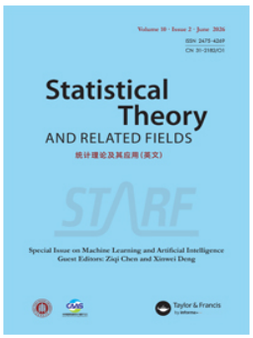

Statistical Theory and Related Fields

ISSN: 2475-4269 (Print) 2475-4277 (Online) Journal homepage: www.tandfonline.com/journals/tstf20

Deep quantile forests for high-dimensional data

Chuanquan Li, Yourong Zhang, Qi Kuang & Yini Liu

To cite this article: Chuanquan Li, Yourong Zhang, Qi Kuang & Yini Liu (2026) Deep quantile forests for high-dimensional data, Statistical Theory and Related Fields, 10:2, 232-250, DOI: 10.1080/24754269.2026.2616873

To link to this article: https://doi.org/10.1080/24754269.2026.2616873

© 2026 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.

Published online: 30 Jan 2026.

Submit your article to this journal

Article views: 342

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=tstf20

STATISTICAL THEORY AND RELATED FIELDS
2026, VOL. 10, NO. 2, 232–250
https://doi.org/10.1080/24754269.2026.2616873

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

Deep quantile forests for high-dimensional data

Chuanquan Li[a,b], Yourong Zhang[a,b], Qi Kuang[a] and Yini Liu[a,b]

[a]School of Statistics and Data Science, Jiangxi University of Finance and Economics, Nanchang, Jiangxi, People's Republic of China; [b]Key Laboratory of Data Science in Finance and Economics, Jiangxi University of Finance and Economics, Nanchang, Jiangxi, People's Republic of China

ABSTRACT

Quantile regression is essential for analyzing the relationship between conditional quantiles of independent and dependent variables, widely applied in economics, education, social science, and beyond. However, traditional methods often struggle with nonlinearity, high dimensionality, and other complexities in data. To address these challenges, this paper innovatively integrates the cascade architecture into quantile regression forests, proposing the deep quantile forest method. Unlike existing deep-quantile regression estimators, our approach requires fewer hyperparameters and less training data while offering better performance and enhanced interpretability. Extensive numerical simulations and real-world data experiments demonstrate that our proposed method outperforms competing methods, showcasing its effectiveness and robustness in handling the complex structures of high-dimensional data.

ARTICLE HISTORY
Received 20 May 2025
Revised 3 November 2025
Accepted 11 January 2026

KEYWORDS
Quantile regression; deep forest; cascade architecture; nonlinearity; high dimensional data

1. Introduction

Since quantile regression was proposed by Bassett and Koenker (1978), it has been a powerful statistical tool, focussing on estimating the conditional quantiles of a response variable given a set of predictor variables. It provides a more comprehensive understanding of the relationship between variables and responses compared to traditional mean-based regression. For example, in financial risk assessment, quantile regression can help estimate the value-at-risk (VaR) or expected shortfall, which is crucial for risk management (H. Wang & Cao, 2025); in education statistics, quantile models can help analyze different academic performance of students from different regions, schools, or socioeconomic backgrounds. Hence, the quantile methods have been widely used in economics, finance, education, and other fields (Koenker, 2017; Zhang et al., 2019).

The quantile methods have been greatly developed in the past forty years, and a series of variant methods have been proposed to deal with various and complex datasets. For the nonlinear circumstances, Powell (1986) proposed the censored regression model estimator, and Manski (1975) gave the maximum score estimator of the binary response model. Other parametric transformation methods, e.g., Box-Cox transformation, are also applied. Meanwhile, a series of nonparametric quantile methods emerged as a result of the growing need for

CONTACT Chuanquan Li lichuanquan@jxufe.edu.cn School of Statistics and Data Science, Jiangxi University of Finance and Economics, Nanchang, Jiangxi 330013, People's Republic of China; Key Laboratory of Data Science in Finance and Economics, Jiangxi University of Finance and Economics, Nanchang, Jiangxi 330013, People's Republic of China

© 2026 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

more flexible and robust statistical techniques. For example, Chaudhuri (1991) introduced the local polynomial quantile regression estimators; Koenker and Machado (1999) and He et al. (2002) proposed the quantile smoothing splines; S. Lee (2003) and Y. K. Lee et al. (2010) extended the local polynomial approach to partially linear and additive models, respectively.

Another direction is to develop a series of tree-based methods for estimating the conditional quantiles. A typical example is the random quantile forests, proposed by Meinshausen (2006). This method is similar to random forests (Breiman, 2001), which grows an ensemble of decision trees, but uses the quantile loss to select the split points for each tree. In addition, the final predicted result is given by selecting the given quantile points from the results of different trees. These strategies, including the random sample and variable subset of individual trees, as well as the aggregated prediction of multiple trees, often lead to better generalization performance than many methods. Overall, random quantile forests have been proven to show good performance, easy interpretation, and robustness to overfitting, especially in high-dimensional data scenarios.

Meanwhile, quantile regression with neural networks is combined to estimate the conditional quantiles of a response variable for many years; see Atomsa and Zhou (2023), Chen and White (1999), and White (1992). Recently, several deep quantile regression methods have been proposed, which can capture the nonlinear relationships and complex patterns, making them suitable for a wide range of applications, including investment, finance, economics, medicine, and engineering. For example, Jia and Jeong (2022) proposed a new deep quantile regression method, which is suitable for survival data with right censoring; Feng et al. (2024) investigated how to perform nonparametric quantile regression using deep neural networks in the situation of covariate shift. From the view of the theorem, Shen et al. (2021) explored that deep quantile regression can alleviate the curse of dimensionality through ensemble methods and also provided the theoretical support; Shen et al. (2024) proved the nonparametric estimation problem of non-crossing quantile functions, and provided nonasymptotic error bounds with polynomial prefactors.

