

# IQR: QUANTILE REGRESSION WITH QR ORTHOGONAL DECOMPOSITION FOR RESOURCE SCHEDULING OPTIMIZATION WITHOUT EMPIRICAL MODEL

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

Optimal resource scheduling aims to cover resource demand with minimum economic cost, which is far from model-based constraint optimization or data-driven prediction based scheduling process. To address this model-free constraint optimization issue, a sparse system identification framework with quantile regression and QR orthogonal decomposition (iQR) is proposed for complex systems ranging from small to large scales. It leverages quantile optimization with  $L_1$ -norm to construct a business-driven strategy to reach the proportion of meeting resource demand. It also involves a complete-mapping Fourier Transformation process and an orthogonal least squares technique to select basis vectors in advance to achieve fast regression with sparse mathematical expression, which reduces the number of basis functions from thousands to hundreds and to dozens. iQR represents a specific expectation for single time series prediction, which only achieves predictions that deviate from the true values as little as possible, but aims for predictions consistently higher than the actual values of real demand. Numerical experiments was conducted on eight datasets, including commonly used time series and real-world CPU resource data. The results indicate that most neural network-based methods fail to balance both resource demands and prediction accuracy effectively. In contrast, iQR can achieve optimal scheduling with the minimum economic cost and it is easier to satisfy the business constraints with quantile tuning. Notably, iQR is lightweight with a training speed in seconds and does not rely on the support of computing power of GPU resources. This study may provide new insight into investigations on resource scheduling optimization issues.

## 1 INTRODUCTION

Optimal resource scheduling seeks to meet demand with minimal cost, which is essential across various industries such as services, energy, and communications due to its substantial economic benefits (He et al., 2012; Zhang et al., 2018). Such constraint optimization problems often lack explicit formulas and is typically solved by either model-based planning methods or data-driven prediction based approaches. However, both of them have significant shortcomings with little consideration of empirical modeling errors or business constraints. Recently, numerous methods have been proposed to tackle this issue. Firstly, model-based optimization techniques are applied in resource scheduling, such as NFS (Jones & Peet, 2021), STS (Wang et al., 2021), and DD-TMPC (Zhao et al., 2024). Despite their utility, these methods struggle to capture the complexities of dynamic environments and incur significant computational costs for model maintenance and updates. Secondly, prediction-based scheduling solutions are also prevalent. Contemporary neural network approaches for time series forecasting include models like iTransformer (Liu et al., 2023), FITS (Xu et al., 2024), PatchTST (Nie et al., 2023), and TimesNet (Wu et al., 2023). These models are capable of providing long-term forecasts for a variety of multivariate data types, including electricity, exchange rates, weather, and disease data (Zhou et al., 2022; Zhang & Yan, 2022; Zeng et al., 2023; Bi et al., 2023; Zhou et al., 2024). However, these methods fall short of adequately meeting business requirements, even when the objective function is simply modified to quantile regression to provide a confidence of forecasting. Additionally, they necessitate GPU resources, which are not accessible to some smaller laboratories. Thirdly, direct application of quantile regression for scheduling is another approach. These methods

054 provide good interpretability and aligns well with business requirements (Zarnani et al., 2019; Zhang  
055 et al., 2018). Nonetheless, due to the inherent complexity of time series data, these methods often  
056 require extensive running time, which exceeds practical tolerance limits. In light of these challenges,  
057 we aim to develop a lightweight, business-oriented, more interpretable (with explicit formulas)  
058 framework for resource scheduling problems. This framework will be specifically designed to meet  
059 business requirements and deliver accurate and rapid predictions without relying on complex neural  
060 network models or extensive pre-training process.

061 Thus, we propose a sparse system identification framework with quantile regression and QR orthog-  
062 onal decomposition (iQR) for optimal resource scheduling. Specifically, iQR applies the Fourier  
063 transform to time series, selecting the most influential basis vectors to build a basis vectors library.  
064 Additionally, we apply Seasonal-Trend decomposition using LOESS (STL) decomposition (Cleve-  
065 land et al., 1990) to the input sequence, allowing individual identification for different time series  
066 and improving the precision of long-term scheduling. To enhance efficiency, we initially screen  
067 the basis vectors before incorporating business constraints. We design the Stepwise Orthogonal  
068 Regression (SOR) algorithm to retain only the most important basis vectors for training. Subsequently,  
069 we employ  $L_1$ -norm optimization with quantile regression to solve the business-driven problem. The  
070 main contributions of iQR can be summarized as follows:

- 071 • A novel business-driven identification-prediction framework has been introduced, offering  
072 remarkable flexibility by enabling the customization of the objective function according  
073 to specific business satisfaction levels. It enables rapid, cost-effective scheduling across  
074 various scenarios, offering new insights for future resource scheduling applications.
- 075 • By incorporating global information in the form of Fourier bases into local identifications  
076 and leveraging STL decomposition to differentiate identifications, iQR enables long-term  
077 and comprehensive identifications for time series. This innovative approach combines data-  
078 driven techniques with explicit mathematical expressions, enhancing both interpretability  
079 and scheduling capabilities.
- 080 • The specially designed SOR algorithm for the further selection of basis vectors and the  
081 utilization of quantile optimization accelerate the optimization process with sparse mathe-  
082 matical expression, which reduces the number of basis functions from thousands to hundreds  
083 and even to dozens. This contributes to the investigation of regression acceleration, and  
084 enables the iQR framework to be swiftly applied in business-driven scheduling scenarios.
- 085 • Impressive scheduling accuracy was achieved on eight benchmark datasets, surpassing  
086 current state-of-the-art (SOTA) outcomes. Our method is more effective in satisfying  
087 scheduling demands, and when constraints are met, it incurs significantly lower economic  
088 costs. Notably, iQR is lightweight and does not rely on GPU computing power support  
089 during the training process compared with neural network based methods.

## 091 2 RELATED WORK

092  
093 **Model-based Optimization for Scheduling** Model-based scheduling optimization methods utilize  
094 explicit empirical models of the environment to optimize schedules. Based on the well-designed  
095 model, many works have been proposed to address the resource scheduling issues with convex or  
096 non-convex optimization solving approaches, including NFS (Jones & Peet, 2021), STS (Wang et al.,  
097 2021), DD-TMPC (Zhao et al., 2024). However, several drawbacks exist in these methods, e.g., the  
098 difficulty of accurately modelling complex dynamics and the high computational cost on updating  
099 models. Model-free methods are increasingly gaining attention due to their enhanced adaptability  
100 and capability of handling complex dynamics behind the data. Latest works develop the model-free  
101 scheduling policy for optimization, involving Zhao et al. (2023); Zhang et al. (2024). Model-free  
102 optimization strategies directly learn optimal policies from interactions without explicit environmental  
103 models, and are more robust to changes in the dynamical process.

104 **Prediction-based Scheduling** Recently, methods for time series prediction using neural networks  
105 have gained widespread attention. Transformer-based methods aim to continuously reduce the time  
106 complexity of the model while improving accuracy. These methods include Timemixer (Wang et al.,  
107 2024), TimesNet (Wu et al., 2023), Basisformer (Ni et al., 2023), iTansformer (Liu et al., 2023),  
PatchTST (Nie et al., 2023), FEDformer (Zhou et al., 2022), FITS (Xu et al., 2024), and others.

Despite the significant advancements of Transformer-based methods, their accuracy and training speed can still be unsatisfactory in some cases. As a result, two optimization approaches have emerged. The first approach involves leveraging larger models (Zhou et al., 2024; Cao et al., 2024) to further enhance accuracy, but this leads to slower total times. The second approach involves abandoning the Transformer-based framework and instead implementing plain neural networks (Zeng et al., 2023; Xu et al., 2024; Das et al., 2023) for time series prediction. Unfortunately, these deep learning methods often excel at solving a specific tasks, and their performance may decline when prediction requirements change or business constraints are introduced.

**Quantile Regression-based scheduling** Common quantile regression include FLM (Zarnani et al., 2019), NWP (Zarnani et al., 2019) or traditional model TRMF (Yu et al., 2016), SINDy (Brunton et al., 2016), TSM (Gao & Yan, 2022) with quantile regression. These models can achieve quantile regression by utilizing their own basis function library without relying on the backpropagation process of the support of GPU. Moreover, there are existing methods that can handle business-driven problems (Dai, 2023; Wang & He, 2024; Li & Zhu, 2008). However, most of these methods lack open-source code, which limits their accessibility. Compared to data-driven approaches, these methods may face challenges in making long-term scheduling, and the prediction accuracy may not be as high as that of neural network-based methods. Besides, when incorporating business-driven constraints into them, although they can be solved in a CPU environment, the execution speed becomes significantly slower, especially when dealing with large basis function libraries. This poses a challenge in meeting the speed requirements of resource scheduling.

### 3 METHODOLOGY

This section details the framework and modules of iQR. Firstly, the overall structure of iQR is presented in Section 3.1, as well as the roles of the key components of iQR. Then, the two-phase identification process including the global-local coordination mechanism is comprehensively introduced in Section 3.2 and 3.3. Moreover, the theoretical description regarding the key basis vectors selection algorithm, Stepwise Orthogonal Regression (SOR), is presented in Section 3.4.

**Problem Definition** Given the historical sequence representing the resource demand at timestamp  $t$ , referred to  $\mathbf{X}_t = [X_{t-H+1}, X_{t-H+2}, \dots, X_t] \in \mathbb{R}^{H \times d_x}$  of length  $H$ , the task is to generate a future sequence  $\hat{\mathbf{Y}}_t = [\hat{Y}_{t+1}, \hat{Y}_{t+2}, \dots, \hat{Y}_{t+P}]$  of length  $P$ . Its values exceed the ground truth  $\mathbf{Y}_t = [Y_{t+1}, Y_{t+2}, \dots, Y_{t+P}] \in \mathbb{R}^{P \times d_y}$  while minimizing the gap between them, ensuring efficient resource allocation. Here,  $d_x$  and  $d_y$  denote the dimensions of input and output features, respectively. The problem is denoted as

$$\begin{aligned} \min \quad & (\hat{\mathbf{Y}}_t - \mathbf{Y}_t) \cdot \mathbb{I}(\hat{\mathbf{Y}}_t \geq \mathbf{Y}_t), \\ \text{s.t.} \quad & \mathbb{I}(\hat{\mathbf{Y}}_t \geq \mathbf{Y}_t) \geq \tau, \end{aligned} \tag{1}$$

where  $\mathbb{I}(u)$  is the indicator function, equals to 1 if  $u \geq 0$  and 0 otherwise. The quantile  $\tau \in [0, 1]$  represents the percentage of identification results that are greater than or equal to the true values. For further visualization and better understanding, please refer to Appendix B. Note that Eq. (1) established in a resource-constrained environment, e.g., CPU resource scheduling in cloud networks, is the focal issue in this study, which is far from a single prediction problem with no business constraint here. One must first satisfy the constraint, then address the optimization.

#### 3.1 iQR FRAMEWORK

iQR is a business-driven framework based on system identification, combining basis selection strategy and quantile regression. As shown in Figure 1, it consists of two main phases: global identification and local identification. The two-phase identification (Liu et al., 2024) process is adopted to capture macroscopic and microscopic trends, respectively. Each channel of iQR contains only one univariate sequence, and each sequence is modeled separately. For channel  $i$ , the sequence  $X^i$  is split into a training set  $X_{train}^i$  and a test set  $X_{test}^i$ . Concretely, the Fast Fourier Transform (FFT) is employed to extract periods  $\mathbf{T}$  in the series, which are used to construct the basis vectors library  $\Theta$ . Stepwise Orthogonal Regression (SOR) eliminates independent basis vectors and yields significant ones  $\hat{\Theta}$ ,

which plays the role of second basis selection. The  $L_1$ -norm Quantile Regression ( $L_1$ QR) ensures output  $\hat{Y}$  exceeds the ground truth  $Y$ , controlling the lower bound of output results.

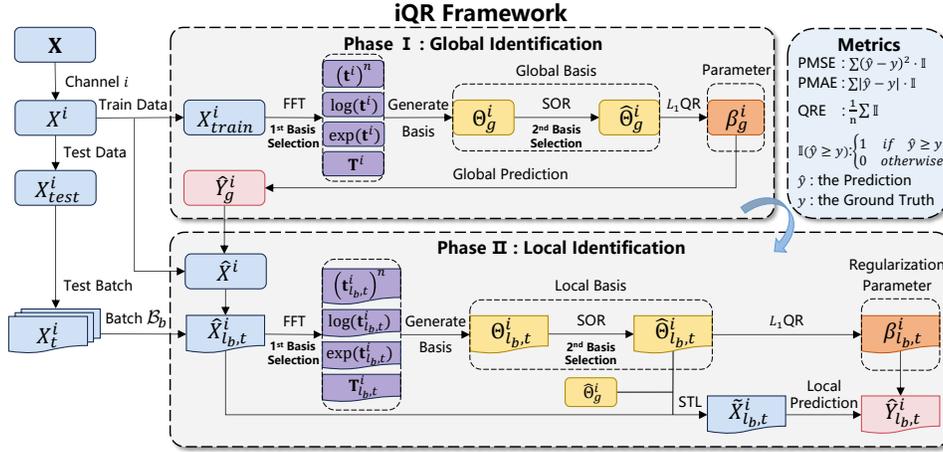


Figure 1: The overall framework of iQR. Global identification captures macroscopic changes. Local identification utilizes macro trends and investigates micro variations to achieve rolling predictions.

The roles of the key components of iQR are described as follows:

**Global-Local Coordination** The coordination identification strategy aims at dynamic analysis of input series. The global identification focus on the pivotal trends, whereas the local identification leverages the macro information provided by its global counterpart, digs micro sensitive variations.

**FFT** The Fourier transform is used to reveal the complete description in frequency domain, since mostly-adopted polynomial basis vectors inspired by the Taylor’s formula can only denote the spontaneous feature of a single point with even known nonlinear function of the signal.

**SOR** SOR is the key selection method of basis vectors suited for sequence identification. It selects the influential ones that contributes most to the model, and ensures the selection at each step is independent and valid by orthogonal projection. The basis vectors are then input into  $L_1$ QR for identification.

**$L_1$ QR** To facilitate sparsity among model coefficients, the LASSO penalty term is introduced into the loss function.  $L_1$ -norm Quantile Regression effectively shrinks unimportant or irrelevant feature weights to zero, thereby generates a robust, compact, and interpretable model structure.

### 3.2 GLOBAL IDENTIFICATION

In the global identification phase, the primary objective is to extract meaningful patterns and trends from historical data that can serve as the foundation for subsequent local identification.

Concretely, global identification applies FFT to the training data. This mathematical operation transforms the time-domain signal into its frequency-domain representation, revealing the underlying frequency components  $f = \text{FFT}(\mathbf{X})$ . Then, we identify the most salient frequencies, typically those with the highest amplitudes. Using the frequency-period relationship  $f = 1/T$ , we convert these significant frequencies into their corresponding periods  $\mathbf{T}^i = [T_1^i, T_2^i, \dots, T_n^i]$ . These periods present the characteristic timescales or cyclic behaviors that are influential in shaping the time series dynamics. Based on the representative periods, the global basis vectors is presented as:

$$\Theta_g^i = \left[ (\mathbf{t}^i)^n, \log(\mathbf{t}^i), \exp(\mathbf{t}^i), \sin\left(\frac{2\pi\mathbf{t}^i}{\mathbf{T}^i}\right), \cos\left(\frac{2\pi\mathbf{t}^i}{\mathbf{T}^i}\right) \right], \quad (2)$$

where  $n \in \{0, 1, \dots\}$ , the timestamp set  $\mathbf{t}^i$  is  $[0, 1, \dots, L_{train}]$ , and  $L_{train}$  is the length of the training set.

Rather than employing a fixed set of basis vectors, we utilize SOR to dynamically select the most informative and relevant basis vectors from the initial basis set identified in the previous phase, the basis vectors are  $\hat{\Theta}_g^i$ , the sparse system identification process with  $L_1$ QR can be represented as

$$\arg \min_{\beta_g^i} \sum_{i=1}^n \rho_{\tau} (Y^i - X^i \beta_g^i) + \lambda \|\beta_g^i\|_1. \quad (3)$$

Given the coefficients  $\beta_g^i$ , the global identification is calculated by  $\hat{Y}_g^i = X^i \beta_g^i$ . Different from methods based on neural networks, the global identification phase utilizes the entire training set for identification, thus reducing the likelihood of accumulated errors propagating into future outputs.

