QUEST: QUANTILE-CONDITIONED SELECTION OF HISTORICAL CONTEXT FOR PROBABILISTIC TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Probabilistic time series forecasting (PTSF) plays an important role in decision-making across a wide range of real-world applications, including economics, transportation, energy, and AIOps. Recent direct prediction methods offer an efficient way to produce probabilistic forecasts without assuming a predefined distribution family or relying on iterative sampling. However, existing methods often construct a shared historical context for different parts of the predictive distribution, while the relevance of historical information may vary across quantile levels. This motivates historical context construction that explicitly accounts for the target probability level. In this paper, we propose QUEST, a probabilistic time series forecasting model based on quantile-conditioned historical selection. Specifically, QUEST divides the observed history into patches and encodes them into shared local representations. For each target quantile, a corresponding query interacts with these representations to determine patch weights conditioned on both the observed history and the target probability level, yielding a quantile-specific predictive context. A shared decoder then combines each context with its corresponding query to predict the target quantile over the entire forecast horizon, enabling multiple quantile forecasts in a single forward pass. Extensive experiments on eight real-world multivariate datasets demonstrate that QUEST achieves state-of-the-art probabilistic forecasting performance. Code and data are available at `https://anonymous.4open.science/r/QUEST-95D1/`.

1 INTRODUCTION

Time series forecasting provides important support for decision-making in economics, transportation, energy, and AIOps (Qiu et al., 2024; 2025; Wu et al., 2025a;b). Compared with point forecasting, which provides only a single estimate of future observations, probabilistic time series forecasting (PTSF) characterizes the conditional distribution of future observations, describing both possible outcomes and their associated uncertainty, thereby providing a basis for constructing prediction intervals and assessing risk (Gneiting & Katzfuss, 2014; Hong & Fan, 2016).

In recent years, advances in deep learning have led to multiple modeling approaches for probabilistic time series forecasting (Lim & Zohren, 2021). Figure 1 illustrates three representative paradigms: parametric methods, generative methods, and direct prediction methods. Among them, parametric methods estimate the parameters of predefined distributions, such as Gaussian distributions, from historical observations, but their expressiveness is constrained by the predefined distribution family (Salinas et al., 2020; Rangapuram et al., 2018). Generative methods learn a generative process to flexibly characterize the future distribution, but typically require multiple iterations to generate forecast samples, introducing additional computational overhead (Rasul et al., 2021a; Tashiro et al., 2021; Kollovieh et al., 2023; Liu et al., 2025). Recently, an increasing number of methods have adopted the direct prediction paradigm (Liang, 2025; Shen et al., 2025; Wang et al., 2026; Dai et al., 2026). These methods directly output quantile forecasts at different probability levels, ensembles of future values, or discrete predictive distributions from historical observations, without requiring a predefined distribution family or multi-step iterative sampling.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

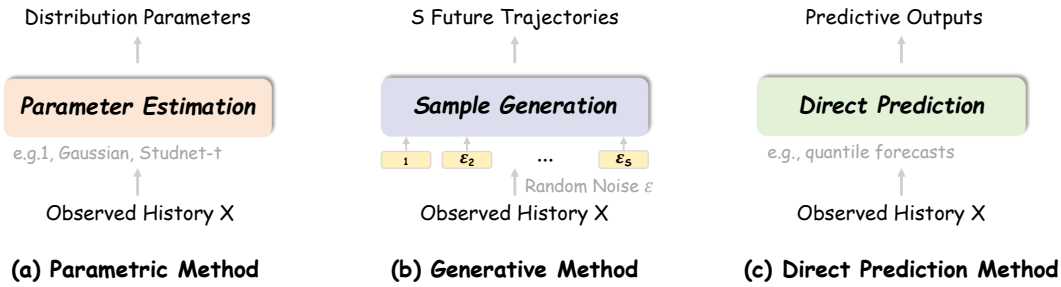


Figure 1: Three paradigms for probabilistic time series forecasting.

In the direct prediction paradigm, existing methods often encode historical observations into a shared predictive context and then produce probabilistic forecasts through output mappings (Wen et al., 2017; Shen et al., 2025). However, different parts of the future distribution may depend on historical information in different ways. (Zhu et al., 2023; Chavleishvili & Manganelli, 2024) For example, in daily maximum temperature forecasting, the degree of dependence on the previous day's observation differs between central and extreme quantiles (Castillo-Mateo et al., 2024). Aggregating historical information into a shared context means that a single representation must simultaneously retain the information needed to characterize different parts of the distribution. Although output mappings can use this information in different ways, the historical aggregation process itself does not explicitly account for these differences in forecasting requirements. These differences suggest explicitly accounting for the target part of the predictive distribution when selecting and aggregating historical information. For quantile forecasting, the target quantile specifies the position of interest in the predictive distribution at each future time step and can serve as a condition for historical selection. This motivates a quantile-conditioned historical context selection mechanism to accommodate the forecasting requirements of different quantiles.

Motivated by these observations, we propose QUEST, a probabilistic time series forecasting model built around quantile-conditioned historical selection. Specifically, QUEST first divides the observed history into patches and encodes them with a shared MLP encoder to extract local historical representations as the basic units for subsequent selection. To adapt the selection of historical information to different probability levels, we learn a corresponding query for each target quantile and let the query interact with the patch representations to determine the relative weight of each patch. The resulting weights depend on both the observed history and the target quantile, allowing the same historical pattern to contribute to different quantile forecasts to different extents. The weighted representations are then aggregated into quantile-specific contexts, and a shared decoder combines each context with the corresponding quantile query to output the target quantile at every time step over the entire forecast horizon. Our main contributions are summarized as follows:

- We propose QUEST, a probabilistic time series forecasting method that separates shared local representation learning from quantile-specific context construction, enabling direct prediction of multiple quantiles over the entire forecast horizon in a single forward pass.
- We design a quantile-conditioned historical selection mechanism in which quantile queries interact with shared patch representations to determine aggregation weights jointly from historical observations and target quantiles, forming quantile-specific predictive contexts.
- Extensive experiments on eight real-world multivariate datasets across four forecasting horizons against ten probabilistic forecasting baselines demonstrate that QUEST achieves best or tied-best horizon-averaged CRPS and NMAE on seven datasets each.

2 Related Work

2.1 Probabilistic Time Series Forecasting

Probabilistic time series forecasting (PTSF) plays an important role in decision- making across a wide range of real-world applications. Early deep learning methods combined temporal modeling with probabilistic estimation. DeepAR (Salinas et al., 2020) used recurrent networks to parameterize

observation distributions, while DeepState (Rangapuram et al., 2018) integrated recurrent networks with state space models. Attention-based models further enriched temporal dependency modeling: ProTran (Tang & Matteson, 2021) introduced attention into hierarchical latent-variable models, and MANF (Feng et al., 2023) combined multi-scale attention with conditional normalizing flows. Diffusion-based approaches, including TimeGrad (Rasul et al., 2021a) and TSDiff (Kollovieh et al., 2023), learn predictive distributions through denoising processes. Recent research further investigates generative conditions and priors. CN-Diff (Rishi et al., 2025) incorporates nonlinear transformations and learnable conditions into forward diffusion, while TSFlow (Kollovieh et al., 2025) employs Gaussian process priors within conditional flow matching to capture temporal structure. CW-Gen (Yang et al., 2026) introduces conditional whitening using estimated means and covariances, and KITE (Cheng et al., 2026) combines history-conditioned initialization with knowledge-guided exogenous conditioning. Alternatively, K^2VAE (Wu et al., 2025b) combines variational modeling with Koopman dynamics and Kalman-based uncertainty refinement.

Direct prediction methods map historical observations to quantile forecasts, prediction ensembles, or discrete predictive distributions. MQ-RNN (Wen et al., 2017) and Temporal Fusion Transformers (Lim et al., 2021) jointly estimate multiple conditional quantiles across forecast horizons. AnyQuantile (Smyl et al., 2026) studies flexible quantile evaluation, enabling a single trained model to produce forecasts at arbitrary probability levels. Beyond quantile outputs, DistPred (Liang, 2025) produces prediction ensembles in a single forward pass, OCE-TS (Wang et al., 2026) estimates probabilities over ordered value intervals, and TimePrism (Dai et al., 2026) directly outputs future scenarios together with their probabilities. These methods offer diverse representations of predictive uncertainty. However, specifying the predictive distribution or quantile does not by itself determine how historical information should be selected and aggregated for different probability levels.