However, traditional deep learning methods, such as deep neural network (DNN), convolutional neural network (CNN), and recurrent neural network (RNN), have several limitations. They often have many hyperparameters, e.g., the depth of the network, the width of the network, the activation function, the optimizer, regularization parameters, and so on, making the training process very tricky and highly dependent on careful parameter tuning. In addition, deep learning models typically require a huge amount of training data, substantial computational resources, and training time, especially when dealing with large-scale datasets (LeCun et al., 2015; Zhou & Feng, 2018). In contrast, random forests, while being more robust to data size and hyperparameter settings, cannot learn complex hierarchical feature representations in the same way as deep learning models.

In this paper, we propose the deep quantile forest that aims to combine the advantages of deep learning and random forests in the context of quantile regression. It has the potential to overcome some of the limitations of traditional methods, which can potentially handle complex data distributions and high-dimensional data more effectively in quantile regression tasks, providing more accurate and reliable conditional quantile estimation.

2. Preliminaries

In this section, we give some preliminaries for several methods. The notations are first summarized, which will appear later in this article. Without confusion, we denote $X =$

$(x_1, \ldots, x_p)^\top \in \mathbb{R}^p$ as the explanatory variables, $Y \in \mathbb{R}$ as the response variable, and $\beta_\tau \in \mathbb{R}^p$ as the coefficients in the τth quantile. The sample size is set to n.

2.1. Quantile regression

Here, we first introduce the quantile regression (Koenker, 2017). The general form of the quantile regression model for the τth quantile is given by

$$Q_\tau(Y \mid X) = X\beta_\tau,$$

where $Q_\tau(Y \mid X)$ is the conditional quantile τth of the dependent variable Y given the independent variables X, and β_τ is the regression coefficient vector for the quantile point τ. The coefficients β_τ are estimated by minimizing the following loss function:

$$\min_{\beta_\tau} \sum_{i=1}^{n} \rho_\tau(Y_i - X_i\beta_\tau),$$

where Y_i and X_i are the sample response and predictors, respectively. And $\rho_\tau(u)$ is the check function, which is defined as

$$\rho_\tau(u) = u \cdot (\tau - I(u < 0)),$$

where $I(u < 0)$ is the indicator function equal to 1 if $u < 0$ and 0 otherwise. The check function $\rho_\tau(u)$ is a piecewise linear function that assigns different weights to positive and negative residuals depending on the quantile point τ.

2.2. Quantile forest regression

Quantile forest regression(QFR), proposed by Meinshausen (2006), extends the traditional random forest framework to estimate the quantile response value based on the explaining variables. Here, two main parts are different from random forests. First, unlike standard regression (which minimizes the mean squared error), the splits are made to minimize a quantile loss function for a specific target quantile (e.g., the median). Secondly, the final prediction is not a single mean value. The collected and weighted distribution of all these training values is used to compute the empirical conditional quantile (e.g., the 90th percentile) for the new test observations, which is defined as

$$\widehat{F}(y \mid X = X_i) = \sum_{i=1}^{n} w_i(x) I(Y \le y),$$

This aggregation process ensures the robustness and reduces the variance in the quantile estimation. QFR is particularly advantageous for tasks requiring uncertainty quantification, such as prediction intervals, anomaly detection, and robust regression, as it provides a non-parametric and flexible approach to modelling the conditional distribution of Y given X. Additionally, QFR inherits the desirable properties of random forests, including scalability, handling of high-dimensional data, and robustness to outliers.

In addition, we introduce the complete random forests, proposed by Breiman (2001), which can be extended to the complete quantile forests. This method chooses the features and splits points entirely at random for each feature. Although this method has lower accuracy than the random forest, it has a fast computation time.

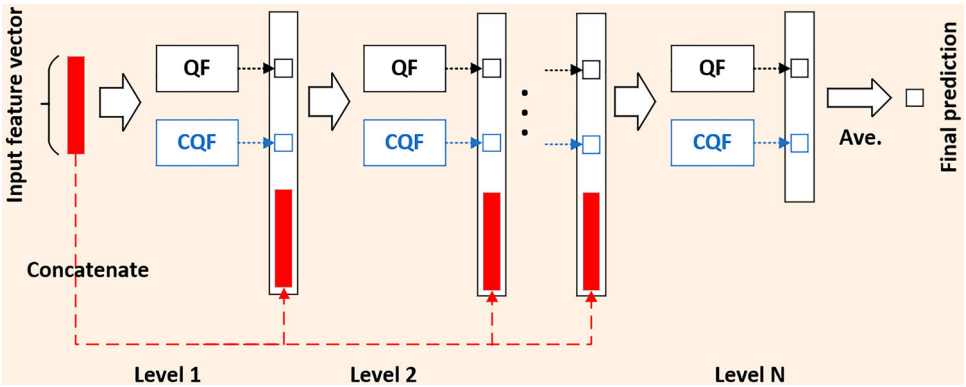

Figure 1. The structure of the cascade forest. Each level has one quantile regression forest (abbreviated as QF, with black colour) and one completely quantile regression forest (abbreviated as CQF, with blue colour). Each forest will output a quantile value, which is then concatenated for re-representation of the input.

2.3. Deep forest

Deep forest, proposed by Zhou and Feng (2018), is an ensemble learning method that integrates the advantages of random forests and the concepts of deep learning. This method is suitable for handling classification and regression tasks involving image data, high-dimensional data, and other complex patterns. A core advantage of deep forests is no need for large-scale backpropagation training, and has lower computational resource requirements and fewer hyperparameters, making the model easier to tune. Hence, it can enable the performance of the small-scale datasets. This stands in sharp contrast to deep neural networks, which typically require large volumes of training data.

The core idea of deep forests is to use the cascade learning for transforming the input data into a more discriminative representation in each layer. The deep forests consist of multiple layers of random forests, where each layer is composed of multiple decision trees. The framework of deep forests is similar to Figure 1, where the element part, quantile regression forests, is replaced by random forests. For the regression problem, the final prediction result of the model is obtained by aggregating the results of multiple forests in the last layer.