### 3.3 LOCAL IDENTIFICATION

The local identification phase builds upon the knowledge of global identification and focuses on the accurate rolling output on the test set.

Concretely, the overall architecture of local identification is similar to global identification phase, and the pivotal distinction lies in the segmentation of the test dataset into numerous batches, denoted as  $\mathcal{B}_b$ , for  $i$  ranging from 1 to  $N$ . Conventional regression methods and advanced deep learning frameworks customarily generate subsequent sequence directly from the test sequence. However, the neglect of global trends significantly decrease the prediction accuracy. Consequently, a knowledge fusion technique is introduced, which is accomplished via a weighted mean strategy that synthesizes the global inferred sequence  $Y_g^i$  with the local entries  $X^i$ , formalized as follows:

$$\hat{X}^i = \mathbf{w} X^i + (1 - \mathbf{w}) \hat{Y}_g^i, \quad (4)$$

where  $\mathbf{w}$  is a vector constructed by generating  $L_{batch}$  equally spaced points on the interval  $[\alpha^{1/\gamma}, \xi^{1/\gamma}]$ , and then raising each of these points to the power of  $1/\gamma$ . Formally,  $w_i = (\alpha^{1/\gamma} + b\Delta)^\gamma$ , for  $b = 0, \dots, L_{batch} - 1$  and  $\Delta = (\xi^{1/\gamma} - \alpha^{1/\gamma}) / (L_{batch} - 1)$  defines the uniform spacing between consecutive points. The weighted coefficient  $\mathbf{w}$  balances the influence of overarching tendencies against localized fluctuations. Applying the strategy to  $\mathcal{B}_b$  yields the refined inputs.

FFT is used to extract an initial set of basis vectors. For a test batch  $\mathcal{B}_b$ , the local basis vectors are

$$\Theta_{l_b,t}^i = \left[ (\mathbf{t}_{l_b,t}^i)^n, \log(\mathbf{t}_{l_b,t}^i), \exp(\mathbf{t}_{l_b,t}^i), \sin\left(\frac{2\pi \mathbf{t}_{l_b,t}^i}{\mathbf{T}_{l_b,t}^i}\right), \cos\left(\frac{2\pi \mathbf{t}_{l_b,t}^i}{\mathbf{T}_{l_b,t}^i}\right) \right], \quad (5)$$

where  $\mathbf{t}_{l_b,t}^i$  uniformly samples from 0 to the length of the training part in test batch  $\mathcal{B}_b$  and  $\mathbf{T}_{l_b,t}^i$  defines the representative periods. To refine and emphasize higher frequency constituents, the SOR algorithm is invoked, leading to the optimized local basis vectors  $\hat{\Theta}_{l_b,t}^i$ . Before applying the selected basis vectors for identification, the local time series is enhanced by performing STL decomposition. STL decomposes the data into seasonal, trend, and residual components, enabling isolate and adjust for any seasonality presented in the test data. The resulting adjusted series serves as the input for the next step. After the LASSO identification, the input sequence is

$$\tilde{X}_{l_b,t}^i = \text{LASSO}(\mathcal{S}_{l_b,t}^i) + \text{LASSO}(\mathcal{T}_{l_b,t}^i) + \mathcal{R}_{l_b,t}^i. \quad (6)$$

For the input series  $\tilde{X}_{l_b,t}^i$ , the objective function of  $L_1$ QR is

$$\arg \min_{\beta_{l_b,t}^i} \sum_{i=1}^n \rho_{\tau} (Y_{l_b,t}^i - \tilde{X}_{l_b,t}^i \beta_{l_b,t}^i) + \lambda \|\beta_{l_b,t}^i\|_1. \quad (7)$$

Utilizing the fitted model, we perform local rolling predictions by iteratively updating the model with new observations. This process continually adapts the model to the most recent data, capturing local fluctuations while maintaining the global context provided by the global identification.

### 3.4 THE KEY MODULE OF IQR FRAMEWORK: SOR

To identify significant basis vectors and further eliminate irrelevant ones, the key module of iQR, Stepwise Orthogonal Regression (SOR), has been proposed. The detail theory is shown as follows.

Given the initial set of basis vectors  $\Theta = \{\theta_1, \theta_2, \dots, \theta_p\}$ , we employ the following Lemma 1 to ensure orthogonality among basis vectors.

**Lemma 1.** (*QR Decomposition (Golub & Van Loan, 2013)*) Given a real  $n \times n$  square matrix  $A$ , there exist  $n \times n$  orthogonal matrix  $Q$  and upper triangular matrix  $R$  such that  $A = QR$ .

Then, the Lemma 1 is extended to a general scenario, that is, matrices of arbitrary dimensions.

**Corollary 1.** Any real  $m \times n$  matrix  $A$  can be decomposed into the product of an  $m \times n$  orthogonal matrix  $Q$ , whose columns are mutually orthogonal, and an  $n \times n$  upper triangular matrix  $R$ .

**Corollary 2.** Any real  $m \times n$  matrix  $A$  can be factorized into the product  $A = VR$ , where  $V$  is an orthogonal matrix with its columns being pairwise orthogonal, and  $R$  is an upper triangular matrix with diagonal entries equal to 1.

According to Corollary 2, the basis vectors  $\Theta$  is orthogonalised to filter the insignificant basis vectors, and subsequently serve for the later lemma. The following lemmas show a stepwise regression strategy to efficiently select the optimal subset of basis vectors.

**Lemma 2.** (*Orthogonality Principle*) For an unbiased optimal estimator (i.e., the least squares estimation), if the basis vectors are orthogonal, then the error is orthogonal to any basis vector.

**Lemma 3.** (*Orthogonal Greedy Optimality*) If the basis vectors are orthogonal, then a greedy stepwise optimal selection of basis vectors will yield the same result as the global optimal solution.

**Remark 1.** Since the basis vectors are orthogonal, the contribution of each function to the model can be evaluated independently of the others. At each step of the greedy algorithm, the basis vector that minimizes the sum of squares of the residuals is selected. Due to orthogonality, this incremental selection is equivalent to evaluating the contributions of all possible subsets of vectors at the same time, ensuring that the greedy selection is consistent with the global optimum. Then, we have the following the key theorem of SOR, which plays the role of the second filtering strategy of basis vectors, and ensuring obtaining the optimal basis vectors.

**Theorem 1.** (*Stepwise Orthogonal Regression*) If the set of basis vectors  $\{\theta_1^*, \theta_2^*, \dots, \theta_n^*\}$  is orthogonal, then the greedy stepwise optimal selection of basis vectors will yield the same result as the global optimal solution.

**Remark 2.** SOR obtains the same result as the global optimum by sequentially selecting orthogonal basis vectors. It independently evaluate the contribution of each basis vectors to the model at each step of the greedy algorithm, selecting the basis vectors that minimizes the sum of squared residuals. The orthogonality ensures that this incremental selection is equivalent to simultaneously evaluating the contributions of all possible subsets of vectors, thus guaranteeing consistency with the global optimum. The insight is to use Stepwise Orthogonal Regression to filter out significant basis vectors from the initial set and eliminate irrelevant ones, ultimately obtaining the optimal basis vectors.

The proof of Lemma 2, Lemma 3, and Theorem 1 are present in Appendix F.1, Appendix F.2, and Appendix F.3, separately. Thus, the stepwise optimal basis vectors selection is feasible.

## 4 EXPERIMENTS

To evaluate the performance of iQR, especially the demand satisfaction rate on resource scheduling tasks, we conduct extensive experiments on eight real-world time series benchmarks, and compare the performance with recent state-of-the-art (SOTA) methods.

**Datasets** Experiments are performed on various domains of datasets, covering electricity, energy, exchange and resource. All datasets are split into training, validation and test set chronologically with the ratio 7 : 1 : 2. For fair comparison, the datasets are normalized with the z-score normalization. The details regarding datasets and experimental settings are provided in Appendix D.1.

**Baselines** A comprehensive comparison of effectiveness is performed against 7 baselines, including the SOTA long-term forecasting models TimeMixer (Wang et al., 2024) and PatchTST (Nie et al., 2023), the SOTA short-term forecasting models TimesNet (Wu et al., 2023) and FEDformer (Zhou et al., 2022), as well as other representative and competitive models, i.e., BasisFormer (Ni et al., 2023), iTransformer (Liu et al., 2023) and DLinear (Zeng et al., 2023). Please refer to Appendix D.2 for additional implementation details of baselines.

**Implementation Details** iQR is implemented in Python on a CPU environment, yet other experiments are performed with a single NVIDIA RTX A6000 GPU using PyTorch. All baselines are trained with a learning rate of  $10^{-4}$  and the same quantile loss. We take Positive Mean Squared Errors ( $PMSE = \sum_{i=1}^N (\hat{y} - y)^2 \cdot \mathbb{I}(\hat{y} \geq y)$ ), Positive Mean Absolute Errors ( $PMAE = \sum_{i=1}^N |\hat{y} - y| \cdot \mathbb{I}(\hat{y} \geq y)$ ), and Quantile Regression Errors ( $QRE = \frac{1}{N} \sum_{i=1}^N \mathbb{I}(\hat{y} \geq y)$ ) as the evaluation metrics. For more evaluation details, please refer to Appendix D.3 and Appendix D.4.

#### 4.1 MAIN RESULTS

Resource scheduling is crucial in electricity, weather, and exchange rates as it optimizes resource allocation, and ensures system reliability. Resource datasets are usually multivariate time series and are considered as multichannel signals. PatchTST (Nie et al., 2023) has shown the effectiveness of channel-independent models. The experiments of iQR are conducted individually on each channel. For a better comparison, we follow the experimental setups in TimeMixer (Wang et al., 2024).

To assess the satisfaction rate in business demand, we conduct extensive resource scheduling experiments on eight datasets. Primary evaluation metric is the proportion of predictions exceeding the ground truth, also known as QRE. As shown in Table 1, iQR achieves the SOTA performance on all benchmarks, involving various time series with different frequencies and domains. Notably, iQR surpasses the latest model TimeMixer with an average QRE improvement of 3.53%. Specifically, iQR significantly outperforms the top-performing model Basisformer, with an 6.13% QRE increase in the weather dataset and a 3.52% QRE increase in the CPU dataset.

Additionally, scheduling CPU resources in advance is crucial for network services in parks. To meet the requirements of daily tasks, the allocated CPU resources must exceed the potential demand. As shown in Table 1, iQR achieves a higher satisfaction rate than TimeMixer on the CPU dataset, with an average increase of 3.98%. Note that iQR always meets the demands under different quantile setups. This further demonstrates the effectiveness of iQR, particularly in resource scheduling scenarios. The visualization results on three datasets are depicted in Figure 2. The local identification results shows the effectiveness of our scheduling strategy.

Table 1: QRE results for resource scheduling on eight datasets with historical length  $H = 96$  and prediction length  $P \in \{96, 192, 336, 720\}$ . The historical and prediction length are set to  $H = 36$  and  $P \in \{24, 36, 48, 60\}$  for ILI. The ratio presents satisfaction rate of business demands. The best result are highlighted in **red bold**, and the second best results are marked with a blue underline.