2.2 Historical Context Modeling for Time Series Forecasting

Historical context modeling concerns how past observations are represented and integrated to support future prediction. Existing methods differ in both the granularity of historical representations and the mechanisms used to select and combine them. Patch-based approaches group adjacent observations into local segments, enabling temporal modeling at the subseries level. PatchTST (Nie et al., 2023) embeds these segments as tokens and uses a channel-independent Transformer to capture dependencies across patches. PITS (Lee et al., 2024) explores independent patch representation learning through patch-wise MLP encoding and reconstruction, complemented by contrastive learning. These approaches illustrate different ways to learn local historical features and incorporate relationships among temporal segments.

The construction of forecasting contexts also depends on which historical patterns are emphasized and how they are organized. Pathformer (Chen et al., 2024) combines multiple patch sizes with intra- and inter-patch attention, using adaptive pathways to adjust multi-scale modeling according to the input. SRSNet (Wu et al., 2025a) learns to select and reorder candidate patches, constructing a selective representation space for subsequent prediction. Learned queries provide another mechanism for integrating historical information. TQNet (Lin et al., 2025) uses periodically shifted learnable queries with input-derived keys and values to capture inter-variable correlations. For irregularly sampled time series, APN (Liu et al., 2026) learns temporal windows and applies time-aware weighting to construct patch representations. It then uses a learnable query for each channel to summarize these representations into a historical context, which is combined with the prediction-time embedding during decoding. These methods adapt historical representations to temporal structure, input content, or variable-specific requirements. Explicitly incorporating the target probability level into historical context construction remains a distinct modeling consideration.

3 Preliminaries

Problem formulation. Let $\mathbf{X} \in \mathbb{R}^{L \times C}$ denote an observed history of L time steps and C channels, and let $\mathbf{Y} \in \mathbb{R}^{H \times C}$ denote the future observations over a forecasting horizon of length H. We consider probabilistic forecasting through direct prediction of multiple conditional quantiles. For a fixed grid of probability levels $\mathcal{T} = \{\tau_q\}_{q=1}^{Q}$ satisfying $0 < \tau_1 < \cdots < \tau_Q < 1$, a model with parameters θ produces

$$\widehat{\mathbf{Y}} = f_\theta(\mathbf{X}; \mathcal{T}) \in \mathbb{R}^{H \times C \times Q}, \tag{1}$$

where $\widehat{Y}_{h,c,q}$ estimates the conditional τ_q-quantile of $Y_{h,c}$ for future time step h and channel c. In QUEST, each channel's forecast is conditioned on its own history $\mathbf{x}_c = \mathbf{X}_{:,c} \in \mathbb{R}^L$.

Quantile regression. Conditional quantiles can be estimated using the pinball loss (Koenker & Bassett, 1978). For an observation y and a prediction $\widehat{y}$, define the residual as $e = y - \widehat{y}$. The loss at probability level τ is

$$\rho_\tau(e) = \max\{\tau e, (\tau - 1)e\}. \tag{2}$$

This loss weights underprediction by τ and overprediction by $1 - \tau$. Applying this loss at each target level provides supervision for direct multi-quantile forecasting without specifying a parametric distribution family.

4 METHODOLOGY

Given an observed history $\mathbf{X} \in \mathbb{R}^{L \times C}$ and a set of target quantile levels $\mathcal{T}$, QUEST predicts multiple quantiles over a horizon of length H using quantile-specific historical contexts. As shown in Figure 2, the model consists of three components. Local Patch Encoding independently maps historical patches to local representations shared across all target quantiles. Quantile-Conditioned Historical Selection maps each target probability level to a query and uses it to weight and aggregate these representations into a quantile-specific context. The Forecast Decoder combines each context with its query to predict the corresponding quantile at all H future time steps. Channels are processed independently, with model parameters shared across channels.

4.1 LOCAL PATCH ENCODING

Local Patch Encoding extracts representations of historical segments as the basic units for subsequent selection. As shown in Figure 2(a), we first normalize the history of each channel using instance normalization (Kim et al., 2022):

$$\widetilde{X}_{t,c} = \frac{X_{t,c} - \mu_c}{\sigma_c}. \tag{3}$$

Here, μ_c and σ_c are the mean and stabilized standard deviation of channel c over the observed history. These statistics are retained to restore the scale of the predictions. We then divide the normalized history into patches of length K with stride S (Nie et al., 2023):

$$\begin{aligned} \mathbf{p}_{c,m} &= \widetilde{\mathbf{X}}_{(m-1)S+1:(m-1)S+K,c} \in \mathbb{R}^K, \\ M &= \left\lfloor \frac{L-K}{S} \right\rfloor + 1, \qquad m = 1, \ldots, M. \end{aligned} \tag{4}$$

Here, M denotes the number of complete patches, with $1 \le S \le K \le L$. We encode patches independently, leaving the aggregation of their representations to quantile-conditioned selection. Specifically, we apply a shared linear projection and an MLP encoder to each patch (Lee et al., 2024), then add its positional encoding:

$$\mathbf{z}_{c,m} = f_{\mathrm{loc}}(\mathbf{W}_p \mathbf{p}_{c,m} + \mathbf{b}_p) + \mathbf{r}_m \in \mathbb{R}^d, \tag{5}$$

where $\mathbf{W}_p \in \mathbb{R}^{d \times K}$ and $\mathbf{r}_m$ is a fixed sinusoidal positional encoding (Vaswani et al., 2017). The local encoder consists of E residual blocks, each applying a linear transformation, GELU, and dropout, followed by residual addition and layer normalization. The resulting patch representations $\mathbf{Z}_c = [\mathbf{z}_{c,1}, \ldots, \mathbf{z}_{c,M}]^\top \in \mathbb{R}^{M \times d}$ contain local content and positional information for historical selection. This set of representations is shared by all target quantiles.

4.2 QUANTILE-CONDITIONED HISTORICAL SELECTION

To select historical information for different target quantiles, we construct a context from $\mathbf{Z}_c$ conditioned on each probability level. This module consists of two steps: Quantile Query Encoding maps the probability level to a query, and Quantile-Conditioned Patch Aggregation uses the query to combine the local patch representations.

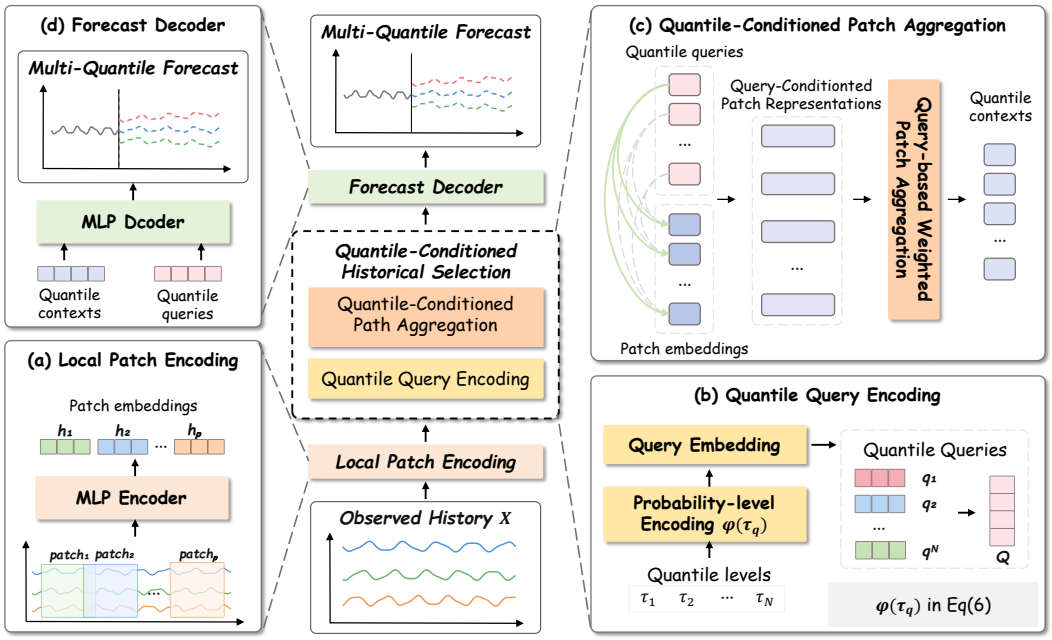


Figure 2: The architecture of QUEST. (a) Local Patch Encoding maps historical patches to shared local representations. (b) Quantile Query Encoding constructs a query for each target probability level. (c) Quantile-Conditioned Patch Aggregation uses these queries to weight the patch representations and obtain a historical context for each quantile. Components (b) and (c) form Quantile-Conditioned Historical Selection. (d) The Forecast Decoder combines each context with its query and outputs the corresponding quantile over the entire forecast horizon.