Assuming that the K forests have equal weights, the final prediction result is

$$\mathbf{y}_{\text{final}}(\mathbf{x}) = \frac{1}{K}\sum_{k=1}^{K}\mathbf{y}_k(\mathbf{x}),$$

where $\mathbf{y}_k(\mathbf{x})$ is the predicted result for the kth forest with the given predictor $\mathbf{x}$.

In addition, deep forests can also incorporate the multi-grained scanning mechanism with the cascade structure to handle the raw data with spatial or sequential relationships. This mechanism slides a window of size $s \times s$ with step length t across the data to generate sub-samples. Each subsample is then mapped to a feature vector via a random forest, forming a multi-grained feature matrix. This mechanism improves the representational learning capacity of the model, especially for visual data and textual data.

2.4. Evaluations

First, we introduce an evaluated metric for the quantile model, which is effective in distinguishing the impacts of positive and negative prediction errors by introducing a weight

parameter τ. Adjusting the quantile points τ, we can assign different weights to positive errors (where the predictions are lower than the true values) and negative errors (where the predictions are higher than the true values). Specifically, the quantile loss L_τ at the quantile point τ is defined as

$$L_\tau = \frac{1}{n}\sum_{i=1}^{n}\left[\tau \cdot |y_i - \hat{y}_i| \cdot I(y_i - \hat{y}_i > 0) + (1-\tau) \cdot |y_i - \hat{y}_i| \cdot I(y_i - \hat{y}_i \le 0)\right], \quad (1)$$

where $\hat{y}_i$ denotes the predicted value of the ith sample.

Next, quantile scoring R^2_τ at the τ quantile point is proposed based on the above quantile loss, which can provide a more robust evaluation method than traditional scoring R^2:

$$R^2_\tau = 1 - \frac{\sum_{i=1}^{n}\left[\tau \left|y_i - \hat{y}_i\right| I(y_i - \hat{y}_i > 0) + (1-\tau)\left|y_i - \hat{y}_i\right| I(y_i - \hat{y}_i \le 0)\right]}{\sum_{i=1}^{n}\left[\tau \left|y_i - \bar{y}\right| I(y_i - \bar{y} > 0) + (1-\tau)\left|y_i - \bar{y}\right| I(y_i - \bar{y} \le 0)\right]}, \quad (2)$$

where $\bar{y}$ is the mean of all actual values.

3. Method

This section will introduce the algorithm of deep quantile forests in detail, as illustrated in Figure 1. First, we input the feature variables into the different types of quantile regression forests (e.g., quantile regression forests and completely quantile regression forests). It is noted that different types of forests are efficient in encouraging the diversity at each level and avoiding the overfitting of the model. Then the quantile loss is recorded for different quantile regression forests.

Next, the first level consists of two quantile forests, where each forest is trained independently, and outputs the different quantile predictor values. The cascade structure between each level integrates the feature information processed by the preceding level with input feature vectors. Here, there are multiple methods for directly dealing with the two quantile predictor values, e.g., the mean or the predictor values. For subsequent layers, starting from the second layer, the input variables to each layer are a combination of the quantile vectors output by all forests in the previous layer and the original input samples.

The stopping condition during the training procedure is determined by the quantile loss. Specifically, if there is a significant improvement of the quantile loss, it indicates that the addition of the current layer contributes positively to the model's performance, and thus, the cascading process continues. Conversely, if no substantial improvement is observed, it suggests that further increasing the model's complexity may not yield additional benefits. In such cases, the training process can be terminated, and the optimal layer configuration is achieved. Additionally, a maximum number of layers is also predefined to prevent the excessive complexity of the model and avoid overfitting. This approach not only helps to construct an efficient model without unnecessary complexity, but also ensures effective control over computational resource usage.

Finally, each forest in the optimal layer generates its quantile prediction, and these predictions are subsequently aggregated by averaging. During the testing phase, the stored optimal model predicts new features. The features are then updated based on these new predictions. Ultimately, the quantile prediction value is returned based on the result of the Nth layer.

To further explore the feature influences of our method, the variable perturbation algorithm to assess the feature importance is employed. After training the quantile forest,

Algorithm 1 The algorithm flow chart of deep quantile forests

Require: Data: Feature matrix X_i, $i = 1, 2, \ldots, n$;
Hyperparameters: maximum number of layers N, quantile parameter τ, early stopping tolerance ϵ;
Initial value: $L_\tau^{(*)} = 0$, $X^{(0)}$=X.

Ensure: Training Phase

1: **for** each layer j from 0 to $N - 1$ **do**
2: Train each quantile forest, and compute the quantile loss of the out-of-bag data:

$$L_\tau^{(j)} \leftarrow \frac{1}{2}\sum_{i=1}^{2} L_{\tau,i}^{(j)}\left(Y^{(j-1)}, X^{(j-1)}\right),$$

where $X^{(j)}$ is the input feature of the jth layer, and $L_{\tau,i}$ is the τth quantile loss function of the ith quantile forest in the jth layer.
3: **if** $\mathcal{L}_\tau^{(j)} > \mathcal{L}_\tau^* - \epsilon$ **then**
4: Update the loss function: $\mathcal{L}_\tau^* = \mathcal{L}_\tau^{(j)}$
5: Output the response: $\hat{Y}^{(j)}$
6: **else**
7: Terminate training and save optimal layer j^*
8: **break**
9: **end if**
10: **end for**

Ensure: Result output

11: Training the jth layer forest based on the output of the $j - 1$ layer;
12: Average the predicted quantile result of the j^* layer.

for each feature, its values in the test set are shuffled. The change in the quantile loss before and after perturbation is measured. A greater increase in loss after perturbation implies a higher importance of the feature. Specifically, the baseline prediction quantile loss on the original test set is first calculated. Then, for each feature, after shuffling its values in the test set, the new prediction loss is computed, and the difference from the baseline loss reflects the feature's importance. This perturbation process is repeated multiple times (e.g., 100 iterations) for stability, and the mean and standard deviation of importance scores across repeats are reported.