Methods	IMP.			Ours			TimeMixer			TimesNet			Basisformer			iTransformer			PatchTST			Dlinear			FEDformer			
	$\tau$	0.9	0.95	1	0.9	0.95	1	0.9	0.95	1	0.9	0.95	1	0.9	0.95	1	0.9	0.95	1	0.9	0.95	1	0.9	0.95	1			
ETT <sub>h1</sub>	96	0.3%	0.4%	0.2%	<u>0.904</u>	<u>0.953</u>	<b>1.0</b>	0.884	0.938	0.972	<u>0.901</u>	<u>0.949</u>	<b>1.0</b>	0.887	0.943	<u>0.998</u>	0.874	0.929	<b>1.0</b>	0.666	0.678	0.674	0.771	0.867	<b>1.0</b>			
	192	1.5%	0.7%	0.3%	<u>0.904</u>	<u>0.955</u>	<b>1.0</b>	0.872	0.938	1.0	0.885	0.928	0.95	<u>0.891</u>	<u>0.948</u>	<b>1.0</b>	0.887	0.941	<u>0.997</u>	0.876	0.933	<b>1.0</b>	0.639	0.662	0.658	0.793	0.889	<b>1.0</b>
	336	1.8%	0.3%	0.5%	<b>0.9</b>	<b>0.957</b>	<b>1.0</b>	0.875	0.938	1.0	0.849	0.907	0.941	<u>0.884</u>	<u>0.954</u>	<b>1.0</b>	0.883	0.939	<u>0.995</u>	0.864	0.93	<b>1.0</b>	0.604	0.641	0.637	0.866	0.926	<b>1.0</b>
	720	1.9%	-0.2%	1.4%	<b>0.902</b>	<u>0.95</u>	<b>1.0</b>	0.852	0.919	<b>1.0</b>	0.786	0.87	0.918	<u>0.885</u>	<u>0.952</u>	<b>1.0</b>	0.846	0.917	<u>0.986</u>	0.827	0.896	<b>1.0</b>	0.676	0.605	0.601	0.852	0.914	<b>1.0</b>
ETT <sub>h2</sub>	96	6.2%	1.1%	0.3%	<u>0.955</u>	<u>0.954</u>	<b>1.0</b>	0.82	0.933	<b>1.0</b>	0.831	0.921	0.949	<u>0.899</u>	0.943	<b>1.0</b>	0.889	0.941	<u>0.997</u>	0.872	0.941	<b>1.0</b>	0.741	0.765	0.761	0.878	<u>0.944</u>	<b>1.0</b>
	192	6.9%	0.6%	0.5%	<u>0.949</u>	<u>0.965</u>	<b>1.0</b>	0.846	0.933	<b>1.0</b>	0.82	0.909	0.933	0.884	0.928	<b>1.0</b>	0.883	0.938	<u>0.995</u>	0.873	0.921	<b>1.0</b>	0.729	0.745	0.741	0.888	<u>0.959</u>	<b>1.0</b>
	336	7.7%	0.4%	0.6%	<u>0.961</u>	<u>0.953</u>	<b>1.0</b>	0.786	0.943	<b>1.0</b>	0.822	0.893	0.914	0.888	0.901	<b>1.0</b>	0.876	0.93	<u>0.994</u>	0.838	0.915	<b>1.0</b>	0.723	0.732	0.728	0.892	<u>0.949</u>	<b>1.0</b>
	720	8.0%	4.1%	0.8%	<u>0.967</u>	<u>0.99</u>	<b>1.0</b>	0.804	0.923	<b>1.0</b>	0.845	0.894	0.913	0.857	0.924	<b>1.0</b>	0.871	0.934	<u>0.992</u>	0.892	0.931	<b>1.0</b>	0.76	0.722	0.712	0.895	<u>0.951</u>	<b>1.0</b>
ETT <sub>h3</sub>	96	3.0%	1.0%	0.1%	<u>0.901</u>	<u>0.951</u>	<b>1.0</b>	0.851	0.925	<b>1.0</b>	0.808	0.893	0.998	<u>0.875</u>	<u>0.942</u>	<b>1.0</b>	0.864	0.928	<u>0.999</u>	0.866	0.925	<b>1.0</b>	0.86	0.892	0.9	0.844	0.905	<b>1.0</b>
	192	1.9%	0.1%	0.1%	<u>0.907</u>	<u>0.951</u>	<b>1.0</b>	0.866	0.932	<b>1.0</b>	0.832	0.909	0.996	<u>0.89</u>	<u>0.95</u>	<b>1.0</b>	0.868	0.93	<u>0.999</u>	0.842	0.925	<b>1.0</b>	0.854	0.885	0.892	0.845	0.908	<b>1.0</b>
	336	0.4%	0.7%	0.1%	<u>0.907</u>	<u>0.955</u>	<b>1.0</b>	0.856	0.922	<b>1.0</b>	0.861	0.915	0.995	<u>0.903</u>	<u>0.948</u>	<b>1.0</b>	0.868	0.927	<u>0.999</u>	0.856	0.918	<b>1.0</b>	0.843	0.876	0.881	0.846	0.922	<b>1.0</b>
	720	2.4%	0.1%	0.1%	<u>0.907</u>	<u>0.956</u>	<b>1.0</b>	0.866	0.936	<b>1.0</b>	0.87	0.93	0.993	<u>0.886</u>	<u>0.949</u>	<b>1.0</b>	0.868	0.926	<u>0.999</u>	0.85	0.925	<b>1.0</b>	0.865	0.862	0.864	0.82	0.893	<b>1.0</b>
ETT <sub>h2</sub>	96	3.0%	0.9%	0.1%	<u>0.927</u>	<u>0.967</u>	<b>1.0</b>	0.852	0.92	<b>1.0</b>	0.87	0.93	0.991	<u>0.9</u>	0.946	<b>1.0</b>	0.888	0.939	0.998	0.852	0.931	<u>0.999</u>	0.881	0.955	0.971	0.771	<u>0.958</u>	<b>1.0</b>
	192	3.9%	0.5%	0.1%	<u>0.935</u>	<u>0.96</u>	<b>1.0</b>	0.887	0.93	<b>1.0</b>	0.873	0.926	0.989	<u>0.9</u>	0.939	<b>1.0</b>	0.883	0.935	0.997	0.856	0.931	<u>0.999</u>	0.893	<u>0.955</u>	0.967	0.793	<u>0.96</u>	<b>1.0</b>
	336	4.3%	-0.7%	0.1%	<u>0.94</u>	<u>0.961</u>	<b>1.0</b>	0.864	0.925	<b>1.0</b>	0.863	0.926	0.987	<u>0.901</u>	0.943	<b>1.0</b>	0.881	0.933	0.997	0.877	0.936	<u>0.999</u>	0.891	0.954	0.964	0.866	<u>0.968</u>	<b>1.0</b>
	720	8.1%	1.6%	0.1%	<u>0.959</u>	<u>0.982</u>	<b>1.0</b>	0.861	0.947	<b>1.0</b>	0.87	0.916	0.984	0.886	0.929	<b>1.0</b>	0.886	0.937	0.996	0.85	0.925	<u>0.999</u>	0.887	0.947	0.954	0.852	<u>0.967</u>	<b>1.0</b>
Exchange	96	-4.3%	-2.2%	1.6%	0.9	0.956	<b>1.0</b>	0.917	0.969	<b>1.0</b>	0.725	0.835	0.944	<u>0.949</u>	<u>0.973</u>	<b>1.0</b>	0.917	0.96	<u>0.984</u>	0.907	0.968	<b>1.0</b>	0.598	0.652	0.65	0.99	<u>0.994</u>	<b>1.0</b>
	192	-3.3%	-1.9%	2.9%	0.91	<u>0.977</u>	<b>1.0</b>	0.906	0.944	<b>1.0</b>	0.873	0.739	0.745	<u>0.957</u>	0.966	<b>1.0</b>	0.916	0.951	<u>0.972</u>	0.93	0.965	<b>1.0</b>	0.53	0.596	0.593	0.989	<u>0.999</u>	<b>1.0</b>
	336	-3.0%	-1.6%	4.0%	0.914	0.956	<b>1.0</b>	0.932	<u>0.983</u>	<b>1.0</b>	0.83	0.837	0.836	<u>0.965</u>	0.979	<b>1.0</b>	0.925	0.951	<u>0.962</u>	0.928	0.968	<b>1.0</b>	0.363	0.524	0.519	0.994	<u>0.999</u>	<b>1.0</b>
	720	-2.0%	-0.5%	2.9%	0.913	0.953	<b>1.0</b>	0.965	0.981	<b>1.0</b>	0.862	0.852	0.851	<u>0.978</u>	<u>0.994</u>	<b>1.0</b>	0.961	0.97	<u>0.972</u>	0.965	0.99	<b>1.0</b>	0.652	0.353	0.343	0.998	<u>0.999</u>	<b>1.0</b>
Weather	96	0.1%	3.4%	0.2%	<u>0.905</u>	<u>0.966</u>	<b>1.0</b>	0.83	0.889	<b>1.0</b>	0.817	0.857	0.955	0.832	0.912	<b>1.0</b>	0.838	0.905	0.991	0.864	0.913	<u>0.998</u>	<u>0.904</u>	<u>0.934</u>	0.936	0.894	0.856	<b>1.0</b>
	192	2.6%	2.8%	0.2%	<u>0.92</u>	<u>0.951</u>	<b>1.0</b>	0.837	0.907	<b>1.0</b>	0.725	0.835	0.944	0.836	0.898	<b>1.0</b>	0.855	0.899	0.99	0.855	0.901	<u>0.998</u>	<u>0.897</u>	<u>0.925</u>	0.926	0.859	0.845	<b>1.0</b>
	336	0.6%	4.5%	0.2%	<u>0.904</u>	<u>0.958</u>	<b>1.0</b>	0.825	0.882	<b>1.0</b>	0.708	0.839	0.934	0.83	0.897	<b>1.0</b>	0.835	0.887	0.989	0.825	0.907	<u>0.998</u>	<u>0.899</u>	<u>0.917</u>	0.917	0.854	0.906	<b>1.0</b>
	720	-1.3%	1.9%	0.3%	<u>0.909</u>	<u>0.952</u>	<b>1.0</b>	0.833	0.899	<b>1.0</b>	0.737	0.812	0.917	0.839	0.898	<b>1.0</b>	0.831	0.888	0.987	0.844	0.905	<u>0.997</u>	<u>0.921</u>	0.906	0.905	0.827	<u>0.934</u>	<b>1.0</b>
ILI	24	9.7%	9.8%	1.1%	<u>0.905</u>	<u>0.967</u>	<b>1.0</b>	0.805	0.876	<b>1.0</b>	0.807	0.566	0.596	0.818	<u>0.881</u>	<b>1.0</b>	0.658	0.567	0.698	<u>0.825</u>	0.877	<u>0.989</u>	0.486	0.44	0.488	0.753	0.77	0.963
	36	10.3%	3.6%	1.0%	<u>0.901</u>	<u>0.956</u>	<b>1.0</b>	0.807	<u>0.923</u>	<b>1.0</b>	0.776	0.603	0.611	<u>0.817</u>	0.89	<b>1.0</b>	0.623	0.619	0.682	0.789	0.902	<u>0.99</u>	0.477	0.461	0.482	0.741	0.793	0.967
	48	7.8%	3.9%	1.2%	<u>0.902</u>	<u>0.95</u>	<b>1.0</b>	0.779	0.898	<b>1.0</b>	0.837	0.592	0.589	0.742	0.9	<b>1.0</b>	0.633	0.676	0.628	0.824	<u>0.914</u>	<u>0.988</u>	0.461	0.477	0.461	0.729	0.824	0.97
	60	15.3%	2.9%	1.9%	<u>0.912</u>	<u>0.95</u>	<b>1.0</b>	0.789	<u>0.923</u>	<b>1.0</b>	0.764	0.61	0.566	<u>0.871</u>	<u>0.923</u>	<b>1.0</b>	0.553	0.685	0.578	0.775	0.911	<u>0.981</u>	0.44	0.487	0.44	0.673	0.829	0.964
CPU	96	2.4%	1.7%	0.3%	<u>0.906</u>	<u>0.952</u>	<b>1.0</b>	0.849	0.929	<b>1.0</b>	0.877	0.911	0.979	<u>0.885</u>	<u>0.936</u>	<b>1.0</b>	0.824	0.889	<u>0.997</u>	0.864	0.914	<b>1.0</b>	0.711	0.741	0.738	0.841	0.846	<b>1.0</b>
	192	5.5%	3.4%	0.2%	<u>0.905</u>	<u>0.953</u>	<b>1.0</b>	0.848	0.916	<b>1.0</b>	0.858	0.922	0.971	0.857	0.915	<b>1.0</b>	0.82	0.856	<u>0.998</u>	0.857	0.91	<b>1.0</b>	0.367	0.714	0.712	0.842	0.876	<b>1.0</b>
	336	5.1%	1.9%	0.2%	<u>0.907</u>	<u>0.955</u>	<b>1.0</b>	0.812	0.926	<b>1.0</b>	<u>0.863</u>	<u>0.937</u>	0.97	0.836	0.895	<b>1.0</b>	0.779	0.804	<u>0.998</u>	0.81	0.888	<b>1.0</b>	0.566	0.672	0.671	0.825	0.839	<b>1.0</b>
	720	7.9%	7.1%	0.1%	<u>0.912</u>	<u>0.981</u>	<b>1.0</b>	0.833	<u>0.916</u>	<b>1.0</b>	<u>0.845</u>	0.886	0.929	<u>0.845</u>	0.907	<b>1.0</b>	0.742	0.762	<u>0.999</u>	0.813	0.886	<b>1.0</b>	0.736	0.569	0.566	0.778	0.834	<b>1.0</b>
Ratio	-	-	-	100%	100%	100%	12.5%	9.4%	100%	0%	0%	0%	28.1%	21.9%	100%	12.5%	12.5%	0%	12.5%	12.5%	62.5%	6.3%	12.5%	0%	0.1%	31.3%	87.5%	
Count	-	-																										

378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431

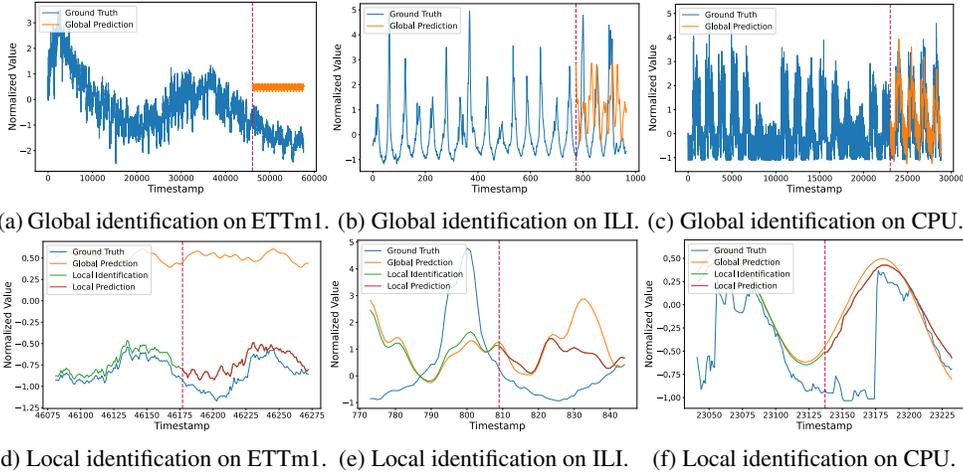


Figure 2: Visualization of identifications on three datasets. The historical and future sequence length are set to 96 for ETTm1 and CPU dataset, while 36 for ILI dataset. The quantile  $\tau$  is 0.9, indicating that over 90% of the identifications are expected to exceed the ground truth.

Furthermore, we have conducted an evaluation of the performance across various quantiles, and the comprehensive results of this analysis can be accessed in Appendix D.5. The findings indicate that iQR is adept at fulfilling business demands while incurring significantly lower costs.

## 4.2 MODEL ANALYSIS

**Ablation Studies** To verify the design of each component in the scheduling framework iQR, we conducted a comprehensive ablation study. The purpose is systematically isolating and evaluating the contribution of the basis vectors selection SOR, the STL, the knowledge fusion technology, and the lasso penalized quantile regression.

Five modified versions are presented, each lacking a key element: (1) model without SOR, denoted as iQR-R, (2) model excludes the STL, marked as iQR-S, (3) model omits the knowledge fusion technology, represented as iQR-K, (4) model without the lasso penalty, termed as iQR-L, and (5) model replaces quantile regression with linear regression, labeled as iQR-Q. The experiments are conducted on ETTh1 dataset.

Table 2 illustrates the performance among variants. iQR outperforms other baselines on all three evaluation metrics, suggesting that it is effective for resource scheduling tasks.

**iQR vs. Maximum Value Baseline** We employ a simple and intuitive evaluation method, the maximum value baseline. Reference line are plotted based on the maximum value observed in the training dataset. The lines symbolize an ideal state where future ground truth is always lower than the maximum values. As shown in Table 3, iQR achieves significant improvement in performance, with an average reduction of 80.49% in PMSE and 65.55% in PMAE.

## 4.3 EFFICIENCY ANALYSIS

**Complexity Analysis** The spatial complexity is  $\mathcal{O}(t \cdot d_y)$ , and the temporal complexity is  $\mathcal{O}((2n + 3n \cdot n_{batch}) \cdot d_y)$ , where  $t$  is the timestamps,  $d_y$  is the number of features, and  $n_{batch}$  is the number of batches in testing phase. Due to channel independence, the model identifies and predicts each feature separately. Specifically, if the dataset contains only one feature, the time complexity of constructing the global basis vectors during the training phase is  $\mathcal{O}(2K^2)$ , which is omitted since  $K \ll t$ ; the

Table 2: Ablation studies on the ETTh1 dataset with  $H = 96$ ,  $P = 96$ , and  $\tau = 0.9$ . The best result are highlighted in **red bold**, and the second best results are marked with a blue underline.

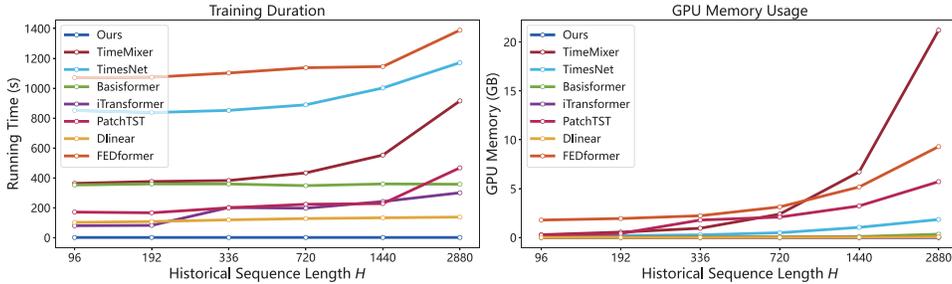
Tasks	iQR	iQR-R	iQR-S	iQR-K	iQR-L	iQR-Q
PMSE	<b>1.185</b>	1.191	1.19	0.909	0.561	<u>0.558</u>
PMAE	<b>0.782</b>	0.787	0.785	0.697	0.574	<u>0.465</u>
QRE	<b>0.904</b>	<u>0.901</u>	0.9	0.845	0.736	0.455

Table 3: Resource scheduling results for iQR and Maximum Value Baseline on ETTh1 datasets with historical length  $H = 96$  and prediction length  $P \in \{96, 192, 336, 720\}$ . The improvement is noted as IMP. and is displayed in the end of the table.

Length	96			192			336			720		
Metrics	PMSE	PMAE	QRE									
ours	2.348	1.282	1.0	2.311	1.279	1.0	2.46	1.309	1.0	2.579	1.348	1.0
Maximum Value	12.412	3.316	1.0	12.198	3.275	1.0	12.436	3.304	1.0	12.654	3.343	1.0
IMP.	81.08%	61.34%	0%	81.05%	60.95%	0%	80.22%	60.38%	0%	79.62%	59.68%	0%

time complexity of the  $L_1$ QR and SOR algorithms are  $\mathcal{O}(n)$ . During the testing phase, both the STL algorithm, the SOR algorithm, and the  $L_1$ QR algorithm have a time complexity of  $\mathcal{O}(n)$ , and the time complexity of constructing local basis vectors can be omitted. Thus, the total time complexity during testing is  $\mathcal{O}(3n)$ . Additionally, the model has fewer parameters due to the absence of neural networks, and the spatial complexity can be approximated as the data scale  $t \cdot d_y$ .

**Memory Cost and Running Time** We compare the GPU memory cost and running time against the current SOTA baselines, covering MLP-based TimeMixer (Wang et al., 2024) and Transformer-based iTransformer (Liu et al., 2023), etc. In Figure 3, iQR outperforms other baselines in terms of running time and memory usage. Specifically, iQR is much faster for various historical sequence length  $H$  from 96 to 2880, with the prediction sequence length is fixed at 96. Moreover, Figure 3(b) show that iQR is lightweight. This further demonstrates the advantages of iQR in terms of resource utilization and time efficiency. Additional efficiency analysis results are presented in Appendix D.8.



(a) Training duration efficiency analysis.

(b) Memory Efficiency Analysis.

Figure 3: Efficiency analysis of eight algorithms. The results are derived from the experiments on the ETTh1 dataset, and we report the average result from ten experiments.