4.2.1 Quantile Query Encoding

Quantile Query Encoding provides a query for each target probability level to guide the selection of historical patches. As shown in Figure 2(b), for τ_q in the fixed grid $\mathcal{T} = \{\tau_q\}_{q=1}^{Q}$, we first combine its numerical value with cosine features (Dabney et al., 2018):

$$\phi(\tau_q) = [\tau_q, \cos(\pi\tau_q), \ldots, \cos(F\pi\tau_q)]^\top \in \mathbb{R}^{F+1}, \tag{6}$$

where F is the number of cosine features. A shared two-layer MLP with a GELU activation between the layers maps these probability-level features to a query. To allow each targetquantile to adjust the query produced by this shared mapping, we add a quantile-specific learned residual:

$$\mathbf{e}_q = g_{\text{qry}}(\phi(\tau_q)) + \alpha \mathbf{v}_q, \tag{7}$$

where $g_{\text{qry}} : \mathbb{R}^{F+1} \to \mathbb{R}^d$ is the MLP, $\mathbf{v}_q \in \mathbb{R}^d$ is the residual embedding for τ_q, and $\alpha \geq 0$ is a fixed scaling hyperparameter. We initialize $\mathbf{v}_q$ to zero and optimize it jointly with g_{qry}. The resulting queries $\mathbf{e}qq = 1^Q$ are shared across input histories and channels and used to effectively aggregate relevant historical patches.

4.2.2 Quantile-Conditioned Patch Aggregation

After obtaining the quantile queries, we use them to determine the contributions of historical patches, as illustrated in Figure 2(c). For channel c and target level τ_q, we score each patch by the scaled dot product of its representation and the query (Vaswani et al., 2017). The scores are normalized over all M patches using softmax:

$$s_{c,q,m} = \frac{\mathbf{e}_q^\top \mathbf{z}_{c,m}}{\sqrt{d}}, \tag{8}$$

$$a_{c,q,m} = \frac{\exp(s_{c,q,m})}{\sum_{j=1}^{M} \exp(s_{c,q,j})}, \tag{9}$$

Here, $s_{c,q,m}$ measures the relevance of patch m to query $\mathbf{e}_q$, and $a_{c,q,m}$ is its aggregation weight. We then take the weighted sum of the patch representations to obtain the historical context for τ_q:

$$\mathbf{h}_{c,q} = \sum_{m=1}^{M} a_{c,q,m} \mathbf{z}_{c,m} \in \mathbb{R}^d. \tag{10}$$

The weights are determined jointly by the historical representations and the quantile query. Thus, the same historical patch can make different contributions to different quantile forecasts. Repeating this aggregation for all queries gives Q quantile-specific contexts per channel, which are passed to the decoder together with their corresponding queries.

4.3 Forecast Decoder

The Forecast Decoder uses $\mathbf{h}_{c,q}$ as the historical context and $\mathbf{e}_q$ to explicitly condition the prediction on the target probability level. As shown in Figure 2(d), we concatenate the two representations and use a shared MLP to predict the corresponding quantile over all H future time steps:

$$\widehat{\widetilde{\mathbf{y}}}_{c,q} = g_{\text{dec}}([\mathbf{h}_{c,q}; \mathbf{e}_q]) \in \mathbb{R}^H, \tag{11}$$

where $[\cdot;\cdot]$ denotes concatenation and $g_{\text{dec}} : \mathbb{R}^{2d} \to \mathbb{R}^H$ is a two-layer MLP with GELU activation. Its parameters are shared across channels and quantiles. The context provides the aggregated historical information, while the query supplies an explicit condition on the target probability level. This gives the shared decoder direct access to the quantile condition in addition to its effect on historical aggregation. Each context is used to predict its target quantile over the entire horizon. Finally, we restore the predictions to the input scale using the statistics retained during normalization:

$$\widehat{Y}_{h,c,q} = \sigma_c \widehat{\widetilde{Y}}_{h,c,q} + \mu_c, \tag{12}$$

where $\widehat{\widetilde{Y}}_{h,c,q}$ is the h-th component of $\widehat{\widetilde{\mathbf{y}}}_{c,q}$. The outputs from all channel–quantile pairs form $\widehat{\mathbf{Y}} \in \mathbb{R}^{H \times C \times Q}$.

4.4 Training and Inference

Training objective. We optimize QUEST using the pinball loss (Koenker & Bassett, 1978) defined in Equation 2. For a minibatch of size B, let $O_{b,h,c} \in \{0,1\}$ indicate whether a target is observed. We minimize

$$\mathcal{L}_{\text{Q}} = \frac{\sum_{b,h,c} \sum_{q=1}^{Q} O_{b,h,c}\, \rho_{\tau_q}\left(Y_{b,h,c} - \widehat{Y}_{b,h,c,q}\right)}{Q \max\{1, \sum_{b,h,c} O_{b,h,c}\}}. \tag{13}$$

This objective averages the quantile losses over observed target coordinates and jointly trains the patch encoder, quantile queries, and forecast decoder. The pinball loss is computed after reversing instance normalization, on the training-set standardized scale. Predictions are restored to the original data scale for evaluation.

Direct inference. QUEST predicts all targetquantiles in a single forward pass. Since monotonicity across quantile levels is not explicitly enforced, we sort the predictions at each future coordinate to obtain nondecreasing quantile forecasts. For benchmark evaluation, we then interpolate these sorted predictions on a fixed probability grid to obtain deterministic marginal forecast values. The complete procedure, including endpoint clipping, is given in Appendix A.2.

5 Experiments

5.1 Experimental Setup

Datasets. We conduct experiments on eight real-world datasets based on ProbTS (Zhang et al., 2024), a comprehensive benchmark for probabilistic forecasting. We follow the data splitting and preprocessing procedures of ProbTS. For the four ETT datasets, the first 12, 4, and 4 months are used for training, validation, and testing, respectively, following the benchmark's 30-day month

Table 1: Dataset Summary.

Dataset	Variables	Frequency	Timesteps	Description
ETTh1 / ETTh2	7	Hourly	17,420	Electricity transformer temperature per hour
ETTm1 / ETTm2	7	15 min	69,680	Electricity transformer temperature every 15 min
Electricity	321	Hourly	26,304	Electricity consumption (KWh)
Traffic	862	Hourly	17,544	Road occupancy rates
Weather	21	10 min	52,696	Local climatological data
ILI	7	Weekly	966	Ratio of patients seen with influenza-like illness

convention. The remaining datasets are divided chronologically into training, validation, and test sets with a ratio of approximately $7:1:2$. Each variable is standardized using training-set statistics.

The context length is 96, with forecasting horizons of $\{96, 192, 336, 720\}$, except for ILI, which uses a context length of 36 and horizons of $\{24, 36, 48, 60\}$. This yields 32 dataset–horizon settings. Predictions are restored to the original data scale for evaluation. Dataset statistics are summarized in Table 1, and more detailed descriptions of the datasets and corresponding forecasting settings are provided in Appendix A.1.

