and V. Kuleshov, "Autoregressive quantile flows for predictive uncertainty estimation," *arXiv:2112.04643*, 2021.
[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*.
[28] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.
[29] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Neural Information Processing Systems*, 2021.
[30] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *Advances in Neural Information Processing Systems*, 2021.
[31] M. Ruchte and J. Grabocka, "Multi-task problems are not multi-objective," *arXiv preprint arXiv:2110.07301*, 2021.
[32] H.-F. Yu, N. Rao, and I. S. Dhillon, "Temporal regularized matrix factorization for high-dimensional time series prediction," *NeurIPS*, vol. 29, 2016.

A Hyperparameter Tuning

The experiment setup is extended from prior work [23]. For all Attention based models, proposed and otherwise, we tuned the hyperparameters of the number of sparse Attention layers in the grid $[1, 2, ..., 10]$ for each dataset's forecasting task. For each such configuration, we sampled a learning rate on the log-scale uniformly at random $[10^{-4}, 10^{-1}]$ and ran it for 3 separate seeds resulting into 30 configurations per dataset, each optimized for 20 epochs. Notably, we kept the rest of the hyperparameters same as the gaussian autoregressive sparse attention model [3]. This setup ensured a fair comparison. We also used the author's original code[8]. Moreover, we can note that the batch size was fixed to 64 for all experiments.

Additionally, for our proposed models, `GQFormer` and `GQFormer-BASE`, we also note the hyperparameters defining the length of input time series reconstruction as auxiliary task.

[8] github.com/hihihihiwsf/AST

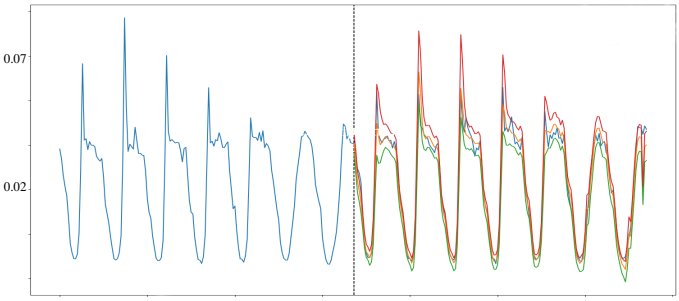

Fig. 2: Quantile forecasting for the $\texttt{traffic}_{168}$ task.

For the direct forecasting task, where the task was to generate 168 forecast horizons, we reconstructed all the input of 168 observations (as noted in Table.I), and for the rolling forecasting task, we reconstructed double the number of forecast horizons, 48 input observations where the task was to forecast the next 24. Notably, in fairness to other baselines as we describe below, we did not tune the number of shared fully connected layers for quantile generation and neither Transposed Quantile Attention layers, which were both kept fixed to 1. Additionally, we did not tune the weights for the loss components Energy score based approximation CRPS loss and the quantile loss approximated CRPS loss. Notably, these hyperparameters and length of reconstruction per forecasting task, further tuned, could possibly improve the model performance.

We also describe the hyperparameter tuning details for the RNN based baselines `DeepAR`[9], `SQF-RNN`[9],`IQN-RNN`[10] and `MQ-RNN`[9]. In our experiments, the number of RNN layers $[4, 8]$, and the cell sizes $[256, 512]$ hyperparameters were tuned. Similar to before, each configuration was run for 3 seeds with learning rates sampled uniformly at random. Given that the RNN baselines required less computational time compared to the Transformer baselines, we scheduled the corresponding experiments to 40 epochs which ensured a computationally fair budget. The Variational autoencoder baselines `CVAE MSE`, `CVAE DIL`, `MCVAE MSE` are also based on RNNs. Therefore, we tuned the hyperparameters of RNN layers and the cell sizes similar to the other RNN baselines as described earlier. Moreover, for these baselines we additionally tuned the dimensionality of the fully-connected layer $[512, 1024]$ for the forecast output. Notably, this hyperparameter was not tuned for the RNN based baselines earlier described and kept fixed as the cell-size hyperparameter. The variational models were also trained for 40 epochs. Once the first-stage optimization of the `CVAE DIL` model was completed, we optimized for the time and shape loss concerning the `STRIPE DIL` model[11].

B Qualitative Results

We showcase qualitative results for the Traffic dataset using the `GQFormer` model for different quantile levels in Fig. 2.

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