## 5 CONCLUSION REMARKS

In this study, we address the crucial issue of resource scheduling across diverse industrial contexts by introducing a novel framework, termed iQR, which leverages sparse system identification techniques. This framework deviates significantly from conventional model-based or prediction-based scheduling approaches, adopting instead a global-local coordination strategy to bolster identification performance. Specifically, we first utilize Fast Fourier Transform (FFT) to comprehensively extract vital features from a frequency domain perspective. Furthermore, iQR incorporates Stepwise Orthogonal Regression (SOR) to derive more prominent basis vectors and employs  $L_1$ -norm Quantile Regression ( $L_1$ QR) to ensure precise scheduling across various quantile levels. The deployment of  $L_1$ QR for resource scheduling not only fulfills resource demands with high confidence but also minimizes resource allocation costs. Rigorous experiments conducted on eight datasets demonstrate that iQR attains state-of-the-art (SOTA) performance, characterized by a higher satisfaction rate and reduced economic cost, compared to recent deep learning-based prediction methods incorporating quantile regression. Notably, iQR is lightweight, with training durations measured in seconds, and operates efficiently without the need for GPU resources. Looking ahead, we aim to broaden the applicability of iQR to a wider range of scheduling scenarios and explore more potent and efficient solutions for time series analysis in resource-constrained environments.

## REFERENCES

- 486  
487  
488 Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-  
489 range global weather forecasting with 3d neural networks. *Nature*, 619(7970):533–538, 2023.
- 490 Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from  
491 data by sparse identification of nonlinear dynamical systems. *Proc. Natl. Acad. Sci.*, 113(15):  
492 3932–3937, 2016.
- 493 Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. Tempo:  
494 Prompt-based generative pre-trained transformer for time series forecasting. In *ICLR*, 2024.
- 495 Robert B Cleveland, William S Cleveland, Jean E McRae, Irma Terpenning, et al. Stl: A seasonal-  
496 trend decomposition. *J. Off. Stat.*, 6(1):3–73, 1990.
- 497 Sheng Dai. Variable selection in convex quantile regression: L1-norm or L0-norm regularization?  
498 *Eur. J. Oper. Res.*, 305(1):338–355, 2023.
- 499 Abhimanyu Das, Weihao Kong, Andrew Leach, Rajat Sen, and Rose Yu. Long-term forecasting with  
500 tide: Time-series dense encoder. In *TMLR*, 2023.
- 501 Urban Fasel, J Nathan Kutz, Bingni W Brunton, and Steven L Brunton. Ensemble-sindy: Robust  
502 sparse model discovery in the low-data, high-noise limit, with active learning and control. *Proc. R.  
503 Soc. A.*, 478(2260):20210904, 2022.
- 504 Ting-Ting Gao and Gang Yan. Autonomous inference of complex network dynamics from incomplete  
505 and noisy data. *Nat. Comput. Sci.*, 2(3):160–168, 2022.
- 506 Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.
- 507 Yifeng He, Bala Venkatesh, and Ling Guan. Optimal scheduling for charging and discharging of  
508 electric vehicles. *IEEE Trans. Smart Grid*, 3(3):1095–1105, 2012.
- 509 Morgan Jones and Matthew M. Peet. Extensions of the dynamic programming framework: Battery  
510 scheduling, demand charges, and renewable integration. *IEEE Trans. Autom. Control*, 66(4):  
511 1602–1617, 2021.
- 512 Roger Koenker and Gilbert Bassett Jr. Regression quantiles. *Econometrica*, pp. 33–50, 1978.
- 513 Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term  
514 temporal patterns with deep neural networks. In *SIGIR*, 2018.
- 515 Youjuan Li and Ji Zhu.  $L_1$ -norm quantile regression. *J. Comput. Graph. Stat.*, 17(1):163–185, 2008.
- 516 Xiaoyi Liu, Duxin Chen, Wenjia Wei, Xia Zhu, and Wenwu Yu. Interpretable sparse system  
517 identification: Beyond recent deep learning techniques on time-series prediction. In *ICLR*, 2024.
- 518 Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long.  
519 itransformer: Inverted transformers are effective for time series forecasting. In *ICLR*, 2023.
- 520 Alvaro Mendez-Civieta, M. Carmen Aguilera-Morillo, and Rosa E. Lillo. Adaptive sparse group  
521 lasso in quantile regression. *Adv. Data Anal. Classif.*, 15(3):547–573, 2021.
- 522 Zelin Ni, Hang Yu, Shizhan Liu, Jianguo Li, and Weiyao Lin. Basisformer: Attention-based time  
523 series forecasting with learnable and interpretable basis. In *NeurIPS*, 2023.
- 524 Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64  
525 words: Long-term forecasting with transformers. In *ICLR*, 2023.
- 526 Pan Shang and Lingchen Kong.  $\ell_1$ -norm quantile regression screening rule via the dual circumscribed  
527 sphere. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(10):6254–6263, 2021.
- 528 Lan Wang and Xuming He. Analysis of global and local optima of regularized quantile regression in  
529 high dimensions: A subgradient approach. *Economet. Theory*, 40(2):233–277, 2024.
- 530  
531  
532  
533  
534  
535  
536  
537  
538  
539

- 540 Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and  
541 Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. In *ICLR*,  
542 2024.
- 543 Xi Wang, Zhiwu Li, and W. M. Wonham. Real-time scheduling based on nonblocking supervisory  
544 control of state-tree structures. *IEEE Trans. Autom. Control*, 66(9):4230–4237, 2021.
- 545 Ines Wilms, Sumanta Basu, Jacob Bien, and David S Matteson. Sparse identification and estimation  
546 of large-scale vector autoregressive moving averages. *J. Am. Stat. Assoc.*, 118(541):571–582, 2023.
- 547 Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet:  
548 Temporal 2d-variation modeling for general time series analysis. In *ICLR*, 2023.
- 549 Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with  $10k$  parameters. In *ICLR*,  
550 2024.
- 551 Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. Temporal regularized matrix factorization for  
552 high-dimensional time series prediction. In *NeurIPS*, 2016.
- 553 Ashkan Zarnani, Soheila Karimi, and Petr Musilek. Quantile regression and clustering models of  
554 prediction intervals for weather forecasts: A comparative study. *Forecasting*, 1(1):169–188, 2019.
- 555 Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series  
556 forecasting? In *AAAI*, 2023.
- 557 Mingyang Zhang, Hongrong Yang, Yinliang Xu, and Hongbin Sun. Learning-based real-time  
558 aggregate flexibility provision and scheduling of electric vehicles. *IEEE Trans. Smart Grid*, 2024.
- 559 Wenjie Zhang, Hao Quan, and Dipti Srinivasan. An improved quantile regression neural network for  
560 probabilistic load forecasting. *IEEE Trans. Smart Grid*, 10(4):4425–4434, 2018.
- 561 Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for  
562 multivariate time series forecasting. In *ICLR*, 2022.
- 563 Zhuoli Zhao, Jiawen Xu, Juntao Guo, Qiang Ni, Biyun Chen, and Loi Lei Lai. Robust energy  
564 management for multi-microgrids based on distributed dynamic tube model predictive control.  
565 *IEEE Trans. Smart Grid*, 15(1):203–217, 2024.
- 566 Zihuai Zhao, Wanchun Liu, Daniel E Quevedo, Yonghui Li, and Branka Vucetic. Deep learning for  
567 wireless networked systems: a joint estimation-control-scheduling approach. *IEEE Internet Things*  
568 *J.*, 2023.
- 569 Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang.  
570 Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*, 2021.
- 571 Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency  
572 enhanced decomposed transformer for long-term series forecasting. In *ICML*, 2022.
- 573 Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis  
574 by pretrained lm. In *NeurIPS*, 2024.
- 575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593

## 594 A NOTATIONS

595

596 A comprehensive table of all notations used in this work is provided here. The following tables serve  
 597 as a reference to clarify the meaning and usage of each symbol throughout this study.  
 598

599

**Numbers and Arrays**

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

$x$	A scalar
$X$	A vector
$\mathbf{X}$	A matrix
$\text{diag}(\mathbf{a})$	A square, diagonal matrix with diagonal entries given by $\mathbf{a}$
$\beta$	The vector of regression coefficients to be estimated
$\lambda$	The regularization parameter. A sparse solution is obtained when $\lambda$ is large; otherwise, more features are retained
$\tau$	The quantile $\tau$ denotes the ratio that $100 \times \tau\%$ of predictions are greater than the ground truth, $\tau \in [0, 1]$
$\mathbf{w}$	The weighted vector, $w_i = (\alpha^{1/\gamma} + b\Delta)^\gamma$ , for $b = 0, \dots, L_{batch} - 1$ and $\Delta = (\xi^{1/\gamma} - \alpha^{1/\gamma}) / (L_{batch} - 1)$
$\mathbf{T}$	The significant periods for global identification, $\mathbf{T} = [T_1, T_2, \dots, T_n]$

**Sets**

$\mathbb{R}$	The set of real numbers
$\Theta$	The set of basis vectors
$\{0, 1, 2, 3\}$	The set containing 0, 1, 2, and 3

**Indexing**

$\mathcal{B}_b$	The $b$ th test batch, the test dataset is split into batches
$X^i$	The values of $\mathbf{X}$ at feature dimension $i$
$\hat{X}^i$	The input sequence after elimination of outliers by knowledge fusion strategy, $\hat{X}^i = \mathbf{w}X^i + (1 - \mathbf{w})\hat{Y}g^i$
$\tilde{X}_{l_b,t}^i$	The output sequence for the $b$ th test batch $\mathcal{B}_b$ at feature dimension $i$ derived by STL
$\mathbf{X}_t$	The historical sequence with a look back window of $H$ , $\mathbf{X}_t = [X_{t-H+1}, X_{t-H+2}, \dots, X_t] \in \mathbb{R}^{H \times d_x}$
$Y^i$	The values of $\mathbf{Y}$ at feature dimension $i$
$Y_t$	The multivariate values of $d_y$ distinct series at timestamp $t$ , $Y_t \in \mathbb{R}^{d_y}$
$Y_g^i$	The global identification at feature dimension $i$
$Y_{l_b,t}^i$	The local identification for the $b$ th test batch $\mathcal{B}_b$ at feature dimension $i$
$\mathbf{Y}_t$	The future sequence with a horizon window of $P$ , $\mathbf{Y}_t = [Y_{t+1}, Y_{t+2}, \dots, Y_{t+P}] \in \mathbb{R}^{P \times d_y}$

648	$\hat{Y}_t$	The prediction sequence with a horizon window of $P$ , $\hat{Y}_t = [\hat{Y}_{t+1}, \hat{Y}_{t+2}, \dots, \hat{Y}_{t+P}] \in \mathbb{R}^{P \times d_y}$
649		
650		
651	$\mathbf{T}_{l_b,t}^i$	The significant periods at feature dimension $i$ for a test batch
652	$\Theta_g^i$	The global basis vectors $\Theta_g^i = [\sin(\frac{2\pi t^i}{T^i}), \cos(\frac{2\pi t^i}{T^i})]$
653	$\hat{\Theta}_g^i$	The significant global basis vectors at feature dimension $i$
654		obtained by SOR filtering method
655		
656	$\Theta_{l_b,t}^i$	The local basis vectors at feature dimension $i$ for the $b$ th test
657		batch $\mathcal{B}_b$ , $\Theta_{l_b,t}^i = [\sin(\frac{2\pi t_{l_b,t}^i}{T_{l_b,t}^i}), \cos(\frac{2\pi t_{l_b,t}^i}{T_{l_b,t}^i})]$
658		
659	$\hat{\Theta}_{l_b,t}^i$	The significant local basis vectors for the $b$ th test batch $\mathcal{B}_b$
660		at feature dimension $i$ obtained by SOR filtering method
661	$\beta_g^i$	The global regression coefficients vector at feature dimension
662		$i$ to be estimated
663		
664	$\beta_{l_b,t}^i$	The local regression coefficients vector at feature dimension
665		$i$ to be estimated
666		

### Functions

668	$\ \mathbf{x}\ _1$	$L_1$ norm of $\mathbf{x}$
669	$\mathbb{I}(\text{condition})$	is 1 if the condition is true, 0 otherwise
670		
671	$\rho_\tau(u)$	The Koenker and Bassett check function, $\rho_\tau(u) = u(\tau -$
672		$\mathbb{I}(u < 0))$
673		
674		

## B PROBLEM VISUALIZATION AND PERFORMANCE ANALYSIS

To visually illustrate the problem, we set the quantile  $\tau = 1$  and explain the following three figures. The grey-shaded areas present the difference between the prediction and the ground truth, also known as economic cost.

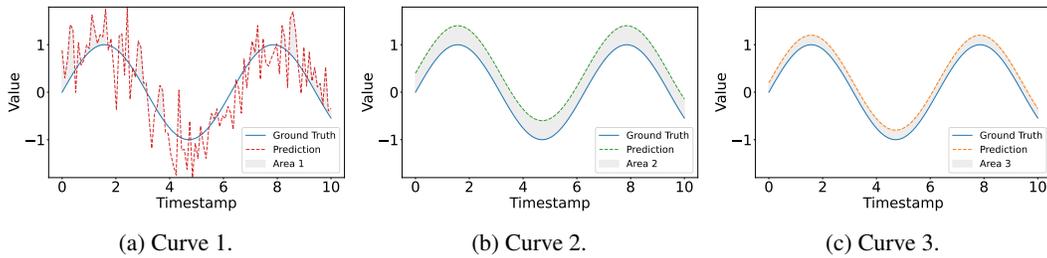


Figure 4: Comparison of three predictive model performance.

Figure 4(a) shows that 50% of the predicted values exceed the ground truth. This model is difficult to meet practical demands as it cannot guarantee that all predicted values will be greater than the actual demand, potentially leading to insufficient resources.

Figure 4(b) shows that 100% of the predicted values exceed the ground truth, which meets the requirements for resource scheduling. It effectively prevents task failure by ensuring adequate resource allocation. Therefore, the results in Figure 4(b) are better than in Figure 4(a).

Figure 4(c) shows another prediction value curve. The grey area is smaller, indicating that the additional cost brought by the model, after meeting the requirements, is lower, meaning higher resource utilization efficiency. Thus, the results in Figure 4(c) are better than in Figure 4(a) and Figure 4(b).

## C INTRODUCTION TO THE KEY COMPONENTS

**Sparse System Identification** Sparse System Identification (SSI) (Brunton et al., 2016; Wilms et al., 2023; Fasel et al., 2022) infers the underlying dynamics and the algebraic representation of a control system from observed data. This methodology is rooted in the principle of parsimony, seeking to disentangle the most concise and informative representation of system behavior by selecting a minimal set of relevant basis vectors from a predefined library.

SSI is inherently characterized by the basis function library, which commonly encompasses polynomial series, trigonometric functions, and other suitable functional forms. Given this library, we are able to identify a sparse combination of basis vectors that best approximates the system dynamics. This is achieved by minimizing a sparsity-promoting penalty term within the optimization framework. SSI eliminates superfluous terms, thereby yielding compact and interpretable dynamical models.

SSI has demonstrated remarkable progress in revealing hidden system dynamics within data, resulting in highly accurate predictions in various contexts (Liu et al., 2024). However, the efficacy of SSI for business-driven identification prediction is not without challenges. Direct application of the method without tailored strategies for basis vectors design may lead to accumulation of errors over extended prediction horizons, compromising its suitability for long-range predictions.

**$L_1$ -norm Quantile Regression** Traditional regression analysis focuses on the relationship between covariates and the conditional expectation of the response variable. Models derived from such analyses, such as linear or generalized linear regressions, aim to construct a function that predicts the conditional mean of the response given observed covariate values. While these methods effectively capture the central tendency of the response distribution in many instances, they provide a single-dimensional view and can be susceptible to outliers or data heterogeneity. To address these limitations, Quantile Regression (QR) has emerged as an alternative approach, particularly when interest lies in other aspects of the response distribution.

Quantile regression extends the scope of conventional regression by investigating the association between covariates and conditional quantiles of the response variable rather than solely its conditional expectation. Specifically, quantile regression models estimate the conditional quantile of the dependent variable given the value of the independent variable. QR offers a comprehensive portrayal of the response distribution across different quantiles, encompassing tail behavior, asymmetry, and potential non-linear effects.