Above all, the flow chart of our algorithm is shown in Algorithm 1.

4. Simulation

In this section, we compare the performance of several quantile regression methods based on the different simulated datasets. To be specific, the compared methods of quantile regression are given below.

- Traditional linear quantile regression (LQR) was proposed by Bassett and Koenker (1978). And in the Python library statsmodels, the quantile regression model is implemented through the QuantReg class.

- Deep quantile regression (DQR), proposed by Shen et al. (2021), can be implemented in Python via Pytorch and use Adam as the optimization algorithm with the default learning rate of 0.01 and default $\beta = (0.9, 0.99)$ (coefficients used to calculate running averages of gradients and their squares).
- Quantile regression forests (QRF), proposed by Meinshausen (2006), is a tree-based nonparametric ensemble method for estimating conditional quantiles, applicable to high-dimensional data and uncertainty estimation. The quantile-forest package provides a Python implementation of quantile-regression forests compatible with scikit-learn.
- Deep quantile forest (DQF), set the number of trees as 500 in each quantile random forest (Breiman, 2001), to balance predictive performance and computational cost. The maximum number of cascade layers is set to 10 (cascade_layer = 10). And the parameter sensitivity of cascade layers will be introduced in Section 4.4.

4.1. Data generation mechanism

Here, we first give the complex and diverse relationships between the input variables X and response Y, and then test the prediction performance with different quantiles. We consider the following three types of basis functions: the additive model, the multivariate single index model, and the deep interaction model. The simulation setting of the first two models is similar to Shen et al. (2021), and the last one is from the paper of Z. Wang (2023). For each of them, we consider two different variants: the sin and exponential function.

(1) *Setting 1: Multivariate Single Index Model*

$$Y = f(X\beta) + \epsilon.$$

- Variant v1: sin function

$$Y = (20 - \sin(X\beta))^2 + \epsilon$$

where $\beta = (1.5, -2.0, 4.0, 1.0, 0.5, 0, 0.2, 0.8, 0, \ldots, 0)^\top$.

- Variant v2: exponential function

$$Y = \exp(X\beta) + \epsilon$$

where $\beta = (2.2831, -1.4818, 5.1966, 0, 0, 0.0515, 0, \ldots, 0)^\top$.

(2) *Setting 2: Additive Model*

$$Y = f_1(x_1) + f_2(x_2) + \cdots + f_p(x_p) + \epsilon$$

where f_j are non-linear functions acting on different predictor variables.

- Variant v1 : sin function

$$Y = 3\log(3 + x_1^2) + 5(x_2 - 0.5)^3 + 8\cos(2\pi x_3) - 3|x_4 - 0.8| + \epsilon.$$

- Variant v2 : exponential function

$$Y = \exp\left(4(x_1 - 0.5)\right) + 9(x_2 - 0.5)^2 + 10\sin(2\pi x_3) - 7|x_4 - 0.5| + \epsilon.$$

(3) *Setting 3: Deep Interaction Model*

$$Y = f(x_1, x_2) + g(x_3, x_4) + \epsilon,$$

where f and g are complex interaction functions involving multiple predictor variables.

- Variant v1 : sin function

$$Y = 4\cos(3\pi x_1 x_3) + 2(x_2^2 + x_4^3) + \epsilon.$$

- Variant v2 : exponential function

$$Y = 5\sin(2\pi x_1 x_2) + 3\exp(x_3 + x_4) + \epsilon.$$

Here, the number of variables is set to 200, and the sample size is set to 1000. Meanwhile, we also consider the different error terms. The error term ϵ follows a normal distribution with zero mean, but it has three variance structures:

- *Homoscedastic Variance*: The variance of the error term is constant, $\mathrm{Var}(\epsilon) = \sigma^2$.
- *Linear Heteroscedastic Variance*: The variance of the error term changes linearly with a predictor variable, $\mathrm{Var}(\epsilon) = \sigma^2(1 + x_1)$.
- *Non-linear Heteroscedastic Variance*: The variance of the error term changes non-linearly with a predictor variable, $\mathrm{Var}(\epsilon) = \sigma^2 \exp(x_1)$.

4.2. Simulation result

We generate the eighteen simulated situations based on Section 4.1. Then, the three quantile points at [0.1, 0.5, 0.9] are considered to test the performance of different methods. To ensure the stability and reliability of the results, each experimental scenario is repeated 50 times. The quantile loss L_τ in Equation (1) and the quantile scoring R_τ^2 in Equation (2) are both recorded.

In the single-index model, apart from the LQR method, the other three methods perform well, but our method achieves the best results. According to the paper of Shen et al. (2021), DQR can be applied within the framework of single-index models. However, in this study, the data exhibit a sparse structure, which slightly disadvantages DQR, while the quantile forests demonstrate a more significant advantage. Furthermore, in additive models, the adjustment accuracy (R_τ^2) of the DQR and LQR methods only reaches around 0.2 and 0.3 under different basis functions and even becomes negative under exponential basis functions. Similarly, under the L_τ metric, DQR and LQR methods also perform poorly. This indicates that the DQR method is not suitable for high-dimensional data.

In this section, we focus only on showing the comparison of model results at the quantile of 0.5. The results in the 0.5 quantile are shown in Tables 1–3. We can observe that our method consistently outperforms other approaches across all simulation settings from Tables 1–3. The results for quantiles 0.1 and 0.9 are similar and are put in Tables A1–A6 of Appendix.

4.3. Computation time

In this part, we compare the computation time of the four methods by the Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50 GHz. For simplicity, we choose the sin function of Setting 1 as our simulation object, and the error distribution is set as the homoscedastic variance. Different quantile results (e.g., 0.1, 0.5, 0.9) are recorded in Figure 2.