Implementation Details. We implement QUEST in PyTorch with Python 3.10 and conduct experiments on an NVIDIA Tesla-A800. We train the model with Adam by minimizing the pinball loss over 19 quantile levels, $\mathcal{T} = \{0.05, 0.10, \ldots, 0.95\}$. The checkpoint with the lowest validation CRPS is selected for testing. Each experiment is repeated with five different random seeds, and we report the mean and sample standard deviation in the full results. Additional implementation details are provided in Appendix A.3. To ensure a fair comparison, we do not apply the "Drop Last" trick during testing (Qiu et al., 2024).

Baselines. We compare QUEST with ten probabilistic forecasting baselines, including GRU-MAF, GRU-NVP, Trans-MAF (Rasul et al., 2021b), TimeGrad (Rasul et al., 2021a), CSDI (Tashiro et al., 2021), TSDiff (Kollovieh et al., 2023), K^{2}VAE (Wu et al., 2025b), DistPred (Liang, 2025), OCE-TS (Wang et al., 2026), and PPM (Li et al., 2026). These baselines cover a diverse range of probabilistic forecasting paradigms, including normalizing flows, diffusion-based models, variational approaches, and recent distributional forecasting methods. This broad comparison allows us to evaluate QUEST against representative methods with different probabilistic modeling strategies.

Evaluation Metrics. We use two commonly used metrics, CRPS (Continuous Ranked Probability Score) and NMAE (Normalized Mean Absolute Error), to evaluate the probabilistic forecasts. CRPS measures the overall quality of the predicted distribution by comparing it with the observed outcome, while NMAE evaluates the accuracy of the resulting point estimates under normalization. Together, they provide complementary assessments of distributional quality and predictive accuracy. Detailed descriptions of these metrics can be found in Appendix A.2.

5.2 Main Results

Probabilistic forecasting results are listed in Table 2, with the best in red and the second-best in blue. Each entry is averaged over the four forecasting horizons, and the full results are provided in Appendix B.1.

We have the following observations: 1) QUEST achieves strong overall forecasting performance, ranking first on seven of the eight datasets in CRPS and seven in NMAE, including ties. It ranks among the top two on every dataset for each metric. 2) The improvements are particularly pronounced on Weather and Traffic. Compared with the strongest baseline for each dataset and metric, QUEST reduces average CRPS by 11.8% and 8.0%, and average NMAE by 7.5% and 7.3%, respectively. These gains demonstrate improvements in both probabilistic forecasts and their predictive medians on the two datasets. QUEST also reduces average CRPS by 5.7% on ETTm1 and 4.8% on ILI, extending the gains to energy and epidemiological forecasting. 3) QUEST performs consistently across different forecasting horizons. It achieves the lowest CRPS in 26 of the 32 dataset–horizon settings and the lowest NMAE in 26 settings. In particular, it obtains the best results for both metrics at all four horizons on ETTm1, ETTh2, Traffic, and Weather, demonstrating strong forecasting performance across datasets and prediction lengths.

Table 2: CRPS and NMAE averaged over four forecasting horizons on each dataset (lower is better). Each entry is the equal-weight average of the four horizon-level means, rounded to three decimal places. **Red**: best; blue: second-best. Highlighting and first-place counts use the displayed precision; tied best scores each count as a first place. Full results are provided in Appendix B.1.

Models	QUEST (ours)		PPM		OCE-TS		K^2VAE		DistPred		TSDiff		CSDI		TimeGrad		GRU-MAF		Trans-MAF		GRU-NVP	
Metrics	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE
ETTm1	**0.247**	**0.317**	0.275	0.322	0.336	0.435	0.262	0.328	0.291	0.355	0.407	0.519	0.324	0.421	0.587	0.736	0.412	0.528	0.477	0.605	0.453	0.585
ETTm2	**0.137**	**0.173**	0.145	0.177	0.140	**0.173**	0.165	0.194	**0.137**	**0.173**	0.276	0.338	0.173	0.222	0.448	0.546	0.309	0.377	0.272	0.339	0.414	0.542
ETTh1	**0.271**	0.361	0.299	0.364	0.283	0.371	0.294	0.375	0.278	**0.356**	0.457	0.586	0.479	0.603	0.502	0.651	0.353	0.440	0.355	0.453	0.441	0.559
ETTh2	**0.156**	**0.201**	0.172	0.211	0.160	0.206	0.221	0.236	0.159	0.205	0.339	0.417	0.242	0.311	0.435	0.545	0.480	0.554	0.282	0.358	0.597	0.736
Electricity	0.076	**0.097**	0.085	0.107	0.097	0.123	**0.066**	0.105	0.083	0.107	0.407	0.519	-	-	0.102	0.126	0.094	0.122	-	-	0.101	0.127
Traffic	**0.177**	**0.221**	0.222	0.280	0.244	0.306	0.192	0.239	0.222	0.279	0.327	0.392	-	-	0.211	0.246	-	-	-	-	0.198	0.245
Weather	**0.062**	**0.079**	0.084	0.101	0.070	0.086	0.081	0.090	0.077	0.096	0.132	0.134	0.077	0.093	0.125	0.155	0.133	0.165	0.123	0.154	0.119	0.147
ILI	**0.106**	**0.133**	0.147	0.164	0.116	0.139	0.117	0.144	0.112	0.144	0.248	0.259	0.276	0.290	0.284	0.310	0.129	0.161	0.124	0.151	0.283	0.309
1^{st} Count	**7**	**7**	0	0	0	1	1	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0

Due to the excessive time and memory consumption, some results are unavailable and denoted as -.

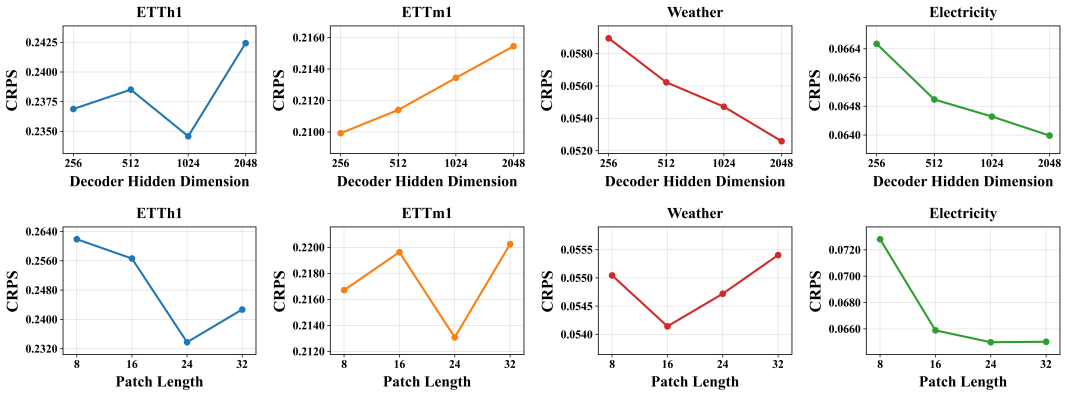


Figure 3: Hyperparameter sensitivity of QUEST. The top row varies the decoder hidden dimension, and the bottom row varies the patch length on ETTh1, ETTm1, Weather, and Electricity. All results use a prediction length of 96 and one random seed. Each panel has its own vertical scale; lower CRPS is better.

5.3 Model Analysis

5.3.1 Hyperparameter Sensitivity

Figure 3 examines how decoder capacity and patch configuration affect probabilistic forecasting performance. The effect of decoder hidden dimension differs across datasets. On ETTh1, CRPS changes non-monotonically and reaches its minimum of 0.235 at a dimension of 1024, before increasing to 0.242 at 2048. ETTm1 favors the smallest tested dimension: its CRPS rises from 0.210 at 256 to 0.215 at 2048. In contrast, Weather and Electricity benefit from wider decoders throughout the tested range. Their CRPS decreases from 0.059 to 0.053 and from 0.067 to 0.064, respectively. Hence, increasing decoder capacity does not yield a consistent improvement across datasets.