Recently,  $L_1$ -norm Quantile Regression ( $L_1$ QR) has been proposed as a robust and flexible approach for modeling conditional quantiles (Li & Zhu, 2008; Shang & Kong, 2021).  $L_1$ QR employs a penalty term based on the absolute values of the regression coefficients, thereby promoting sparsity in the model. Unlike the  $L_2$ -norm, some regression coefficients derived by the  $L_1$ -norm likely to be estimated as zero. This facilitates variable selection, particularly in high-dimensional settings, by identifying a subset of key covariates with significant influence on specific quantiles. For a given quantile  $\tau$ , the objective function of  $L_1$ QR is formulated as:

$$\min_{\beta} \sum_{i=1}^n \rho_{\tau}(Y^i - X^i \beta) + \lambda \|\beta\|_1,$$

where  $y^i$  is the response value for the  $i$ th observation,  $X^i$  represents the corresponding covariate values,  $\beta$  is the vector of regression coefficients to be estimated, and  $\rho_{\tau}(u)$  is the Koenker and Bassett check function (Koenker & Bassett Jr, 1978), that is

$$\rho_{\tau}(u) = \begin{cases} \tau \cdot u & \text{if } u > 0, \\ -(1 - \tau) \cdot u & \text{otherwise.} \end{cases}$$

**Fourier Transformation** The Fourier transform (FT) is a mathematical tool used to convert signals from the time domain to the frequency domain, revealing the underlying periodic components in a signal. Mathematically, the Fourier transform of  $f(t)$  is given by:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt,$$

756 where  $\omega$  is the angular frequency,  $j$  is the imaginary unit, and  $e$  is the base of the natural logarithm.  
 757 The transformed function  $F(\omega)$  represents the amplitude and phase of each frequency component in  
 758 the original signal, enabling the identification of dominant frequencies and their relative contributions  
 759 to the overall signal structure. For digital signals or sampled time series data, the continuous-time  
 760 Fourier transform is replaced by the Discrete Fourier Transform (DFT). The DFT computes the  
 761 frequency-domain representation of a finite-length discrete-time signal  $x_n$ :

$$762 \quad X_k = \sum_{n=0}^{N-1} x_n e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1,$$

763 where  $X_k$  represents the amplitude and phase of the  $k$ th frequency component in the discrete signal,  
 764 and the angular frequency  $\omega = 2\pi k/N$  corresponds to the  $k$ th harmonic of the fundamental frequency  
 765  $\omega_0 = 2\pi/N$ .

766 While the DFT is a powerful tool for frequency analysis, its computational complexity is  $\mathcal{O}(n^2)$ ,  
 767 which can become prohibitively expensive for large datasets. The Fast Fourier Transform (FFT) is  
 768 an algorithmic innovation that significantly reduces the computational cost of computing the DFT,  
 769 achieving a time complexity of  $\mathcal{O}(n \log n)$ . All waveforms are actually just the sum of simple  
 770 sinusoids of different frequencies. The FFT exploits the symmetries and periodicity inherent in the  
 771 DFT formula to compute the transform more efficiently, making it the preferred choice for practical  
 772 applications involving large time series datasets.

773 **Stepwise Orthogonal Regression** Stepwise Orthogonal Regression (SOR) is a sequential feature  
 774 selection algorithm specifically tailored for model building in the context of time series analysis  
 775 and forecasting. SOR combines the principles of forward selection and orthogonal least squares to  
 776 efficiently identify and incorporate the most influential variables into a model.

777 The SOR algorithm proceeds in a stepwise manner, iteratively adding one predictor at a time to an  
 778 initially empty model. At each iteration, the algorithm selects the variable that maximally improves  
 779 the model fit while minimizing collinearity issues through the use of orthogonal projections. The  
 780 improvement in model fit is typically assessed via a goodness-of-fit criterion, such as the reduction  
 781 in the sum of squared errors or the increase in the coefficient of determination. Once a variable is  
 782 added, its contribution is made orthogonal to the existing model terms, ensuring that subsequent  
 783 selections are independent and do not merely replicate previously captured information. This process  
 784 continues until a stopping criterion is met, which could be based on a predefined maximum number  
 785 of predictors, a threshold for the improvement in model fit, or a desired level of explained variance.

786 In the realm of quantile regression for time series forecasting, SOR can be employed to select the most  
 787 relevant predictors for each quantile level of interest. Given the potential heterogeneity in the effects  
 788 of different variables across different quantiles, SOR enables the construction of separate, tailored  
 789 models for each quantile, ensuring that each model includes only the most influential predictors for  
 790 that specific quantile. As a result, SOR-enhanced quantile regression models provide more accurate  
 791 and reliable forecasts across the entire conditional distribution of the time series.

792 **Seasonal-Trend decomposition using LOESS** Seasonal-Trend decomposition using LOESS (STL)  
 793 is a nonparametric filtering procedure, proposed by Cleveland (Cleveland et al., 1990), with LOcally  
 794 Estimated Scatterplot Smoothing (LOESS) used for estimating nonlinear relationships. STL has  
 795 gained popularity due to its ability to handle diverse types of time series data, including those  
 796 exhibiting irregular seasonal patterns and time-varying trends. The simplicity of STL allows fast  
 797 computation for long sequence. STL decomposes time series  $X$  into three additive components,  
 798 namely trend  $T$ , seasonal  $S$ , and residual  $R$ .

$$803 \quad X = T + S + R,$$

804 in which the trend component  $T$  captures the long-term behavior of a time series, embodying its  
 805 overall direction and pattern over time. Seasonality  $S$  refers to recurring patterns or cycles occurring  
 806 within fixed time frames, such as daily, weekly, or annual patterns. The residual component  $R$ ,  
 807 also known as the remainder or irregular component, represents unexplained time series variations  
 808 after removing the trend and seasonal components. It encapsulates random fluctuations, noise, or  
 809 irregularities not accounted for by the trend and seasonal patterns.

STL decomposition serves as a critical preprocessing step in time series forecasting models. By decomposing the raw time series into its constituent parts, we gain deeper insights into the underlying structure and identify potential patterns and anomalies. This decomposition facilitates the development of more accurate prediction models by modeling each component separately and incorporate relevant features into the forecasting framework.

## D EXPERIMENTAL DETAILS

In this section, we provide a comprehensive introduction of experiments, including 8 real-world datasets, 7 baselines and 3 evaluation metrics.

### D.1 DATASETS

To evaluate the performance of prediction models, the experiments are conducted on 8 well-established datasets, involving Electricity, Exchange rate, Weather, Illness and CPU usage. The detailed description is presented in Table 4.

**Electricity Transformer Temperature (ETT) (Zhou et al., 2021)** It contains data from 2 power transformers in two regions of a province in China, recorded every minute (ETTh). Hourly data (ETTm) is also available. The data includes 8 characteristics, including date, target predicted value “oil temperature” and 6 different types of external power load characteristics <sup>1</sup>.

**Exchange (Lai et al., 2018)** It contains exchange rate data from 8 countries (Australia, British, Canada, China, Japan, New Zealand, Singapore, and Switzerland), recorded every day ranging from 1990 to 2016 <sup>2</sup>.

**Weather** From a weather station of the Max Planck Biogeochemistry Institute in German, it aggregates 21 meteorological variables including humidity and temperature, sampled every 10 minutes throughout 2020 <sup>3</sup>.

**ILI** It illustrates the ratio of patients presenting with influenza-like illness relative to the total patient population. The dataset encompasses weekly data sourced from the Centers for Disease Control and Prevention (CDC) of the United States, spanning from 2002 through 2021 <sup>4</sup>.

**CPU** It collects the CPU usage from a park of Huawei Technologies Co., Ltd. The sampling interval is 1 minute, and the dataset spans from May 8th to May 27th, 2023.

Table 4: Summary of real-world datasets.

Dataset	Frequency	Dim.	Series Length	Dataset Size	Information
ETTh1	Hourly	7	{96, 192, 336, 720}	(10080, 1440, 2880)	Electricity
ETTh2	Hourly	7	{96, 192, 336, 720}	(10080, 1440, 2880)	Electricity
ETTm1	15min	7	{96, 192, 336, 720}	(40320, 5760, 11520)	Electricity
ETTm1	15min	7	{96, 192, 336, 720}	(40320, 5760, 11520)	Electricity
Exchange	Daily	8	{96, 192, 336, 720}	(5120, 665, 1422)	Exchange rate
Weather	10min	21	{96, 192, 336, 720}	(36792, 5271, 10540)	Weather
ILI	Weekly	7	{24, 36, 48, 60}	(617, 74, 170)	National illness
CPU	1min	1	{96, 192, 336, 720}	(20160, 2880, 5760)	CPU usage

<sup>1</sup><https://github.com/zhouhaoyi/ETDataset>

<sup>2</sup><https://github.com/laiguokun/multivariate-time-series-data>

<sup>3</sup><https://www.bgc-jena.mpg.de/wetter/>

<sup>4</sup><https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html>

## 864 D.2 BASELINES

865  
866 Time series forecasting modeling is a long-standing research problem. From early statistical models  
867 to later deep models, many new models have been proposed for sequence modeling and time series  
868 forecasting. However, long-term forecasting still suffers from a lack of accuracy. Considering the  
869 challenges of traditional statistical methods in long-term forecasting, the baselines we compare  
870 are mainly neural network methods. Specifically, we consider the latest time series forecasting  
871 methods, including TimeMixer (Wang et al., 2024), TimesNet (Wu et al., 2023), Basisformer (Ni  
872 et al., 2023), iTransformer (Liu et al., 2023), PatchTST (Nie et al., 2023), Dlinear (Zeng et al., 2023),  
873 and FEDformer (Zhou et al., 2022). The details are introduced as follows

874 **TimeMixer** TimeMixer is an architecture based entirely on multilayer perceptual machines. It  
875 consists of past decoupled mixing and future multi-predictor mixing modules, which decompose multi-  
876 scale sequences and integrate multiple predictors for prediction, respectively. TimeMixer provides  
877 insights, that is, time series exhibit different patterns at different sampling scales. Its scheduling  
878 implementation is public available at <https://github.com/kwuking/TimeMixer>.  
879

880 **TimesNet** TimesNet is an innovative time series analysis method that converts a one-dimensional  
881 series into a multi-periodic two-dimensional tensor, which maps the time series into a two-  
882 dimensional space and enhances the representation of periodic features. It decomposes com-  
883 plex temporal dynamics into intra-periodic fluctuations and inter-periodic trends, and thus per-  
884 forms better on datasets with significant periodic structure. Code is available at the repository  
885 <https://github.com/thuml/Time-Series-Library>.  
886

887 **Basisformer** BasisFormer is an end-to-end time series forecasting architecture that designs learn-  
888 able and interpretable substrates for forecasting. It consists of three core components: acquiring the  
889 substrate through adaptive self-supervised learning; computing the similarity coefficients between the  
890 time series and the substrate in historical perspective through a two-way cross-attention mechanism;  
891 and selecting the substrate based on the similarity coefficients to generate accurate future forecasts.  
892 The source code can be found at <https://github.com/nz15116190/Basisformer>.

893 **iTransformer** iTransformer is an advanced model designed for time series forecasting. Utilizing an  
894 “inverted Transformer” architecture, iTransformer has demonstrated great effectiveness in forecasting  
895 time series data, especially for long term forecasting problems with complex periodicity, trends, and  
896 non-stationarity. The source code is available at the repository [https://github.com/thuml/](https://github.com/thuml/iTransformer)  
897 [iTransformer](https://github.com/thuml/iTransformer).  
898

899 **PatchTST** PatchTST is a powerful Transformer-based model specialized in multivariate time  
900 series prediction and self-supervised representation learning. There are two core components: (i)  
901 segmentation of the time series into subsequence patches, which are then provided to the Transformer;  
902 and (ii) channel independence, where each channel represents a univariate time series with shared  
903 embeddings and Transformer weights for all series. The components in PatchTST greatly improve the  
904 accuracy of long-term forecasts. Code of PatchTST is present at the repository [https://github.](https://github.com/yuqinie98/PatchTST)  
905 [com/yuqinie98/PatchTST](https://github.com/yuqinie98/PatchTST).

906 **Dlinear** DLinear represents a straightforward neural network prediction model that ingeniously  
907 integrates time series decomposition with linear layers, unlike Transformer-based models. The  
908 essence of DLinear lies in its direct multi-step forecasting approach, decomposing historical time  
909 series data into trend and remainder components, each modeled through separate single-layer linear  
910 networks to achieve predictive outputs. The source code for DLinear is openly accessible at [https:](https://github.com/cure-lab/LTSF-Linear)  
911 [//github.com/cure-lab/LTSF-Linear](https://github.com/cure-lab/LTSF-Linear).  
912

913 **FEDformer** FEDformer is a Transformer-based model that is designed to tackle the long-term time  
914 series prediction issue by incorporating Fourier transforms and wavelet basis vectors to enhance the  
915 effectiveness of the model. The time complexity of FEDformer is  $\mathcal{O}(L)$ , where  $L$  is the length of  
916 the sequence, which means that FEDformer can significantly improve the computational efficiency  
917 when dealing with long sequences compared to traditional methods. Its implementation is provided  
in <https://github.com/MAZiqing/FEDformer>.

### D.3 EVALUATION METRICS

Due to the practical need of specific task, the business-driven identification prediction, we adopt three metrics in the experiments, including PMSE (Positive Mean Square Error), PMAE (Positive Mean Absolute Error), and QRE (Quantile regression Error).

Concretely, given the ground truth of future sequence  $\mathbf{Y}_t^i = [Y_{t+1}^i, Y_{t+2}^i, \dots, Y_{t+P}^i]$  with  $Y_t^i = [Y_{t+1}^i, Y_{t+2}^i, \dots, Y_{t+P}^i]$ , and the prediction sequence  $\hat{\mathbf{Y}}_t^i = [\hat{Y}_{t+1}^i, \hat{Y}_{t+2}^i, \dots, \hat{Y}_{t+P}^i]$  with  $\hat{Y}_t^i = [\hat{Y}_{t+1}^i, \hat{Y}_{t+2}^i, \dots, \hat{Y}_{t+P}^i]$ , the metrics are given as follows

$$\begin{aligned}
 PMSE &= \sum_{i=1}^{d_y} \sum_{j=t+1}^{t+P} (\hat{Y}_j^i - Y_j^i)^2 \cdot \mathbb{I}(\hat{Y}_j^i \geq Y_j^i), \\
 PMAE &= \sum_{i=1}^{d_y} \sum_{j=t+1}^{t+P} |\hat{Y}_j^i - Y_j^i| \cdot \mathbb{I}(\hat{Y}_j^i \geq Y_j^i), \\
 QRE &= \frac{1}{d_y \times P} \sum_{i=1}^{d_y} \sum_{j=t+1}^{t+P} \mathbb{I}(\hat{Y}_j^i \geq Y_j^i).
 \end{aligned} \tag{8}$$

### D.4 IMPLEMENTATION DETAILS

All baselines models are trained with the QRE loss, using Adam optimizer with an initial learning rate of  $10^{-4}$ . Batchsize is set to 32 as default. If no reduction in validation loss is observed, training is stopped prematurely after 3 epochs. Then, the model with least validation loss is saved for further evaluation.

For fair comparison, we assign the same search space to the common parameters in each model. And the best performance under the search space is reported. The seed equals to 2024. The data segmentation ratio is 7:1:2, and since the validation set data is often used in the model training stage and there is no validation stage in our algorithm, it is 8:2 for our identification algorithm, shown in Figure 5. In this study, the adaptive sparse group lasso (asgl) Python package developed by Mendez-Civieta et al. (2021), is adopted for quantile regression analysis.

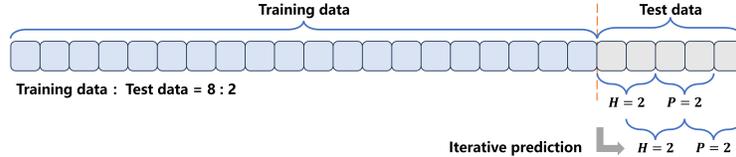


Figure 5: Data division and iterative identification schematic diagram.

Concretely, we adopted a uniform strategy in our experimental design, where the 6 methods (TimeMixer, TimesNet, iTransformer, PatchTST, DLinear, and FEDformer) employees the tools provided by the Python library Time Series Library (TSLib)<sup>5</sup> to perform the time series forecasting task. Moreover, the Basisformer model is performed with its official implementation.