Table 1. The numerical performance of the simulated data of the multivariate single index model at the 0.5 quantile.

		sin			exp		
		Homo	Linear	Nonlinear	Homo	Linear	Nonlinear
L_τ	DQF	7.5076	7.4101	7.4143	4.9114	4.9007	4.8767
		(2.2546)	(1.7297)	(1.7818)	(0.1197)	(0.0969)	(0.1037)
	DQR	11.4543	11.4514	11.5974	9.2346	9.2263	9.1728
		(5.8110)	(6.0998)	(6.3337)	(0.4047)	(0.3770)	(0.3479)
	QRF	10.5828	10.5461	10.5683	5.6207	5.6259	5.6176
		(4.9301)	(4.7908)	(4.6500)	(0.1304)	(0.1256)	(0.1222)
	LQR	29.1458	29.1445	29.1437	7.0883	7.0775	7.0808
		(12.8830)	(12.8996)	(12.9182)	(0.1928)	(0.1921)	(0.1916)
R^2_τ	DQF	0.7907	0.7931	0.7928	0.6121	0.6129	0.6148
		(0.0006)	(0.0004)	(0.0005)	(0.0011)	(0.0010)	(0.0010)
	DQR	0.6818	0.6823	0.6775	0.2708	0.2712	0.2756
		(0.0013)	(0.0012)	(0.0016)	(0.0035)	(0.0035)	(0.0031)
	QRF	0.7059	0.7068	0.7061	0.5561	0.5557	0.5563
		(0.0011)	(0.0011)	(0.0011)	(0.0012)	(0.0012)	(0.0012)
	LQR	0.1823	0.1823	0.1824	0.4402	0.4410	0.4407
		(0.0020)	(0.0020)	(0.0020)	(0.0018)	(0.0018)	(0.0018)

Note: Sin function represents the variant v1; exp function represents the variant v2. The L_τ and R^2_τ are both recorded.

Table 2. The numerical performance of the simulated data of the additive model at the 0.5 quantile.

		sin			exp		
		Homo	Linear	Nonlinear	Homo	Linear	Nonlinear
L_τ	DQF	0.6011	0.5644	0.5666	0.3093	0.2444	0.2457
		(0.0015)	(0.0014)	(0.0016)	(0.0003)	(0.0002)	(0.0002)
	DQR	2.1382	2.1467	2.1230	3.1125	3.0609	3.0948
		(0.1067)	(0.0608)	(0.0731)	(0.0382)	(0.0298)	(0.0332)
	QRF	0.6449	0.6174	0.6171	0.3333	0.2811	0.2819
		(0.0017)	(0.0017)	(0.0016)	(0.0003)	(0.0002)	(0.0002)
	LQR	2.4736	2.4697	2.4688	3.1247	3.1179	3.1170
		(0.0106)	(0.0114)	(0.0114)	(0.0233)	(0.0236)	(0.0231)
R^2_τ	DQF	0.8124	0.8237	0.8230	0.8811	0.9059	0.9054
		(0.0002)	(0.0002)	(0.0002)	(0.0001)	(0.0000)	(0.0000)
	DQR	0.3338	0.3290	0.3365	−0.1957	−0.1779	−0.1909
		(0.0090)	(0.0071)	(0.0081)	(0.0056)	(0.0051)	(0.0056)
	QRF	0.7987	0.8071	0.8072	0.8719	0.8918	0.8915
		(0.0002)	(0.0002)	(0.0002)	(0.0001)	(0.0000)	(0.0000)
	LQR	0.2281	0.2287	0.2289	−0.2005	−0.1997	−0.1993
		(0.0017)	(0.0017)	(0.0017)	(0.0036)	(0.0038)	(0.0038)

Note: Sin function represents the variant v1; exp function represents the variant v2. The L_τ and R^2_τ are both recorded.

We conclude that (1) although the computation time of QRF and LQR is less, the accuracy is lower, and relative to other deep learning methods, the computation time of our method is less than the DQR; (2) when the sample size grows more than 20,000, the computation time of our method has an obvious advantage. Above all, we conclude that our method has less computation time when compared with DQR.

4.4. The parameter sensitivity: cascade layer

In this part, we evaluate the parameter sensitivity of our model, cascade layer, through two synthetic datasets: additive (Variant v1) and deep interaction (Variant v1). For consistency,

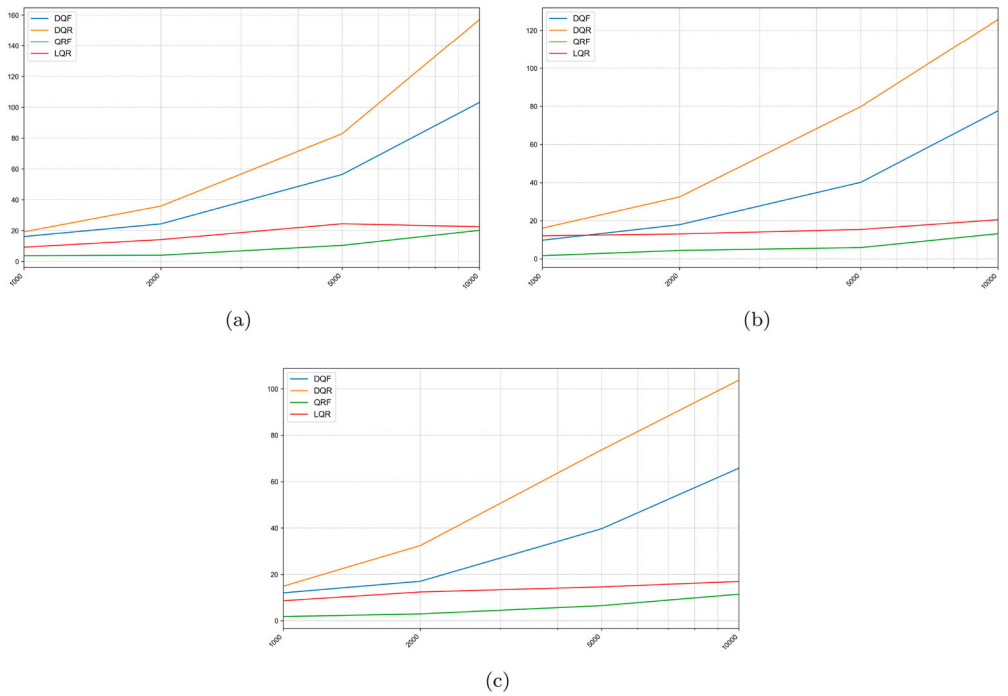

Figure 2. The computation time of four quantile methods: (a) at the 0.1th quantile, (b) at the 0.5th quantile, and (c) at the 0.9th quantile.