The patch configurations also exhibit distinct preferences. On ETTh1, increasing patch length from 8 to 24 reduces CRPS from 0.262 to 0.234, whereas increasing it further to 32 raises CRPS to 0.243. ETTm1 likewise obtains its lowest CRPS at a patch length of 24 (0.213). Weather instead favors a length of 16 (0.054). For Electricity, the largest improvement occurs between lengths 8 and 16, with CRPS decreasing from 0.073 to 0.066; lengths 24 and 32 both yield approximately 0.065 after rounding. Overall, these results show that the preferred decoder width and patch configuration depend on the characteristics of each dataset, and that simply increasing either hyperparameter does not necessarily guarantee better CRPS performance across different settings.

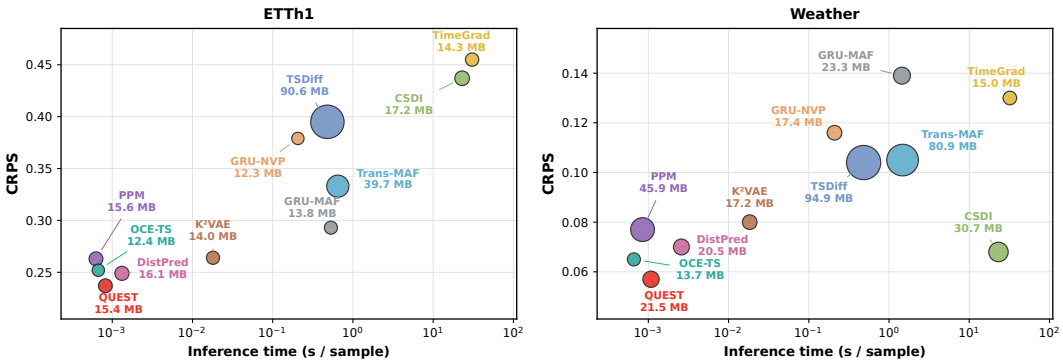


Figure 4: Model efficiency comparison on ETTh1 and Weather. The context length and forecast horizon are both 96, with batch size 1. The horizontal axis shows inference time on a logarithmic scale, where each sample denotes one multivariate input window. Bubble area is proportional to peak allocated GPU memory, with the same scale used across both panels. Lower CRPS and inference time indicate better performance.

5.4 Efficiency Analysis

We evaluate model efficiency from three perspectives: probabilistic forecasting accuracy measured by CRPS, inference time per sample, and peak allocated GPU memory. Figure 4 compares QUEST with ten baselines on ETTh1 and Weather, with both the context length and forecast horizon set to 96 and a batch size of one. Each sample corresponds to one multivariate input window, and bubble area represents peak allocated GPU memory.

QUEST achieves the lowest CRPS among the compared methods on both datasets while maintaining millisecond-level inference latency. Specifically, it requires only 0.82 ms per sample on ETTh1 and 1.08 ms on Weather, corresponding to approximately $22\times$ and $17\times$ faster inference than K^2VAE, respectively. Its peak allocated GPU memory is 15.4 MB on ETTh1 and 21.5 MB on Weather. These results show that QUEST combines accurate probabilistic forecasts with low latency and modest memory requirements. Although PPM and OCE-TS achieve lower inference latency, QUEST provides better forecasting accuracy on both datasets, offering a competitive trade-off between predictive performance and computational cost.

This efficiency is consistent with QUEST's direct quantile prediction architecture. QUEST encodes historical patches once and reuses these representations across quantile queries to produce all target quantiles in a single forward pass. The shared decoder generates predictions for the entire forecast horizon without autoregressive rollout. Consequently, QUEST requires neither explicit trajectory sampling nor computationally expensive iterative denoising, supporting flexible quantile-specific predictions with a compact and efficient inference procedure.

6 Conclusion

In this paper, we proposed QUEST, a probabilistic time series forecasting model that incorporates the target quantile into historical context construction. By combining shared local patch encoding with quantile-conditioned historical selection, QUEST allows different quantiles to aggregate information from the same observed history with different weights. A shared decoder then produces multi-quantile forecasts over the entire prediction horizon in a single forward pass. Experiments on eight real-world datasets across four forecasting horizons demonstrate strong performance against ten probabilistic forecasting baselines. For both CRPS and NMAE, QUEST achieves the best or tied-best horizon-averaged results on seven of the eight datasets. These results support quantile-conditioned historical context construction as a promising approach to probabilistic time series forecasting. Future work will explore extending this approach to model dependencies across variables and future time steps.

6.1 AI USE STATEMENT

We used ChatGPT to help polish the manuscript and improve the clarity of the writing. All methodological design and experimental analyses were conducted independently by the authors, and the use of AI was limited to auxiliary functions. The authors take full responsibility for the final content of this work.

6.2 ETHICS STATEMENT

This work focuses on methodological research for probabilistic time series forecasting and does not involve human subjects or the collection of personal data. All experiments are conducted on existing benchmark datasets following established evaluation protocols. We do not release any new dataset containing sensitive or personally identifiable information. The proposed method is intended for general-purpose forecasting research, and we are not aware of any direct ethical risks arising from the methodology itself.

6.3 REPRODUCIBILITY STATEMENT

The QUEST architecture and training procedure are described in Section 4, and the experimental settings are provided in Section 5.1. Dataset descriptions and implementation details are provided in Appendix A. Code for the main experiments is available at `https://anonymous.4open.science/r/QUEST-95D1/`.

REFERENCES

Jorge Castillo-Mateo, Alan E. Gelfand, Jesús Asín, Ana C. Cebrián, and Jesús Abaurrea. Bayesian joint quantile autoregression. *TEST*, 33(1):335–357, 2024.

Sulkhan Chavleishvili and Simone Manganelli. Forecasting and stress testing with quantile vector autoregression. *Journal of Applied Econometrics*, 39(1):66–85, 2024.

Peng Chen, Yingying Zhang, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. In *ICLR*, pp. 10527–10545, 2024.

Hanyin Cheng, Jingrong Zhou, Yang Shu, and Chenjuan Guo. KITE: Knowledge-guided probabilistic modeling for time series forecasting with exogenous variables. In *ICML*, 2026.

Will Dabney, Georg Ostrovski, David Silver, and Remi Munos. Implicit quantile networks for distributional reinforcement learning. In *ICML*, pp. 1096–1105, 2018.

Xilin Dai, Zhijian Xu, Wanxu Cai, and Qiang Xu. From samples to scenarios: A new paradigm for probabilistic forecasting. In *ICLR*, 2026.

Shibo Feng, Chunyan Miao, Ke Xu, Jiaxiang Wu, Pengcheng Wu, Yang Zhang, and Peilin Zhao. Multi-scale attention flow for probabilistic time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36(5):2056–2068, 2023.

Tilmann Gneiting and Matthias Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1(1):125–151, 2014.

Tao Hong and Shu Fan. Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*, 32(3):914–938, 2016.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *ICLR*, 2022.

Roger Koenker and Gilbert Bassett, Jr. Regression quantiles. *Econometrica*, 46(1):33–50, 1978.

Marcel Kollovieh, Abdul Fatir Ansari, Michael Bohlke-Schneider, Jasper Zschiegner, Hao Wang, and Yuyang (Bernie) Wang. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. In *NeurIPS*, pp. 28341–28364, 2023.

Marcel Kollovieh, Marten Lienen, David Lüdke, Leo Schwinn, and Stephan Günnemann. Flow matching with Gaussian process priors for probabilistic time series forecasting. In *ICLR*, 2025.

Seunghan Lee, Taeyoung Park, and Kibok Lee. Learning to embed time series patches independently. In *ICLR*, pp. 27599–27620, 2024.

Jinglin Li, Jun Tan, Qi Fang, and Ning Gui. Parametric prior mapping framework for non-stationary probabilistic time series forecasting. In *ICML*, 2026.

Daojun Liang. DistPred: A distribution-free probabilistic inference method for regression and forecasting. In *SIGKDD*, pp. 753–764, 2025.

Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

Bryan Lim, Sercan Ö. Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4): 1748–1764, 2021.