### D.5 DETAILED RESULTS

For the resource scheduling task, we design an innovative approach, also known as iQR, that focuses on the Quantile Regression Error (QRE) as a pivotal performance metric. In order to investigate the model performance under different parameter settings, the loss function is set with different quantiles ( $\tau = 1.0, 0.95, 0.9$ ). The primary goal is to evaluate whether the identified values consistently exceed the ground truth. The Positive Mean Square Error (PMSE) and Positive Mean Absolute Error (PMAE) of these predictions are then minimized under this constraint.

<sup>5</sup><https://github.com/thuml/Time-Series-Library>

Under the tightest constraint, i.e., all predictions must exceed the actual values, our model iQR performs well. In several datasets such as ETT, Exchange, and Weather, there is still a low PMSE and PMAE for a QRE of 1. This suggests that not only are the predictions generally higher than the true values, but the error in these over-valued predictions (measured in terms of the PMSE and PMAE) is also minimized. For example, as shown in Table 5, in a dataset with a history length of 96 and predicted lengths varying from 96 to 720, the iQR model demonstrates high accuracy and low error, outperforming several competing methods including TimeMixer, and FEDformer.

Then, the constraint is relaxed to allow for a small percentage of predictions to be lower than the ground truth, the quantile is set to 0.95. iQR still maintains the high satisfaction rate regarding the business demand, with PMSE and PMAE lower than many competitive baselines. The results in Table 6 indicates that the model provides highly accurate predictions even existing a certain margin of error. The QRE is consistently higher than 0.95, demonstrating the robustness of designed algorithm.

Moreover, we further relax the constraints, that is, the quantile decreases to 0.9. iQR still satisfies the business demand, although exhibits the second best results on ETTh2 dataset. iQR achieves an optimal or suboptimal performance ratio of 92/96 across evaluated metrics, illustrating its broad effectiveness. Notably, even the prediction constraints are relaxed, the model maintains a low predictive error, thereby highlighting its particular superiority in the resource scheduling tasks.

Table 5: Resource scheduling results on eight datasets with historical length  $H = 96$  and prediction length  $P \in \{96, 192, 336, 720\}$ . The historical and prediction length are set to  $H = 36$  and  $P \in \{24, 36, 48, 60\}$  for ILI. Lower PMSE and PMAE for similar QRE indicate more accurate predictions and lower resource usage. If PMSE or PMAE is greater than 100, the symbol - is used to replace the original value. The quantile is set to 1. If the QRE in the result is less than the quantile, it will not be used for comparison. The ratio presents the percentage by which prediction exceeds ground truth. The best result are highlighted in **red bold**, and the second best results are marked with a blue underline.

Methods	IMP			Ours			TimeMixer			TimesNet			Basisformer			iTransformer			PatchTST			Dlinear			FEDformer					
Metrics	PMSE	PMAE	QRE	PMSE	PMAE	QRE	PMSE	PMAE	QRE	PMSE	PMAE	QRE	PMSE	PMAE	QRE	PMSE	PMAE	QRE	PMSE	PMAE	QRE	PMSE	PMAE	QRE	PMSE	PMAE	QRE			
ETTh	96	78.6%	51.4%	0.0%	<b>3.26</b>	<b>1.61</b>	<b>1.0</b>	-	-	<b>1.0</b>	3.66	1.53	0.97	-	-	<b>1.0</b>	16.01	<u>3.31</u>	<b>1.0</b>	79.24	7.5	<b>1.0</b>	0.49	0.49	0.67	<u>15.26</u>	3.73	<b>1.0</b>		
	192	81.3%	58.3%	0.0%	<b>3.23</b>	<b>1.48</b>	<b>1.0</b>	-	-	<b>1.0</b>	4.34	1.65	0.95	-	-	<b>1.0</b>	18.54	<u>3.55</u>	<b>1.0</b>	94.5	8.18	<b>1.0</b>	0.54	0.52	0.66	<u>17.3</u>	3.97	<b>1.0</b>		
	336	83.5%	64.7%	0.0%	<b>3.12</b>	<b>1.47</b>	<b>1.0</b>	-	-	<b>1.0</b>	3.72	1.5	0.94	-	-	<b>1.0</b>	21.64	3.83	0.99	-	9.03	<b>1.0</b>	0.58	0.55	0.64	<u>18.95</u>	<u>4.16</u>	<b>1.0</b>		
	720	74.3%	52.1%	0.0%	<b>3.77</b>	<b>1.72</b>	<b>1.0</b>	-	-	<b>1.0</b>	4.35	1.63	0.92	-	-	<b>1.0</b>	25.39	4.11	0.99	-	10.35	<b>1.0</b>	0.68	0.61	0.6	<u>14.67</u>	<u>3.59</u>	<b>1.0</b>		
ETTh2	96	19.9%	9.7%	0.0%	<b>2.34</b>	<b>1.3</b>	<b>1.0</b>	-	-	<b>1.0</b>	0.97	0.75	0.95	-	21.22	<b>1.0</b>	<u>2.92</u>	<u>1.44</u>	<b>1.0</b>	25.31	4.37	<b>1.0</b>	0.24	0.36	0.76	38.13	5.94	<b>1.0</b>		
	192	28.8%	15.7%	0.0%	<b>2.35</b>	<b>1.29</b>	<b>1.0</b>	-	-	<b>1.0</b>	0.89	0.73	0.93	-	-	<b>1.0</b>	<u>3.3</u>	<u>1.53</u>	<b>1.0</b>	39.41	5.41	<b>1.0</b>	0.27	0.38	0.74	38.82	6.04	<b>1.0</b>		
	336	91.6%	74.0%	0.0%	<b>3.22</b>	<b>1.57</b>	<b>1.0</b>	-	-	<b>1.0</b>	0.86	0.7	0.91	-	-	<b>1.0</b>	3.64	1.62	0.99	59.13	6.62	<b>1.0</b>	0.29	0.39	0.73	<u>38.14</u>	<u>6.03</u>	<b>1.0</b>		
	720	87.1%	67.1%	0.0%	<b>3.36</b>	<b>1.62</b>	<b>1.0</b>	-	-	<b>1.0</b>	0.95	0.75	0.91	-	-	<b>1.0</b>	4.41	1.79	0.99	84.73	7.97	<b>1.0</b>	0.34	0.43	0.71	<u>26.06</u>	<u>4.93</u>	<b>1.0</b>		
ETTh3	96	85.8%	62.9%	0.0%	<b>2.57</b>	<b>1.4</b>	<b>1.0</b>	-	-	<b>1.0</b>	20.23	<u>3.77</u>	<b>1.0</b>	-	-	<b>1.0</b>	71.87	6.81	<b>1.0</b>	-	36.8	<b>1.0</b>	0.82	0.73	0.9	<u>18.11</u>	4.09	<b>1.0</b>		
	192	83.8%	62.9%	0.0%	<b>2.88</b>	<b>1.5</b>	<b>1.0</b>	-	-	<b>1.0</b>	24.3	4.07	<b>1.0</b>	-	-	<b>1.0</b>	83.69	7.35	<b>1.0</b>	-	46.74	<b>1.0</b>	0.9	0.75	0.89	<u>17.75</u>	<u>4.04</u>	<b>1.0</b>		
	336	81.1%	57.9%	0.0%	<b>3.32</b>	<b>1.64</b>	<b>1.0</b>	-	-	<b>1.0</b>	21.7	<u>3.9</u>	<b>1.0</b>	-	-	<b>1.0</b>	99.42	8.02	<b>1.0</b>	-	56.56	<b>1.0</b>	0.97	0.77	0.88	<u>17.6</u>	4.03	<b>1.0</b>		
	720	78.7%	56.4%	0.0%	<b>3.5</b>	<b>1.68</b>	<b>1.0</b>	-	-	<b>1.0</b>	23.16	4.04	0.99	-	-	<b>1.0</b>	-	8.82	<b>1.0</b>	-	68.32	<b>1.0</b>	1.05	0.8	0.86	<u>16.46</u>	<u>3.85</u>	<b>1.0</b>		
ETTh4	96	91.4%	70.9%	0.0%	<b>1.21</b>	<b>0.94</b>	<b>1.0</b>	-	-	<b>1.0</b>	3.21	1.48	0.99	-	-	<b>1.0</b>	<u>14.08</u>	<u>3.23</u>	<b>1.0</b>	-	21.38	<b>1.0</b>	0.57	0.67	0.97	29.94	5.24	<b>1.0</b>		
	192	80.2%	54.9%	0.0%	<b>3.43</b>	<b>1.62</b>	<b>1.0</b>	-	-	<b>1.0</b>	3.45	1.58	0.99	-	-	<b>1.0</b>	<u>17.32</u>	<u>3.59</u>	<b>1.0</b>	-	26.25	<b>1.0</b>	0.67	0.73	0.97	31.79	5.39	<b>1.0</b>		
	336	83.1%	58.2%	0.0%	<b>3.42</b>	<b>1.62</b>	<b>1.0</b>	-	-	<b>1.0</b>	4.14	1.7	0.99	-	-	<b>1.0</b>	<u>20.26</u>	<u>3.88</u>	<b>1.0</b>	-	33.11	<b>1.0</b>	0.74	0.76	0.96	42.99	6.26	<b>1.0</b>		
	720	84.4%	60.0%	0.0%	<b>3.55</b>	<b>1.64</b>	<b>1.0</b>	-	-	<b>1.0</b>	4.77	1.82	0.98	-	-	<b>1.0</b>	<u>22.73</u>	<u>4.1</u>	<b>1.0</b>	-	47.49	<b>1.0</b>	0.84	0.8	0.95	39.67	6.04	<b>1.0</b>		
Exchange	96	69.1%	41.5%	0.0%	<b>1.76</b>	<b>1.2</b>	<b>1.0</b>	-	-	<b>1.0</b>	0.26	0.37	0.81	-	33.75	<b>1.0</b>	0.88	0.74	0.98	6.21	<u>2.05</u>	<b>1.0</b>	0.11	0.24	0.65	<u>5.69</u>	2.32	<b>1.0</b>		
	192	75.8%	54.6%	0.0%	<b>1.71</b>	<b>1.14</b>	<b>1.0</b>	-	-	<b>1.0</b>	0.34	0.42	0.75	-	-	<b>1.0</b>	1.17	0.85	0.97	9.15	<u>2.51</u>	<b>1.0</b>	0.18	0.32	0.59	<u>7.06</u>	2.56	<b>1.0</b>		
	336	79.4%	57.3%	0.0%	<b>1.85</b>	<b>1.22</b>	<b>1.0</b>	-	-	<b>1.0</b>	0.66	0.61	0.84	-	-	<b>1.0</b>	1.6	1.01	0.96	15.1	3.19	<b>1.0</b>	0.3	0.41	0.52	<u>8.96</u>	<u>2.86</u>	<b>1.0</b>		
	720	77.7%	56.0%	0.0%	<b>3.01</b>	<b>1.53</b>	<b>1.0</b>	-	-	<b>1.0</b>	1.27	0.87	0.85	-	-	<b>1.0</b>	2.68	1.33	0.97	30.52	4.5	<b>1.0</b>	1.05	0.77	0.34	<u>13.51</u>	<u>3.48</u>	<b>1.0</b>		
Weather	96	95.7%	85.8%	0.0%	<b>30.9</b>	<b>2.37</b>	<b>1.0</b>	-	-	<b>1.0</b>	2.76	1.05	0.96	-	-	<b>1.0</b>	18.09	2.66	0.99	-	<u>16.73</u>	<b>1.0</b>	0.5	0.58	0.94	-	30.47	<b>1.0</b>		
	192	89.1%	78.3%	0.0%	<b>84.18</b>	<b>3.77</b>	<b>1.0</b>	-	-	<b>1.0</b>	3.21	1.13	0.94	-	-	<b>1.0</b>	21.46	2.88	0.99	-	<u>17.38</u>	<b>1.0</b>	0.57	0.62	0.93	-	30.71	<b>1.0</b>		
	336	89.6%	78.5%	0.0%	<b>84.44</b>	<b>3.84</b>	<b>1.0</b>	-	-	<b>1.0</b>	3.72	1.22	0.93	-	-	<b>1.0</b>	25.97	3.17	0.99	-	<u>17.87</u>	<b>1.0</b>	0.64	0.66	0.92	-	32.72	<b>1.0</b>		
	720	89.6%	78.3%	0.0%	<b>86.06</b>	<b>3.95</b>	<b>1.0</b>	-	-	<b>1.0</b>	4.73	1.3	0.92	-	-	<b>1.0</b>	29.49	3.42	0.99	-	<u>18.24</u>	<b>1.0</b>	0.74	0.71	0.91	-	34.61	<b>1.0</b>		
ILI	96	98.2%	85.9%	0.0%	<b>30.62</b>	<b>4.76</b>	<b>1.0</b>	-	-	<b>1.0</b>	<u>33.79</u>	<b>1.0</b>	6.03	1.83	0.6	-	34.41	<b>1.0</b>	7.69	2.02	0.7	-	8.61	0.99	4.79	1.66	0.49	29.72	4.75	0.96
	192	98.0%	85.9%	0.0%	<b>31.6</b>	<b>4.84</b>	<b>1.0</b>	-	-	<b>1.0</b>	66.32	<b>1.0</b>	6.81	1.99	0.61	-	<u>34.36</u>	<b>1.0</b>	7.06	1.93	0.68	-	8.53	0.99	4.82	1.66	0.48	31.62	4.97	0.97
	336	98.1%	86.3%	0.0%	<b>38.09</b>	<b>5.07</b>	<b>1.0</b>	-	-	<b>1.0</b>	76.31	<b>1.0</b>	5.73	1.72	0.59	-	<u>37.13</u>	<b>1.0</b>	4.49	1.55	0.63	64.81	6.48	0.99	4.11	1.47	0.46	26.03	4.48	0.97
	720	96.7%	82.3%	0.0%	<b>39.14</b>	<b>5.23</b>	<b>1.0</b>	-	-	<b>1.0</b>	<u>29.63</u>	<b>1.0</b>	4.75	1.69	0.57	-	<u>56.89</u>	<b>1.0</b>	3.71	1.44	0.58	52.08	5.91	0.98	4.23	1.48	0.44	22.65	4.18	0.96
CPU	96	89.8%	68.3%	0.0%	<b>2.0</b>	<b>1.29</b>	<b>1.0</b>	-	-	11.58	<b>1.0</b>	9.88	2.85	0.98	-	28.22	<b>1.0</b>	<u>19.61</u>	<u>4.07</u>	<b>1.0</b>	97.41	8.83	<b>1.0</b>	0.86	0.73	0.74	28.29	5.21	<b>1.0</b>	
	192	92.2%	71.7%	0.0%	<b>2.13</b>	<b>1.34</b>	<b>1.0</b>	-	-	16.85	<b>1.0</b>	11.28	3.0	0.97	-	55.6	<b>1.0</b>	<u>27.35</u>	<u>4.74</u>	<b>1.0</b>	-	13.66	<b>1.0</b>	0.98	0.8	0.71	29.04	5.26	<b>1.0</b>	
	336	92.3%	73.5%	0.0%	<b>2.24</b>	<b>1.38</b>	<b>1.0</b>	-	-	19.11	<b>1.0</b>	13.16	3.2	0.97	-	85.21	<b>1.0</b>	<u>33.27</u>	<u>5.21</u>	<b>1.0</b>	-	15.59	<b>1.0</b>	1.04	0.83	0.67	<u>28.91</u>	5.25	<b>1.0</b>	
	720	93.8%	71.3%	0.0%	<b>2.53</b>	<b>1.62</b>	<b>1.0</b>	-	-	20.85	<b>1.0</b>	12.59	2.99	0.93	-	46.05	<b>1.0</b>	<u>40.63</u>	<u>5.65</u>	<b>1.0</b>	-	17.18	<b>1.0</b>	0.93	0.77	0.57	-	11.6	<b>1.0</b>	
Ratio	-	-	-	100%	-	-	100%	-	-	9.38%	-	-	100%	-	-	50%	-	-	-	87.5%	-	-	0%	-	-	-	87.5%	-		
Count	-	-	-	96	-	-	34	-	-	4	-	-	34	-	-	37	-	-	-	34	-	-	0	-	-	49	-			

## D.6 RESULTS WITH DIFFERENT HISTORICAL SEQUENCE LENGTH

In this section, we further explore the model’s adaptability to varying historical sequence lengths  $H$  while keeping the future sequence length  $P$  fixed. Specifically, we evaluated the performance when the historical length  $H$  takes values from the set  $\{192, 336, 720\}$ , comparing these results with those of existing baseline models. The experimental outcomes demonstrate that across all tested historical lengths, our model iQR significantly outperforms other models in terms of predictive accuracy. This finding further substantiates the robustness of iQR when dealing with data of differing historical

Table 6: Resource scheduling results on eight datasets with historical length  $H = 96$  and prediction length  $P \in \{96, 192, 336, 720\}$ . The historical and prediction length are set to  $H = 36$  and  $P \in \{24, 36, 48, 60\}$  for ILI. The quantile is set to 0.95. If the QRE in the result is less than the quantile, it will not be used for comparison. The ratio presents the percentage by which prediction exceeds ground truth. The best result are highlighted in **red bold**, and the second best results are marked with a **blue underline**.