we fix the feature dimension at $d = 200$ and the sample size at $n = 1000$, while recording the loss of out-of-bag (OOB) training and the optimal cascade layer in three quantiles ($q = 0.1$, 05, 0.9), which is shown in Figure 3. We can find that for the additive dataset, the loss of OOB training declines dramatically and stabilizes quickly, with the optimal cascade layer converging to the second layer in all quantiles; this demonstrates the efficient capture of additive non-linear patterns by the model. However, for the deep-interaction dataset, the loss decreases more gradually, and the optimal cascade layer shifts to the third layer, reflecting the need for additional cascade iterations to model complex interaction effects. In addition, from Figure 3, it is indicated that the structural complexity of the dataset takes precedence over quantile selection to determine the optimal cascade configuration. Above all, we can conclude that our algorithm can be efficient in finding the optimal layer. And, as usual, the optimal number of cascade layers does not exceed 10.

5. Real data analysis

Here, the credit risk dataset used in this study is sourced from Kaggle, which is often used to compare the risk assessment ability of different methods. The URL is https://www.kaggle.com/datasets/tushtiverma/credit-risk-modelling. Credit risk assessment is a crucial task in the financial field, aiming to evaluate the probability of borrower default or the potential loss extent. With the increasing complexity of financial markets and the growing volume of data, traditional credit risk assessment methods face numerous challenges.

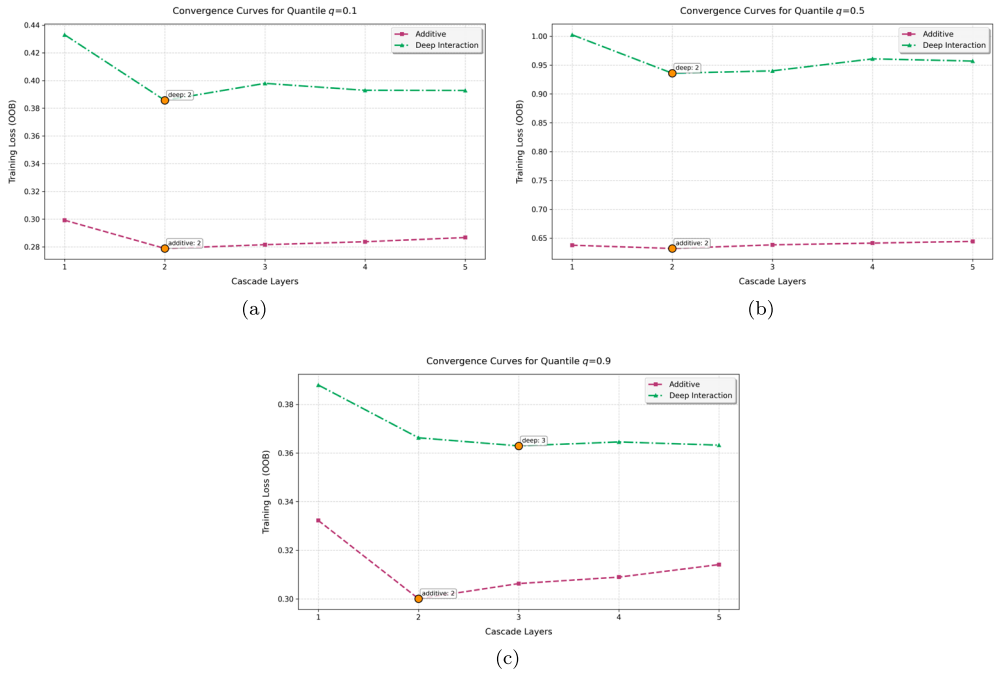

Figure 3. The parameter sensitivity of DQR, cascade layer, (a) at the 0.1th quantile, (b) at the 0.5th quantile, and (c) at the 0.9th quantile.

Table 3. The numerical performance of the simulated data of the deep interaction model at the 0.5 quantile.

		sin			exp		
		Homo	Linear	Nonlinear	Homo	Linear	Nonlinear
L_τ	DQF	0.8711	0.8472	0.8482	0.5251	0.4791	0.4828
		(0.0029)	(0.0029)	(0.0031)	(0.0016)	(0.0015)	(0.0014)
	DQR	1.3861	1.3536	1.3530	1.2696	1.2480	1.2587
		(0.0062)	(0.0054)	(0.0049)	(0.0034)	(0.0051)	(0.0052)
	QRF	0.9331	0.9124	0.9145	0.6363	0.6018	0.6042
		(0.0027)	(0.0030)	(0.0028)	(0.0035)	(0.0032)	(0.0032)
	LQR	1.3577	1.3428	1.3439	1.2720	1.2450	1.2514
		(0.0047)	(0.0043)	(0.0044)	(0.0031)	(0.0027)	(0.0029)
R^2_τ	DQF	0.5127	0.5239	0.5236	0.5917	0.6259	0.6232
		(0.0010)	(0.0011)	(0.0010)	(0.0014)	(0.0013)	(0.0012)
	DQR	0.2246	0.2388	0.2397	0.0136	0.0260	0.0182
		(0.0023)	(0.0029)	(0.0023)	(0.0032)	(0.0044)	(0.0044)
	QRF	0.4781	0.4874	0.4864	0.5054	0.5301	0.5285
		(0.0009)	(0.0010)	(0.0009)	(0.0027)	(0.0024)	(0.0024)
	LQR	0.2400	0.2450	0.2446	0.0118	0.0286	0.0239
		(0.0025)	(0.0024)	(0.0024)	(0.0027)	(0.0024)	(0.0029)

Note: Sin function represents the variant v1; exp function represents the variant v2. The L_τ and R^2_τ are both recorded.