Shengsheng Lin, Haojun Chen, Haijie Wu, Chunyun Qiu, and Weiwei Lin. Temporal query network for efficient multivariate time series forecasting. In *ICML*, pp. 37797–37814, 2025.

Xvyuan Liu, Xiangfei Qiu, Xingjian Wu, Zhengyu Li, Chenjuan Guo, Jilin Hu, and Bin Yang. Rethinking irregular time series forecasting: A simple yet effective baseline. In *AAAI*, volume 40, pp. 23873–23881, 2026.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models. In *ICML*, pp. 39295–39317, 2025.

James E. Matheson and Robert L. Winkler. Scoring rules for continuous probability distributions. *Management Science*, 22(10):1087–1096, 1976.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *ICLR*, 2023.

Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, Zhenli Sheng, and Bin Yang. TFB: Towards comprehensive and fair benchmarking of time series forecasting methods. *Proceedings of the VLDB Endowment*, 17 (9):2363–2377, 2024.

Xiangfei Qiu, Xingjian Wu, Hanyin Cheng, Xvyuan Liu, Chenjuan Guo, Jilin Hu, and Bin Yang. DBLoss: Decomposition-based loss function for time series forecasting. In *NeurIPS*, pp. 27741–27768, 2025.

Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. In *NeurIPS*, 2018.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *ICML*, pp. 8857–8868, 2021a.

Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs M. Bergmann, and Roland Vollgraf. Multivariate probabilistic time series forecasting via conditioned normalizing flows. In *ICLR*, 2021b.

J Rishi, Gvs Mothish, and Deepak Subramani. Conditional diffusion model with nonlinear data transformation for time series forecasting. In *ICML*, pp. 51703–51723, 2025.

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

Lefei Shen, Mouxiang Chen, Xu Liu, Han Fu, Xiaoxue Ren, Jianling Sun, Zhuo Li, and Chenghao Liu. VisionTS++: Cross-modal time series foundation model with continual pre-trained vision backbones. *arXiv preprint arXiv:2508.04379*, 2025.

Slawek Smyl, Boris N. Oreshkin, Paweł Pełka, and Grzegorz Dudek. Any-quantile probabilistic forecasting of short-term electricity demand: Fusing uncertainties from diverse sources. *Information Fusion*, 127:103637, 2026.

Binh Tang and David S Matteson. Probabilistic transformer for time series analysis. In *NeurIPS*, pp. 23592–23608, 2021.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. In *NeurIPS*, pp. 24804–24816, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Jieting Wang, Huimei Shi, Feijiang Li, and Xiaolei Shang. Beyond MSE: Ordinal cross-entropy for probabilistic time series forecasting. In *AAAI*, volume 40, pp. 26269–26276, 2026.

Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. A multi-horizon quantile recurrent forecaster. 2017.

Xingjian Wu, Xiangfei Qiu, Hanyin Cheng, Zhengyu Li, Jilin Hu, Chenjuan Guo, and Bin Yang. Enhancing time series forecasting through selective representation spaces: A patch perspective. In *NeurIPS*, pp. 23328–23354, 2025a.

Xingjian Wu, Xiangfei Qiu, Hongfan Gao, Jilin Hu, Bin Yang, and Chenjuan Guo. k^2VAE: A koopman-kalman enhanced variational AutoEncoder for probabilistic time series forecasting. In *ICML*, pp. 67562–67583, 2025b.

Yanfeng Yang, Siwei Chen, Pingping Hu, Zhaotong Shen, Yingjie Zhang, Zhuoran Sun, Shuai Li, Ziqi Chen, and Kenji Fukumizu. Conditionally whitened generative models for probabilistic time series forecasting. In *ICLR*, pp. 53263–53298, 2026.

Jiawen Zhang, Xumeng Wen, Zhenwei Zhang, Shun Zheng, Jia Li, and Jiang Bian. ProbTS: Benchmarking point and distributional forecasting across diverse prediction horizons. In *NeurIPS*, pp. 48045–48082, 2024.

Qianqian Zhu, Songhua Tan, Yao Zheng, and Guodong Li. Quantile autoregressive conditional heteroscedasticity. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85 (4):1099–1127, 2023.

A Experimental Details

A.1 Datasets

We conduct experiments on eight real-world datasets based on ProbTS (Zhang et al., 2024): ETTh1, ETTh2, ETTm1, ETTm2, Electricity, Traffic, Weather, and ILI. The four ETT datasets record transformer temperature and load measurements, with hourly observations in ETTh1 and ETTh2 and 15-minute observations in ETTm1 and ETTm2. Electricity contains electricity consumption records, and Traffic records road occupancy rates. Weather contains meteorological observations, while ILI records influenza-like illness indicators. These datasets cover energy, transportation, meteorology, and public health.

We follow the data splitting and preprocessing procedures of ProbTS. For the four ETT datasets, the first 12, 4, and 4 months are used for training, validation, and testing, respectively, following the benchmark's 30-day month convention. The remaining datasets are divided chronologically into training, validation, and test sets with a ratio of approximately $7:1:2$.

We standardize each variable using statistics calculated from the training set and restore predictions to the original scale for evaluation. The context length is 96 and the forecasting horizons are $\{96, 192, 336, 720\}$, except for ILI, which uses a context length of 36 and horizons of $\{24, 36, 48, 60\}$. We retain the final incomplete batches during validation and testing so that all evaluation windows are included. In total, the experiments cover 32 dataset–horizon settings.

A.2 Evaluation Metrics

We use two commonly-used metrics NMAE (Normalized Mean Absolute Error) and CPRS (Continuous Ranked Probability Score) in ProbTS (Zhang et al., 2024) to evaluate the probabilistic forecasts.

Normalized Mean Absolute Error (NMAE). The Normalized Mean Absolute Error (NMAE) is a normalized version of the MAE, which is dimensionless and facilitates the comparability of the error magnitude across different datasets or scales. The mathematical representation of NMAE is given by:

$$\mathrm{NMAE} = \frac{\sum_{k=1}^{K}\sum_{t=1}^{T}|x_t^k - \widehat{x}_t^k|}{\sum_{k=1}^{K}\sum_{t=1}^{T}|x_t^k|}. \tag{14}$$

Here, K and T denote the number of variables and forecast steps, respectively ($K = C$ and $T = H$ in our forecasting notation). For our evaluation, $\widehat{x}_t^k$ is the predictive median.

Continuous Ranked Probability Score (CRPS). The Continuous Ranked Probability Score (CRPS) (Matheson & Winkler, 1976) quantifies the agreement between a cumulative distribution function (CDF) F and an observation x, represented as:

$$\mathrm{CRPS} = \int_{\mathbb{R}} \left(F(z) - \mathbb{I}\{x \le z\}\right)^2 dz, \tag{15}$$

where $\mathbb{I}\{x \le z\}$ denotes the indicator function, equating to one if $x \le z$ and zero otherwise.

Evaluation protocol. For evaluation with ProbTS (Zhang et al., 2024), QUEST converts its quantile predictions into $R = 100$ values at each future step and channel. During inference, we linearly interpolate the sorted predictions at midpoint levels $v_r = (r - \frac{1}{2})/R$, $r = 1, \ldots, R$. Levels outside $[\tau_1, \tau_Q]$ use the corresponding endpoint prediction. The resulting deterministic values describe marginal predictive distributions and do not represent jointly sampled trajectories. NMAE uses the median of these values, while the reported CRPS uses the benchmark's normalized quantile-loss approximation over $\{0.05, 0.10, \ldots, 0.95\}$. Both metrics are computed on the original data scale within each forecast window and then averaged across windows.

A.3 Implementation Details

All experiments are conducted using PyTorch in Python 3.10 and execute on an NVIDIA Tesla-A800 GPU. We use the ProbTS framework for data processing and evaluation. QUEST is trained

with the Adam optimizer using the pinball loss on $\mathcal{T} = \{0.05, 0.10, \ldots, 0.95\}$. The checkpoint with the lowest validation CRPS is selected for testing. Each experiment is repeated five times with different random seeds, and we report the mean and sample standard deviation.