Methods	Ours			TimeMixer			TimesNet			Basisformer			iTransformer			PatchTST			Dlinear			FEDformer			
	PMSE	PMSE	QRE	PMSE	PMSE	QRE	PMSE	PMSE	QRE	PMSE	PMSE	QRE	PMSE	PMSE	QRE	PMSE	PMSE	QRE	PMSE	PMSE	QRE	PMSE	PMSE	QRE	
ETT <sub>h</sub>	96	<b>1.441</b>	<b>0.939</b>	<b>0.953</b>	1.727	1.025	0.938	2.673	1.256	0.938	1.655	1.084	0.949	1.94	1.114	0.943	1.644	0.992	0.929	0.514	0.513	0.678	1.433	0.927	0.867
	192	<b>1.644</b>	<b>1.045</b>	<b>0.955</b>	2.052	1.137	0.938	2.939	1.327	0.928	1.814	1.036	0.948	2.359	1.237	0.941	2.073	1.14	0.933	0.573	0.548	0.662	1.789	1.055	0.889
	336	<b>1.915</b>	<b>1.061</b>	<b>0.957</b>	2.769	1.309	0.938	2.807	1.28	0.907	<u>1.963</u>	<u>1.113</u>	<u>0.954</u>	2.848	1.366	0.939	2.618	1.304	0.93	0.631	0.579	0.641	2.484	1.273	0.926
	720	<b>2.231</b>	<b>1.17</b>	<b>0.95</b>	3.693	1.505	0.919	3.157	1.362	0.87	<b>2.001</b>	<b>1.106</b>	<b>0.952</b>	3.775	1.579	0.917	2.865	1.344	0.896	0.755	0.646	0.605	2.705	1.341	0.914
ETT <sub>l</sub>	96	<b>0.982</b>	<b>0.785</b>	<b>0.954</b>	0.608	0.584	0.933	0.754	0.664	0.921	1.011	0.737	0.943	0.691	0.641	0.941	0.686	0.635	0.941	0.267	0.385	0.765	<b>0.699</b>	0.688	0.944
	192	<b>1.197</b>	<b>0.864</b>	<b>0.965</b>	0.769	0.659	0.933	0.754	0.667	0.909	1.246	0.801	0.928	0.826	0.7	0.938	0.691	0.625	0.921	0.306	0.409	0.745	<b>0.93</b>	<b>0.829</b>	<b>0.959</b>
	336	<b>1.053</b>	<b>0.806</b>	<b>0.953</b>	0.942	0.757	0.943	0.785	0.676	0.893	1.217	0.817	0.901	0.892	0.739	0.93	0.79	0.675	0.915	0.317	0.418	0.732	<b>0.989</b>	<b>0.849</b>	<b>0.949</b>
	720	<b>1.512</b>	<b>1.044</b>	<b>0.99</b>	1.269	0.812	0.923	0.889	0.729	0.894	1.349	0.851	0.924	1.088	0.831	0.934	1.001	0.787	0.931	0.379	0.461	0.722	<u>2.058</u>	<u>1.217</u>	<u>0.951</u>
ETT <sub>l2</sub>	96	<b>1.855</b>	<b>1.118</b>	<b>0.951</b>	1.301	0.86	0.925	1.312	0.865	0.893	1.583	0.92	0.942	1.372	0.904	0.928	1.462	0.923	0.925	0.822	0.733	0.892	1.201	0.847	0.905
	192	<b>1.62</b>	<b>1.026</b>	<b>0.951</b>	1.508	0.929	0.932	1.523	0.947	0.909	1.892	<b>1.033</b>	<b>0.95</b>	1.663	1.014	0.93	1.701	1.006	0.925	0.92	0.766	0.885	1.42	0.931	0.908
	336	<b>1.887</b>	<b>1.128</b>	<b>0.955</b>	1.625	0.977	0.922	1.812	1.051	0.915	2.065	1.087	0.948	1.93	1.1	0.927	1.907	1.069	0.918	1.029	0.805	0.876	1.693	1.034	0.922
	720	<b>1.957</b>	<b>1.156</b>	<b>0.956</b>	2.23	1.168	0.936	2.379	1.252	0.93	2.419	1.204	0.949	2.465	1.263	0.926	2.416	1.235	0.925	1.144	0.85	0.862	1.876	1.089	0.893
ETT <sub>l2</sub>	96	0.595	0.629	<b>0.967</b>	0.379	0.459	0.92	0.414	0.493	0.93	0.636	0.592	0.946	0.44	0.523	0.939	0.358	0.458	0.931	<u>0.449</u>	<u>0.589</u>	0.955	<b>0.443</b>	<b>0.53</b>	<b>0.958</b>
	192	0.729	0.682	<b>0.96</b>	0.59	0.563	0.93	0.536	0.557	0.926	0.852	0.681	0.939	0.586	0.601	0.935	0.512	0.543	0.931	<b>0.565</b>	<b>0.661</b>	<b>0.955</b>	<b>0.66</b>	<b>0.647</b>	<b>0.96</b>
	336	0.825	<b>0.721</b>	<b>0.961</b>	0.919	0.673	0.925	0.692	0.636	0.926	1.279	0.841	0.943	0.753	0.679	0.933	0.698	0.636	0.936	<b>0.666</b>	<b>0.716</b>	<b>0.954</b>	<b>0.788</b>	<b>0.73</b>	<b>0.96</b>
	720	<b>0.785</b>	<b>0.715</b>	<b>0.982</b>	1.134	0.83	0.947	0.926	0.728	0.916	1.51	0.912	0.929	1.09	0.822	0.937	0.917	0.726	0.925	0.795	0.78	0.947	<u>1.023</u>	<u>0.86</u>	<u>0.967</u>
Exchange	96	<b>0.29</b>	<b>0.402</b>	<b>0.956</b>	0.449	0.525	0.969	0.285	0.407	0.805	0.493	0.552	<u>0.973</u>	<u>0.377</u>	<u>0.487</u>	0.96	0.48	0.53	0.968	0.128	0.268	0.652	0.878	0.799	<b>0.994</b>
	192	<b>0.422</b>	<b>0.511</b>	<b>0.977</b>	0.824	0.697	0.944	0.421	0.494	0.739	0.447	0.703	0.966	<b>0.688</b>	<b>0.657</b>	0.951	0.865	0.718	0.965	0.225	0.353	0.596	1.359	0.999	<b>0.996</b>
	336	<b>0.519</b>	<b>0.58</b>	<b>0.956</b>	1.998	1.128	<b>0.983</b>	0.767	0.691	0.837	2.801	1.097	0.979	<u>1.201</u>	<u>0.879</u>	0.951	1.643	0.996	0.968	0.312	0.422	0.524	2.361	1.331	<b>0.999</b>
	720	<b>0.822</b>	<b>0.754</b>	<b>0.953</b>	6.254	1.961	0.981	1.494	0.994	0.852	7.001	2.089	<u>0.994</u>	<u>2.751</u>	<u>1.357</u>	0.97	4.874	1.779	0.99	0.462	0.545	0.353	5.28	1.974	<b>0.999</b>
Weather	96	<b>1.079</b>	<b>0.789</b>	<b>0.966</b>	0.595	0.442	0.889	0.628	0.486	0.857	0.911	0.512	0.912	0.909	0.528	0.905	0.779	0.489	0.913	0.454	0.571	0.934	1.377	0.898	0.856
	192	<b>1.289</b>	<b>0.884</b>	<b>0.951</b>	0.786	0.566	0.907	0.74	0.541	0.835	1.136	0.597	0.898	1.25	0.651	0.899	1.021	0.586	0.901	0.533	0.621	0.925	1.436	0.923	0.845
	336	<b>1.709</b>	<b>1.075</b>	<b>0.958</b>	0.86	0.602	0.882	0.906	0.62	0.839	1.475	0.704	0.897	1.466	0.729	0.887	1.338	0.71	0.907	0.624	0.673	0.917	1.664	0.999	0.906
	720	<b>2.729</b>	<b>1.392</b>	<b>0.952</b>	1.212	0.742	0.899	1.027	0.691	0.812	1.871	0.813	0.898	1.771	0.839	0.888	1.794	0.847	0.905	0.744	0.735	0.906	2.43	1.27	0.934
ILI	24	<b>33.407</b>	<b>4.943</b>	<b>0.967</b>	18.574	3.395	0.876	3.859	1.641	0.566	10.712	2.592	0.881	3.159	1.396	0.567	10.857	2.513	0.877	1.49	1.01	0.44	6.417	2.135	0.77
	36	<b>54.821</b>	<b>6.386</b>	<b>0.956</b>	27.936	3.845	0.923	5.7	1.847	0.603	15.209	3.09	0.89	4.654	1.618	0.619	12.832	2.628	0.902	2.107	1.148	0.461	7.18	2.216	0.793
	48	<b>97.343</b>	<b>8.349</b>	<b>0.95</b>	21.284	3.639	0.898	6.217	1.829	0.592	15.746	3.175	0.9	7.857	2.062	0.676	20.766	3.135	0.914	2.722	1.382	0.477	9.23	2.477	0.824
	60	<b>95.065</b>	<b>8.22</b>	<b>0.95</b>	27.673	3.62	0.923	7.231	2.13	0.61	19.6	3.404	0.923	8.204	2.104	0.685	21.469	3.186	0.911	2.743	1.388	0.487	9.876	2.557	0.829
CPU	96	<b>2.766</b>	<b>1.512</b>	<b>0.952</b>	1.827	1.007	0.929	2.832	1.432	0.911	3.483	1.551	0.936	2.21	1.265	0.889	2.694	1.369	0.914	0.768	0.714	0.741	1.78	1.085	0.846
	192	<b>2.917</b>	<b>1.517</b>	<b>0.953</b>	2.259	1.153	0.916	4.417	1.837	0.922	4.042	1.692	0.915	2.318	1.296	0.856	3.597	1.631	0.91	0.801	0.744	0.714	2.379	1.279	0.876
	336	<b>3.096</b>	<b>1.548</b>	<b>0.955</b>	2.825	1.314	0.926	5.29	2.034	0.937	4.208	1.683	0.895	2.129	1.239	0.804	3.649	1.632	0.888	0.84	0.763	0.672	2.008	1.164	0.839
	720	<b>2.484</b>	<b>1.387</b>	<b>0.981</b>	3.343	1.493	0.916	10.061	2.735	0.886	5.906	1.939	0.907	3.041	1.45	0.762	4.705	1.817	0.886	0.635	0.667	0.569	2.396	1.335	0.834
Ratio	100%			9.375%			0%			21.875%			12.5%			12.5%			9.375%			31.25%			
Count	88			1			0			11			8			0			7			21			

Table 7: Resource scheduling results on eight datasets with historical length  $H = 96$  and prediction length  $P \in \{96, 192, 336, 720\}$ . The historical and prediction length are set to  $H = 36$  and  $P \in \{24, 36, 48, 60\}$  for ILI. The quantile is set to 0.9. If the QRE in the result is less than the quantile, it will not be used for comparison. The best result are highlighted in **red bold**, and the second best results are marked with a **blue underline**.

Methods	Ours			TimeMixer			TimesNet			Basisformer			iTransformer			PatchTST			Dlinear			FEDformer			
	PMSE	PMSE	QRE	PMSE	PMSE	QRE	PMSE	PMSE	QRE	PMSE	PMSE	QRE	PMSE	PMSE	QRE	PMSE	PMSE	QRE	PMSE	PMSE	QRE	PMSE	PMSE	QRE	
ETT <sub>h</sub>	96	<b>1.185</b>	<b>0.782</b>	<b>0.904</b>	1.209	0.81	0.884	1.514	0.951	0.884	<u>1.23</u>	<u>0.82</u>	<u>0.901</u>	1.287	0.874	0.887	1.187	0.801	0.874	0.57	0.546	0.66	1.106	0.764	0.771
	192	<b>1.239</b>	<b>0.842</b>	<b>0.904</b>	1.345	0.865	0.872	1.705	1.015	0.885	1.311	0.847	0.891	1.53	0.963	0.887	1.433	0.901	0.876	0.629	0.577	0.639	1.244	0.833	0.793
	336	<b>1.821</b>	<b>1.027</b>	<b>0.9</b>	1.547	0.943	0.875	2.308	1.131	0.849	1.847	0.868	0.884	1.816	1.052	0.883	1.674	0.991	0.864	0.753	0.645	0.604	1.531	0.969	0.866
	720	<b>1.798</b>	<b>1.032</b>	<b>0.902</b>	1.962	1.096	0.852	2.515	1.176	0.786	1.453	0.909	0.885	2.303	1.194	0.846	1.881	1.055	0.827	0.511	0.51	0.676	1.88	1.085	0.852
ETT <sub>l</sub>	96	<b>0.924</b>	<b>0.734</b>	<b>0.955</b>	0.697	0.643	0.82	0.648	0.62	0.831	0.782	0.629	0.899	0.49	0.519	0.889	0.421	0.474	0.872	0.3					

lengths. As depicted in Figure 8, under the condition of a fixed prediction length, iQR achieves higher QRE at each configuration of historical length. Unlike other methods, iQR does not rely on GPU resources, and its training duration is significantly shorter than other approaches, thereby highlighting its performance superiority across various parameter settings.

Table 8: QRE results for resource scheduling on ETTh1 dataset with historical length  $H = \{192, 336, 720\}$  and prediction length  $P = 96$ . The quantile is 0.9. If the QRE in the result is less than the quantile, it will not be used for comparison. The best result are highlighted in **red bold**.

Methods	H	P	PMSE	PMAE	QRE	Time(s)	GPU(GB)
iQR	192	96	1.12	0.74	<b>0.91</b>	1.15	0
iQR	336	96	1.19	0.79	<b>0.92</b>	1.18	0
iQR	720	96	1.22	0.83	<b>0.92</b>	1.21	0
TimeMixer	192	96	1.08	0.75	0.89	375.73	0.55
TimeMixer	336	96	1.01	0.72	0.87	382.28	0.94
TimeMixer	720	96	1.20	0.85	0.89	433.39	2.39
TimesNet	192	96	1.26	0.83	0.88	837.19	0.19
TimesNet	336	96	1.25	0.82	0.88	851.84	0.27
TimesNet	720	96	1.13	0.78	0.85	889.05	0.48
Basisformer	192	96	1.29	0.86	0.89	359.55	0.04
Basisformer	336	96	1.23	0.84	0.87	359.23	0.04
Basisformer	720	96	1.35	0.90	0.89	348.16	0.05
iTransformer	192	96	1.12	0.80	0.89	82.10	0.01
iTransformer	336	96	1.13	0.81	0.88	201.57	0.01
iTransformer	720	96	1.17	0.83	0.89	197.04	0.01
PatchTST	192	96	1.05	0.74	0.88	166.41	0.41
PatchTST	336	96	1.11	0.76	0.88	199.85	1.78
PatchTST	720	96	1.09	0.76	0.88	223.40	2.09
DLinear	192	96	0.46	0.48	0.67	107.57	0.01
DLinear	336	96	0.45	0.47	0.68	119.17	0.01
DLinear	720	96	0.45	0.47	0.67	127.84	0.01
FEDformer	192	96	1.17	0.82	0.87	1073.57	1.94
FEDformer	336	96	1.36	0.91	0.89	1102.01	2.23
FEDformer	720	96	1.39	0.91	0.86	1137.32	3.13

## D.7 RESULTS ON BIG DATASETS

To further explore the computational efficiency and scalability of the iQR framework, we paid particular attention to its performance when handling large-scale datasets. For this evaluation, the Bitcoin historical dataset provided on Kaggle is adopted<sup>6</sup>, which records Bitcoin transaction data every minute from 2012 to 2021. After preprocessing, the dataset reached a length of 3,330,541 records, involving 7 features.