After preprocessing steps such as dimension splicing and feature engineering, this dataset contains 10,000 valid data samples and 4 feature classes with 71 variables, including the borrower's basic information, credit history and behaviour information, loan information, and credit score (Credit_Score) and loan approval indicator (Approved_Flag). In addition, the Credit_Score variable is regarded as the response. All detailed variables are shown in Table 4.

Table 4. The explanatory variables of the credit risk dataset.

Feature class	Variables	Description
Borrower Basic Information	Prospectid	Unique identifier for distinguishing different borrowers
	Marital_status	Marital status, such as married, single, divorced, etc.
	Education	Highest educational level, such as high school, bachelor's, master's, etc.
	Age	Borrower's age
	Gender	Gender
	Net_monthly_income	Net monthly income
Credit History and Behaviour	Time_since_recent_payment	Time since the most recent payment
	Time_since_first_delinquency	Time since the first delinquency
	Time_since_recent_delinquency	Time since the most recent delinquency
	Num_times_delinquent	Number of delinquencies
	Max_delinquency_level	Highest delinquency level
Loan Information	Total_TL	Total number of loans
	Tot_Closed_TL	Number of closed loans
	Tot_Active_TL	Number of active loans
	Total_TL_opened_L6M	Number of loans opened in the past 6 months
Credit Score and Other Information	Credit_Score	Borrower's credit score
	Approved_Flag	Loan approval indicator (0 for not approved, 1 for approved)

Table 5. Quantile prediction results from four quantile methods at the credit risk dataset.

		Model			
	Quantile	DQF	DQR	LQR	QRF
L_τ	0.1	**1.386**	2.336	1.634	1.851
		(0.044)	(2.901)	(0.147)	(0.056)
	0.5	**3.240**	4.594	3.643	4.032
		(0.063)	(5.585)	(0.620)	(0.090)
	0.9	**1.319**	1.693	1.615	1.762
		(0.025)	(0.957)	(1.011)	(0.047)
R^2_τ	0.1	**0.811**	0.683	0.777	0.748
		(0.006)	(0.387)	(0.019)	(0.009)
	0.5	**0.572**	0.395	0.519	0.468
		(0.010)	(0.733)	(0.080)	(0.013)
	0.9	**0.831**	0.783	0.793	0.774
		(0.005)	(0.124)	(0.130)	(0.008)

The four methods are run in an identical experimental environment to ensure the objectivity and comparability of experimental results. 80% of the dataset is divided into training models, and 20% is used for the test data. The results in Table 5 show that the DQF outperforms the other three methods across all quantiles. Specifically, at the 0.1 quantile, DQF achieves an R^2_τ value of 0.8110, which is significantly higher than DQR(0.683), QRF(0.7478) and LQR (0.777). At the 0.5 quantile, the value of DQF R^2_τ is 0.572, surpassing DQR (0.375), QRF (0.468), and LQR (0.519). In the 0.9 quantile, the value of DQF R^2_τ is 0.831, which is significantly higher than DQR (0.783), QRF (0.744), and LQR (0.793). These results demonstrate that the deep quantile forests model has a significant advantage in handling complex and high-dimensional sparse data relationships.

Furthermore, this paper randomly selects 200 samples as the test dataset. And based on the predicted interval length of each sample, we order the samples and plot line graphs of the

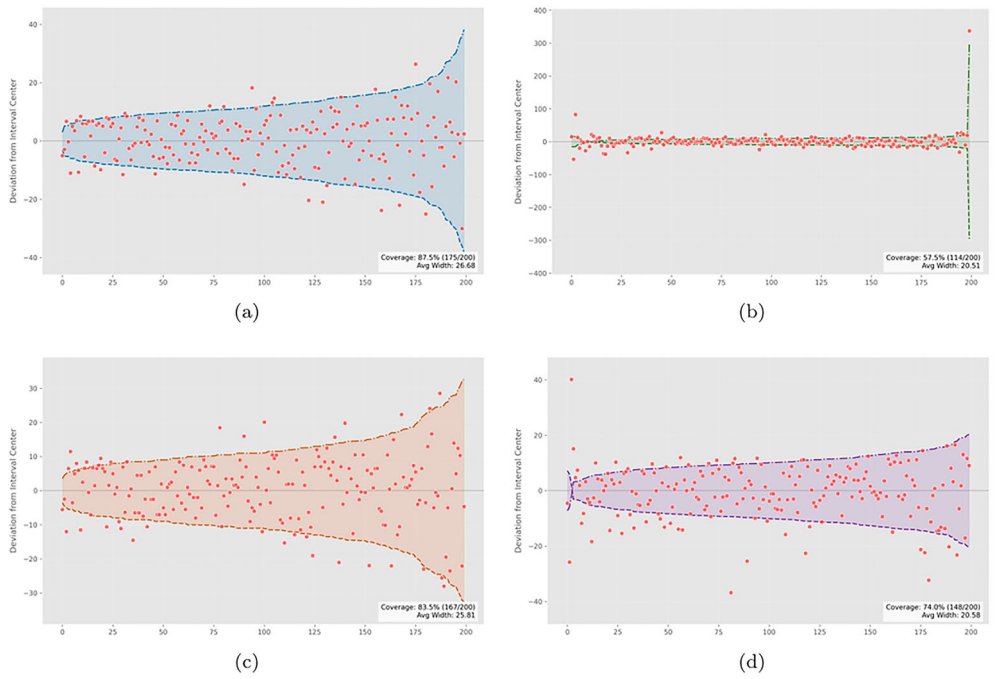

Figure 4. The interval prediction of four methods for the Credit Risk Modelling Data Set. (a) Deep quantile forests, (b) Deep quantile regression, (c) Quantile regression forests and (d) Linear quantile regression.