In the instance-normalization implementation, the historical mean is detached from the computational graph, while the standard deviation remains differentiable. A small positive constant is added to the variance before taking its square root.

B Additional Experimental Results

B.1 Full Forecasting Results

Tables 3 and 4 provide the complete CRPS and NMAE results on the eight datasets. We report each forecasting horizon separately, with the mean and standard deviation over five independent runs. The main table summarizes these results by averaging the four horizon-level means with equal weight for each dataset.

Table 3: Results of CRPS (mean ± standard deviation) across horizons. The context length is 36 for ILI and 96 for the other datasets. Each experiment is repeated five times with different random seeds. Lower values indicate better performance. **Red**: the best, Blue: the 2nd best.

Dataset	Horizon	GRU MAF	Trans MAF	TSDiff	CSDI	TimeGrad	GRU NVP	K^2VAE	DistPred	OCE-TS	PPM	QUEST
ETTm1	96	0.295±0.055	0.313±0.045	0.344±0.050	0.236±0.006	0.522±0.105	0.383±0.053	0.232±0.010	0.264±0.005	0.313±0.004	0.238±0.003	**0.218**±0.005
	192	0.389±0.033	0.424±0.029	0.345±0.035	0.291±0.025	0.603±0.092	0.396±0.030	0.259±0.013	0.284±0.007	0.329±0.004	0.261±0.006	**0.239**±0.006
	336	0.429±0.021	0.481±0.019	0.462±0.043	0.322±0.033	0.601±0.028	0.486±0.032	0.262±0.030	0.296±0.004	0.339±0.004	0.282±0.005	**0.253**±0.007
	720	0.536±0.033	0.688±0.043	0.478±0.027	0.448±0.038	0.621±0.037	0.546±0.036	0.294±0.026	0.321±0.006	0.361±0.006	0.320±0.007	**0.277**±0.005
ETTm2	96	0.177±0.024	0.227±0.013	0.175±0.019	0.115±0.009	0.427±0.042	0.319±0.044	0.126±0.007	0.110±0.003	0.113±0.005	0.116±0.002	**0.106**±0.003
	192	0.411±0.026	0.253±0.037	0.255±0.029	0.147±0.008	0.424±0.061	0.326±0.025	0.148±0.009	0.129±0.003	0.131±0.002	0.136±0.002	**0.128**±0.004
	336	0.377±0.023	0.253±0.013	0.328±0.047	0.190±0.018	0.469±0.049	0.449±0.145	0.164±0.010	**0.144**±0.005	0.146±0.004	0.152±0.004	0.146±0.006
	720	0.272±0.029	0.355±0.043	0.344±0.046	0.239±0.035	0.470±0.054	0.561±0.273	0.221±0.023	**0.166**±0.008	0.168±0.007	0.175±0.006	0.167±0.007
ETTh1	96	0.293±0.037	0.333±0.045	0.395±0.052	0.437±0.018	0.455±0.046	0.379±0.030	0.264±0.020	0.249±0.006	0.252±0.006	0.263±0.007	**0.237**±0.006
	192	0.348±0.075	0.351±0.063	0.467±0.044	0.496±0.051	0.516±0.038	0.425±0.019	0.290±0.016	0.272±0.005	0.278±0.004	0.296±0.003	**0.264**±0.004
	336	0.377±0.026	0.371±0.031	0.450±0.027	0.454±0.025	0.512±0.026	0.458±0.054	0.308±0.021	0.291±0.006	0.295±0.003	0.312±0.003	**0.285**±0.006
	720	0.393±0.043	0.363±0.053	0.516±0.027	0.528±0.012	0.523±0.027	0.502±0.039	0.314±0.011	**0.298**±0.002	0.306±0.002	0.322±0.008	0.299±0.009
ETTh2	96	0.239±0.019	0.263±0.020	0.336±0.021	0.164±0.013	0.358±0.026	0.432±0.141	0.162±0.009	0.136±0.003	0.137±0.002	0.146±0.002	**0.133**±0.004
	192	0.313±0.034	0.273±0.024	0.265±0.043	0.226±0.018	0.457±0.081	0.625±0.170	0.186±0.018	0.155±0.003	0.155±0.002	0.167±0.003	**0.152**±0.003
	336	0.376±0.034	0.265±0.042	0.350±0.031	0.274±0.022	0.481±0.078	0.793±0.319	0.257±0.023	0.171±0.005	0.172±0.002	0.186±0.001	**0.167**±0.001
	720	0.990±0.023	0.327±0.033	0.406±0.056	0.302±0.040	0.445±0.016	0.539±0.090	0.280±0.014	0.173±0.003	0.176±0.003	0.187±0.003	**0.170**±0.004
Electricity	96	0.083±0.009	0.088±0.014	0.344±0.006	0.153±0.137	0.096±0.002	0.094±0.003	0.073±0.002	0.072±0.002	0.089±0.001	0.074±0.002	**0.065**±0.002
	192	0.093±0.024	0.097±0.009	0.345±0.006	0.200±0.094	0.100±0.004	0.097±0.002	0.080±0.004	0.079±0.002	0.093±0.001	0.079±0.002	**0.071**±0.001
	336	0.095±0.001	–	0.462±0.054	–	0.102±0.007	0.099±0.001	**0.054**±0.001	0.085±0.001	0.098±0.001	0.087±0.000	0.077±0.000
	720	0.106±0.007	–	0.478±0.005	–	0.108±0.003	0.114±0.013	**0.057**±0.005	0.097±0.002	0.108±0.001	0.101±0.001	0.090±0.002
Traffic	96	0.215±0.003	0.208±0.004	0.294±0.003	–	0.202±0.004	0.187±0.002	0.186±0.003	0.216±0.004	0.247±0.000	0.218±0.000	**0.172**±0.001
	192	–	–	0.306±0.004	–	0.208±0.003	0.192±0.001	0.188±0.002	0.217±0.001	0.240±0.000	0.218±0.001	**0.174**±0.001
	336	–	–	0.317±0.006	–	0.213±0.003	0.201±0.004	0.195±0.003	0.222±0.004	0.241±0.000	0.222±0.001	**0.178**±0.001
	720	–	–	0.391±0.002	–	0.220±0.002	0.211±0.004	0.200±0.001	0.232±0.002	0.247±0.001	0.231±0.001	**0.184**±0.001
Weather	96	0.139±0.008	0.105±0.011	0.104±0.020	0.068±0.008	0.130±0.017	0.116±0.013	0.080±0.007	0.070±0.002	0.065±0.000	0.077±0.003	**0.057**±0.000
	192	0.143±0.020	0.142±0.022	0.134±0.012	0.068±0.006	0.127±0.019	0.122±0.021	0.079±0.009	0.076±0.001	0.070±0.002	0.082±0.003	**0.062**±0.000
	336	0.129±0.012	0.133±0.014	0.137±0.010	0.083±0.002	0.130±0.006	0.128±0.011	0.082±0.010	0.079±0.002	0.071±0.003	0.086±0.005	**0.063**±0.001
	720	0.122±0.006	0.113±0.004	0.152±0.003	0.087±0.003	0.113±0.011	0.110±0.004	0.084±0.003	0.082±0.001	0.075±0.001	0.092±0.001	**0.066**±0.001
ILI	24	0.097±0.010	0.092±0.019	0.228±0.024	0.250±0.013	0.275±0.047	0.257±0.003	**0.087**±0.003	0.096±0.014	0.093±0.002	0.126±0.002	0.095±0.007
	36	0.117±0.017	0.115±0.011	0.235±0.010	0.285±0.010	0.272±0.057	0.281±0.004	0.113±0.005	0.103±0.006	0.111±0.003	0.139±0.002	**0.095**±0.009
	48	0.128±0.019	0.133±0.022	0.265±0.039	0.285±0.036	0.295±0.033	0.288±0.008	0.124±0.010	0.114±0.008	0.128±0.001	0.157±0.001	**0.108**±0.012
	60	0.172±0.034	0.155±0.018	0.263±0.022	0.283±0.012	0.295±0.083	0.307±0.005	0.142±0.008	0.134±0.008	0.134±0.002	0.166±0.002	**0.128**±0.007

Due to the excessive time and memory consumption, some results are unavailable and denoted as -.