In this assessment, we compared iQR with traditional time series analysis methods, such as AR and ARIMA, as well as deep learning approaches, including LSTM and Transformer. Notably, to ensure a fair comparison, we modified the traditional methods to a quantile version and introduced a quantile regression loss function for the deep learning models.

The test results in Table 9 demonstrated that iQR outperformed these traditional methods in terms of memory usage and processing speed. Specifically, when processing over three million data records, the training time for LSTM and Transformer was 100 and 300 times that of iQR, respectively. This significant difference highlights iQR’s capability to handle large datasets in resource-constrained environments, especially without the need for GPU resource support.

Furthermore, the lightweight nature of iQR allows it to complete training within seconds on CPUs, significantly reducing computational overhead. This increase in efficiency not only showcases the potential in practical applications but also proves its practical value in resource scheduling problems, especially in scenarios that require rapid and accurate predictions.

<sup>6</sup><https://www.kaggle.com/datasets/mczielinski/bitcoin-historical-data>

iQR framework has shown excellent computational efficiency and good scalability when dealing with large-scale datasets, making it an ideal choice for addressing resource scheduling problems, particularly in real-time application scenarios where computational resources are limited.

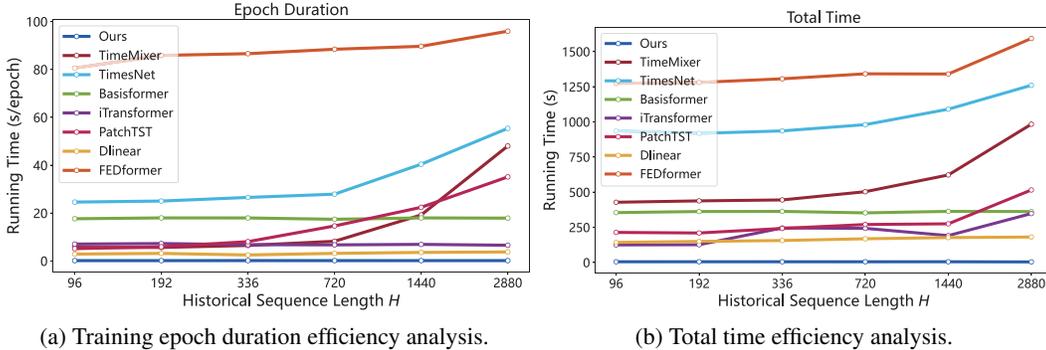


Figure 6: Additional running efficiency analysis.

Table 9: QRE results for resource scheduling on Bitcoin dataset with historical length  $H = 96$  and prediction length  $P = 96$ . If PMSE or PMAE is greater than 100, the symbol - is used to replace the original value. The quantile is 0.9. If the QRE in the result is less than the quantile, it will not be used for comparison. The best result are highlighted in **red bold**.

Methods	Length	Time(s)	GPU(GB)	PMSE	PMAE	QRE
iQR	$10^3$	0.13	0	1.55	1.04	<b>0.93</b>
iQR	$10^4$	1.14	0	1.17	0.87	<b>0.91</b>
iQR	$10^5$	15.21	0	1.28	1.00	<b>0.91</b>
iQR	$10^6$	101.84	0	1.49	1.29	<b>0.90</b>
QAR	$10^3$	1.03	0	-	43.84	0.15
QAR	$10^4$	2.41	0	-	-	0.16
QAR	$10^5$	19.15	0	-	-	0.11
QAR	$10^6$	166.52	0	-	-	0.07
QARIMA	$10^3$	1.67	0	-	16.07	0.59
QARIMA	$10^4$	4.92	0	-	17.52	0.74
QARIMA	$10^5$	20.39	0	-	-	0.38
QARIMA	$10^6$	204.45	0	-	-	0.52
LSTM	$10^3$	10.37	0.02	-	15.83	0.69
LSTM	$10^4$	101.49	0.02	-	-	0.50
LSTM	$10^5$	1179.03	0.02	-	-	0.95
LSTM	$10^6$	11706.33	0.02	-	-	0.90
GRU	$10^3$	10.5	0.02	-	29.75	0.81
GRU	$10^4$	105.73	0.02	-	-	0.79
GRU	$10^5$	1067.28	0.02	-	-	0.97
GRU	$10^6$	11790.11	0.02	-	-	0.97
Transformer	$10^3$	32.15	0.08	-	11.70	0.95
Transformer	$10^4$	302.93	0.08	-	-	0.25
Transformer	$10^5$	3147.43	0.08	-	-	0.98
Transformer	$10^6$	31278.46	0.08	-	-	0.97

## D.8 EFFICIENCY ANALYSIS

We further provide a detailed analysis of total time and one-epoch execution time to compare different models. Specifically, the total time of iQR remains relatively stable as the length of historical sequences increases, showing its efficiency in long sequence training. As shown in Figure 6, as the sequence length grows from 96 to 2880, the total time of iQR only increases from 1.082 s to 1.141 s. In addition, we take the single-channel total time in the iQR model as the one-epoch execution time, and the results show that the computational efficiency of iQR is similar within each channel, regardless of the large increase in sequence length.

For TimeMixer and TimesNet, under the same conditions, both the total time and the duration of individual epochs show a significant increase with sequence length, which directly leads to their poor performance in long sequence prediction tasks. Basisformer maintains similar performance across different scale sequences, outperforming other baselines, but the time spent is still higher than our method. Although PatchTST uses time series segmentation to improve learning efficiency, the result shows a significant increase in total time as the sequence length increases, revealing the efficiency limitations of the segmentation strategy for long sequence processing. This further emphasizes the applicability of iQR on data at different scales. Dlinear outperforms the transformer-based model in terms of less memory usage and time spent. FEDformer consistently exhibits the largest resource usage, showing its limitations in efficiency and scalability. In conclusion, iQR provides both efficient and resource-saving solutions for resource scheduling tasks.

## E PSEUDOCODE

We further provide pseudocode of the proposed algorithm iQR, offering a concise, step-by-step outline of its functionality and logic. The pseudocode can be found in Pseudocode 1.

---

### Algorithm 1: Business-driven identification prediction framework (iQR)

---

**Input:** Time series  $\mathbf{X}$ , historical length  $H$ , prediction length  $P$ , hyperparameters  $\alpha, \beta, \tau, \lambda$

**Output:** Prediction sequence  $\hat{\mathbf{Y}}$

```

for each channel  $i = 1, \dots, d_x$  do
    Split data  $X^i$  into a training set  $X_{train}^i$  and a test set  $X_{test}^i$ 
    /* Phase I: Global Identification */
    Perform data normalization
    Select the 1st set of basis vectors using FFT on  $X_{train}^i$ , generating global basis vectors  $\Theta_g$ 
    Select the 2nd set of basis vectors using SOR, resulting in the significant global basis  $\hat{\Theta}_g^i$ 
    Perform  $L_1$ QR to compute the global identification  $\hat{Y}_g^i$ 
    /* Phase II: Local Identification */
    for each test batch  $b = 1, \dots, N$  do
        Fuse global and local knowledge, yielding the fused sequence  $\hat{X}^i = \mathbf{w}X^i + (1 - \mathbf{w})Y_g^i$ 
        Select the 1st set of local basis using FFT on  $\hat{X}_{l_b,t}^i$ , yielding local basis vectors  $\Theta_{l_b,t}^i$ 
        Select the 2nd set of local basis using SOR, obtaining the significant local basis  $\hat{\Theta}_{l_b,t}^i$ 
        Enhance the input sequence using STL, resulting in  $\tilde{X}_{l_b,t}^i$ 
        Perform  $L_1$ QR to obtain the local identification  $\hat{Y}_{l_b,t}^i$ 
    /* Phase III: Concatenation */
Construct the complete prediction sequence  $\hat{\mathbf{Y}}$ 
return  $\hat{\mathbf{Y}}$ 

```

---

## F THEORETICAL PROOF

### F.1 PROOF OF LEMMA 2

*Proof.* Let  $E$  denote the error,  $Y$  represent the ground truth,  $\mathbf{X}$  be the basis function library, and  $\theta$  be the coefficients. Then, we have  $E = Y - \mathbf{X}\theta$ . Therefore, the optimization function  $O(\theta)$  can be written in the following form:

$$\begin{aligned}
 O(\theta) &= \frac{1}{2}E^\top E = \frac{1}{2}(Y - \mathbf{X}\theta)^\top (Y - \mathbf{X}\theta), \\
 &= \frac{1}{2}\theta^\top \mathbf{X}^\top \mathbf{X}\theta - \mathbf{X}^\top Y\theta + \frac{1}{2}Y^\top Y.
 \end{aligned} \tag{9}$$

1242 Since this is an unbiased least squares estimate, it holds true if and only if the following equation is  
 1243 satisfied

$$1244 \frac{\partial O}{\partial \theta} = \mathbf{X}^\top \mathbf{X} \theta - \mathbf{X}^\top Y = \mathbf{X}^\top (\mathbf{X} \theta - Y) = -\mathbf{X}^\top E = 0. \quad (10)$$

1245  
 1246 Equation 10 proof the orthogonality principle.  $\square$

## 1247 F.2 PROOF OF LEMMA 3

1248  
 1249 *Proof.* The proof of the number of basis vectors is conducted using mathematical induction.

1250  
 1251 (1) When there is only 1 basis function, it is obviously true.

1252  
 1253 (2) Assuming that the statement holds true for  $k$  basis vectors, we consider the case when there are  
 1254  $(k + 1)$  basis vectors. Since the residual corresponding to the representation of the target using  $k$   
 1255 basis vectors is orthogonal to any linear combination of these  $k$  basis vectors (Lemma 2), and the  
 1256 additional  $(k + 1)$ th basis function is orthogonal to the  $k$  basis vectors, the projection of the linear  
 1257 combination of the  $k$  basis vectors onto the  $(k + 1)$  basis vectors and the residual vector must be  
 1258 zero (i.e., the subsequent selections do not affect the influence of the  $(k + 1)$ th basis function on the  
 1259 residual). The problem then reduces to the projection process of the  $(k + 1)$  basis vectors onto  $E$ .  
 1260 According to (1), the first greedy selection is also globally optimal. After the first greedy selection,  
 1261 there remain  $k$  basis vectors, and by the assumption, the conclusion holds.  $\square$

## 1262 F.3 PROOF OF THEOREM 1

1263  
 1264 *Proof.* Consider the regression model  $Y = \mathbf{X} \theta + E$ .

1265  
 1266 According to Corollary 2, the matrix  $X$  is decomposed to  $\mathbf{X} = \mathbf{V} \mathbf{R}$ , where  $\mathbf{V}_{m \times n}$  is a matrix with  
 1267 pairwise orthogonal columns  $\mathbf{V}_i$ , and  $\mathbf{R}_{n \times n}$  is an upper triangle matrix with diagonal entries equal  
 1268 to 1. Besides,  $\mathbf{V}^\top \mathbf{V} = \mathbf{\Lambda} = \text{diag}(d_1, d_2, \dots, d_n)$  with  $d_i = \mathbf{V}_i^\top \mathbf{V}_i$ .

1269  
 1270 Then, the regression model is remarked as

$$1271 Y = \mathbf{V} B + E, \quad (11)$$

1272  
 1273 in which  $\mathbf{R} \theta = B$ . The solution to the regression model is calculated by  $B = (\mathbf{V}^\top \mathbf{V})^{-1} \mathbf{V}^\top Y =$   
 1274  $\mathbf{\Lambda}^{-1} \mathbf{V}^\top Y$ . The values are denoted as  $B_i = (\mathbf{V}_i^\top Y) / (\mathbf{V}_i^\top \mathbf{V}_i)$ ,  $i \in [1, n]$ .

1275  
 1276 According to Lemma 2,  $\mathbf{V}_i$  is orthogonal to  $\mathbf{V}_j$  and  $E$ . By multiplying  $Y$  itself and dividing by  $N$ ,  
 1277 the output variance is given by

$$1278 \frac{1}{N} Y^\top Y = \frac{1}{N} \sum_{i=1}^n B_i^2 \mathbf{V}_i^\top \mathbf{V}_i + \frac{1}{N} E^\top E, \quad (12)$$

1281  
 1282 where  $\frac{1}{N} B_i^2 \mathbf{V}_i^\top \mathbf{V}_i$  is the variance caused by regressor  $\mathbf{V}_i$ . Larger value indicates the regressor is  
 1283 more significant. Thus, the regressor  $\mathbf{V}_i$  reduces the error  $E$  at the following error reduction ratio  $R_i$

$$1284 R_i = \frac{B_i^2 \mathbf{V}_i^\top \mathbf{V}_i}{Y^\top Y}. \quad (13)$$

1285  
 1286 To select  $n_s$  significant regressors, Gram-Schmidt orthogonalization method is adopted here.

1287  
 1288 For  $i \in [1, n]$ , we have

$$1289 \mathbf{V}_1^{(i)} = \mathbf{X}_i, \quad (14)$$

$$1290 B_1^{(i)} = \frac{\mathbf{V}_1^{(i)\top} Y}{\mathbf{V}_1^{(i)\top} \mathbf{V}_1^{(i)},$$

$$1291 R_1^{(i)} = \frac{\left(B_1^{(i)}\right)^2 \mathbf{V}_1^{(i)\top} \mathbf{V}_1^{(i)}}{Y^\top Y},$$

1296 Next, calculate the largest  $R_1$  and represent as  $R_1^{(i_1)} = \max_i R_1^i$ . The corresponding regressor is  
 1297  $\mathbf{V}_1 = \mathbf{V}_1^{(i_1)} = \mathbf{X}_{(i_1)}$ .  
 1298

1299 For  $i \in [1, n]$  at  $k$ th step and  $i \neq i_1, \dots, i_{k-1}$ , we have  
 1300

$$\begin{aligned}
 1301 \quad C_{jk}^{(i)} &= \frac{\mathbf{V}_j^\top \mathbf{X}_i}{\mathbf{V}_j^\top \mathbf{V}_j} \quad \text{with } j = 1, \dots, k-1 \\
 1302 \\
 1303 \quad \mathbf{V}_k^{(i)} &= \mathbf{X}_i - \sum_{j=1}^{k-1} c_{jk}^{(i)} \mathbf{V}_j, \\
 1304 \\
 1305 \\
 1306 \quad B_k^{(i)} &= \frac{\mathbf{V}_k^{(i)\top} Y}{\mathbf{V}_k^{(i)\top} \mathbf{V}_k^{(i)}}, \\
 1307 \\
 1308 \\
 1309 \quad R_k^{(i)} &= \frac{\left(B_k^{(i)}\right)^2 \mathbf{V}_k^{(i)\top} \mathbf{V}_k^{(i)}}{Y^\top Y} \\
 1310 \\
 1311
 \end{aligned} \tag{15}$$

1312 Similarly, the maximum error reduction ratio is  $R_k^{(i_k)} = \max_i R_k^{(i)}$ , and the corresponding regressor  
 1313 is  $\mathbf{V}_k = \mathbf{V}_k^{(i_k)} = \mathbf{X}_{i_k} - \sum_{j=1}^{k-1} C_{jk}^{(i_j)} \mathbf{V}_j$ .  
 1314

1315 If the value of unexplained output variance drops below the limit  $\varepsilon$ , that is,  $1 - \sum_{j=1}^{n_s} R_j^{(i_j)} < \varepsilon$ , the  
 1316 algorithm will be predetermined. Thus, the orthogonal basis vectors  $\mathbf{V}_i$  are obtained.  $\square$   
 1317

1318  
 1319  
 1320  
 1321  
 1322  
 1323  
 1324  
 1325  
 1326  
 1327  
 1328  
 1329  
 1330  
 1331  
 1332  
 1333  
 1334  
 1335  
 1336  
 1337  
 1338  
 1339  
 1340  
 1341  
 1342  
 1343  
 1344  
 1345  
 1346  
 1347  
 1348  
 1349