Table 4: Results of NMAE (mean ± standard deviation) across forecasting horizons. The context length is 36 for ILI and 96 for the other datasets. Each experiment is repeated five times with different random seeds. Lower values indicate better performance. **Red**: the best, Blue: the 2nd best.

Dataset	Horizon	GRU MAF	Trans MAF	TSDiff	CSDI	TimeGrad	GRU NVP	K^2VAE	DistPred	OCE-TS	PPM	QUEST
ETTm1	96	0.402±0.087	0.456±0.042	0.441±0.021	0.308±0.005	0.645±0.129	0.488±0.058	0.284±0.011	0.315±0.005	0.407±0.002	0.282±0.001	**0.278**±0.003
	192	0.476±0.046	0.553±0.012	0.441±0.019	0.377±0.026	0.748±0.084	0.514±0.042	0.323±0.020	0.349±0.005	0.427±0.001	0.308±0.002	**0.306**±0.004
	336	0.522±0.019	0.590±0.047	0.571±0.033	0.419±0.042	0.759±0.015	0.630±0.029	0.330±0.014	0.362±0.002	0.440±0.002	0.330±0.001	**0.326**±0.005
	720	0.711±0.081	0.822±0.034	0.622±0.045	0.578±0.051	0.793±0.034	0.707±0.050	0.373±0.032	0.392±0.002	0.466±0.002	0.368±0.002	**0.357**±0.003
ETTm2	96	0.212±0.082	0.279±0.031	0.224±0.033	0.146±0.012	0.525±0.047	0.413±0.059	0.144±0.011	0.139±0.004	0.141±0.003	0.142±0.004	**0.135**±0.003
	192	0.535±0.029	0.292±0.041	0.316±0.040	0.189±0.012	0.530±0.060	0.427±0.033	0.170±0.009	**0.163**±0.003	**0.163**±0.002	0.166±0.003	**0.163**±0.002
	336	0.407±0.043	0.309±0.032	0.397±0.051	0.248±0.024	0.566±0.047	0.580±0.169	0.187±0.021	0.182±0.006	**0.181**±0.006	0.186±0.005	0.184±0.007
	720	0.355±0.048	0.475±0.029	0.416±0.065	0.306±0.040	0.561±0.044	0.749±0.385	0.275±0.035	**0.207**±0.008	**0.207**±0.008	0.213±0.007	0.211±0.009
ETTh1	96	0.371±0.034	0.423±0.047	0.510±0.029	0.557±0.022	0.585±0.058	0.481±0.037	0.336±0.041	0.322±0.004	0.334±0.004	0.318±0.004	**0.315**±0.007
	192	0.430±0.022	0.451±0.012	0.596±0.056	0.625±0.065	0.680±0.058	0.531±0.018	0.372±0.023	**0.350**±0.005	0.366±0.003	0.357±0.003	**0.350**±0.006
	336	0.462±0.049	0.481±0.041	0.581±0.035	0.574±0.026	0.666±0.047	0.580±0.064	0.394±0.022	**0.371**±0.008	0.384±0.006	0.379±0.007	0.378±0.009
	720	0.496±0.019	0.455±0.025	0.657±0.017	0.657±0.014	0.672±0.015	0.643±0.046	0.396±0.012	**0.382**±0.007	0.400±0.002	0.400±0.011	0.403±0.015
ETTh2	96	0.292±0.012	0.345±0.042	0.421±0.033	0.214±0.018	0.448±0.031	0.548±0.158	0.189±0.010	0.174±0.002	0.177±0.002	0.179±0.002	**0.171**±0.002
	192	0.376±0.112	0.343±0.044	0.339±0.033	0.294±0.027	0.575±0.089	0.766±0.223	0.213±0.021	0.199±0.006	0.199±0.005	0.204±0.005	**0.196**±0.007
	336	0.454±0.057	0.333±0.078	0.427±0.041	0.353±0.028	0.606±0.095	0.942±0.408	0.263±0.039	0.221±0.009	0.220±0.008	0.227±0.008	**0.216**±0.006
	720	1.092±0.019	0.412±0.020	0.482±0.022	0.382±0.030	0.550±0.018	0.688±0.161	0.278±0.020	0.225±0.005	0.228±0.006	0.235±0.004	**0.221**±0.009
Electricity	96	0.108±0.009	0.114±0.010	0.441±0.013	0.203±0.189	0.119±0.003	0.118±0.003	0.093±0.002	0.093±0.003	0.113±0.001	0.092±0.001	**0.084**±0.001
	192	0.120±0.033	0.131±0.008	0.441±0.005	0.264±0.129	0.124±0.005	0.121±0.003	0.102±0.010	0.102±0.004	0.118±0.002	0.099±0.001	**0.090**±0.001
	336	0.122±0.018	–	0.571±0.022	–	0.126±0.008	0.123±0.001	0.107±0.002	0.110±0.002	0.124±0.001	0.109±0.001	**0.099**±0.000
	720	0.136±0.098	–	0.622±0.142	–	0.134±0.004	0.144±0.017	0.117±0.019	0.125±0.003	0.139±0.000	0.128±0.001	**0.116**±0.001
Traffic	96	0.274±0.012	0.265±0.007	0.342±0.042	–	0.234±0.006	0.231±0.003	0.230±0.010	0.273±0.006	0.312±0.003	0.275±0.002	**0.216**±0.002
	192	–	–	0.354±0.012	–	0.239±0.004	0.236±0.002	0.234±0.003	0.273±0.002	0.302±0.001	0.273±0.001	**0.217**±0.001
	336	–	–	0.392±0.006	–	0.246±0.003	0.248±0.006	0.242±0.007	0.279±0.005	0.302±0.000	0.279±0.001	**0.222**±0.001
	720	–	–	0.478±0.006	–	0.263±0.001	0.264±0.006	0.248±0.010	0.293±0.003	0.310±0.002	0.292±0.001	**0.229**±0.001
Weather	96	0.176±0.011	0.139±0.010	0.113±0.022	0.087±0.012	0.164±0.023	0.145±0.017	0.086±0.011	0.088±0.002	0.080±0.003	0.096±0.002	**0.073**±0.002
	192	0.166±0.022	0.160±0.037	0.144±0.020	0.086±0.007	0.158±0.024	0.147±0.025	0.083±0.011	0.095±0.002	0.086±0.003	0.101±0.002	**0.079**±0.001
	336	0.168±0.014	0.170±0.027	0.138±0.033	0.098±0.002	0.162±0.006	0.160±0.012	0.093±0.010	0.099±0.003	0.086±0.000	0.101±0.002	**0.080**±0.001
	720	0.149±0.034	0.148±0.040	0.141±0.026	0.102±0.005	0.136±0.020	0.135±0.008	0.099±0.009	0.101±0.002	0.090±0.002	0.106±0.001	**0.084**±0.002
ILI	24	0.124±0.019	0.118±0.033	0.242±0.086	0.263±0.012	0.296±0.044	0.283±0.001	**0.116**±0.011	0.127±0.016	**0.116**±0.004	0.141±0.003	0.121±0.006
	36	0.144±0.011	0.143±0.089	0.246±0.117	0.298±0.011	0.298±0.048	0.307±0.007	0.142±0.008	0.133±0.006	0.127±0.004	0.158±0.003	**0.120**±0.012
	48	0.159±0.020	0.160±0.039	0.275±0.044	0.301±0.034	0.320±0.025	0.314±0.009	0.152±0.017	0.145±0.010	0.150±0.001	0.175±0.002	**0.132**±0.016
	60	0.216±0.014	0.183±0.019	0.272±0.020	0.299±0.013	0.325±0.068	0.333±0.005	0.167±0.007	0.171±0.014	0.162±0.002	0.184±0.003	**0.159**±0.009

Due to the excessive time and memory consumption, some results are unavailable and denoted as -.