0.1 and 0.9 quantile curves along with the true value curve in Figure 4. This process intuitively demonstrates the performance of different models in interval prediction. As can be seen in the graph, the prediction interval of the deep quantile forests generally covers the true values well in most cases with a coverage rate of 87.5%, indicating that its predictions are highly reliable. In contrast, the prediction intervals of QRF and LQR fail to fully cover the true values in some cases, with 83.5% and 74% percent, respectively, and their predictive performance is inferior to our method. From Figure 4(b), the coverage rate of the deep quantile neural network is only 57.5%, and the interval prediction of several points becomes so large because this method is easy to overfit. This further validates the superior performance of the deep quantile forests model in credit risk assessment, particularly in terms of its ability to ensure that prediction intervals cover the true values more effectively.

To complement the analysis, the variable perturbation algorithm of the DQF method, introduced in Section 3, was applied to the credit risk data, and the result of variable importance is shown in Figure 5. The most influential features (e.g., Age_Oldest_TL, enq_L3m) exhibited higher importance scores, with mean loss increases of 2.549 and 2.196 (accompanied by standard deviations of 0.038 and 0.023, respectively). Most categorical characteristics (e.g., GENDER, Marital_status) were relatively lower important, some even showing negative scores (indicating marginal or negligible impact on prediction loss). The complete importance results of the features, presented as the mean $\pm$ standard deviation of the increase in loss, provide detailed information on the contributions of the features.

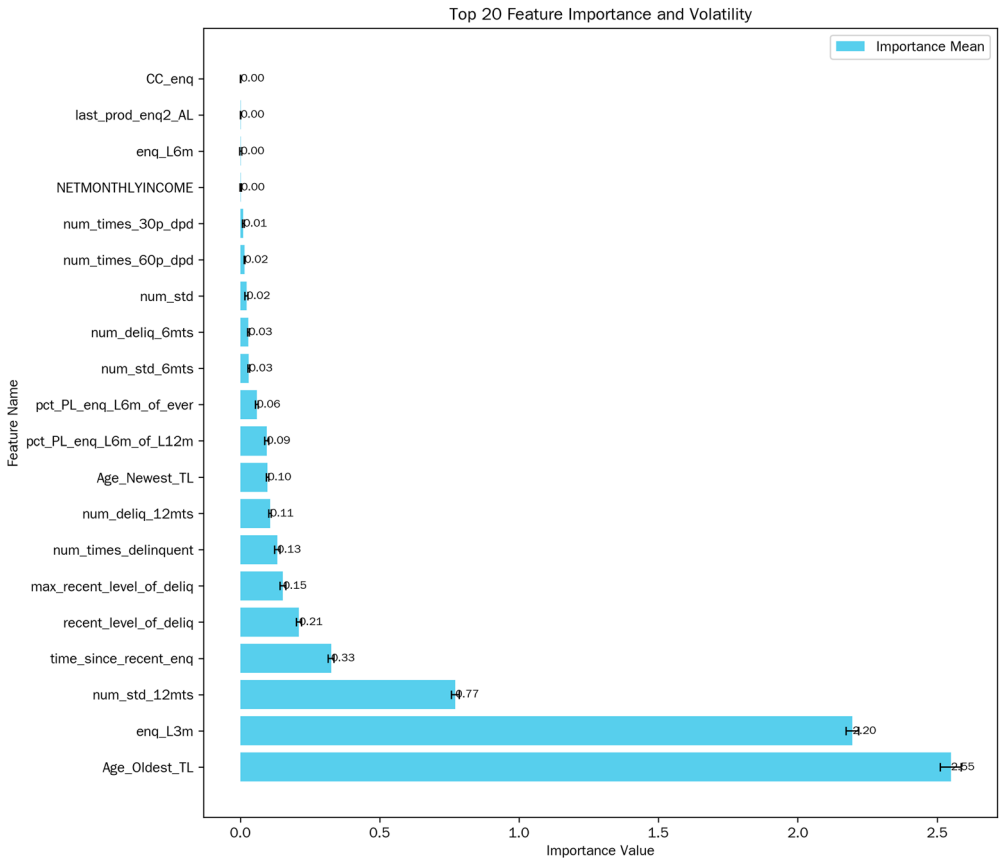

Figure 5. The feature importance of credit risk data based on the deep quantile forests.

6. Conclusion

This paper successfully constructs a deep quantile forest by innovatively integrating deep forests with quantile regression forests. This architecture of deep learning makes the proposed method overcome the limitations of traditional quantile methods in handling high-dimensional data and enhancing its ability to fit complex data distributions. Through numerical simulations and case studies, the effectiveness of the deep quantile forests is fully validated. The comparative results of numerical simulations show that the proposed method performs exceptionally well in quantile prediction. In particular, in handling highly complex and sparse data relationships, the cascade forest structure can extract more useful information, leading to more accurate predictions. The case study, using credit risk assessment as an example, further confirms the model's advantages in practical applications. In addition, the method can provide variable importance measures, offering robust support for related decision-making processes.

Despite the achievements of this article, there is still room for further improvement and expansion of deep quantile forests. For example, the attention mechanisms (Utkin & Konstantinov, 2022) could be applied to the predicted values in each step to enhance the effectiveness of the prediction information. Additionally, refining the early stopping iteration process could further boost model performance while improving computational efficiency.

Furthermore, future studies could incorporate multi-grained scanning mechanisms (Zhou & Feng, 2018) to expand the applicability of deep quantile forests in areas such as image and video processing.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

Chuanquan Li's research was supported by the NSF of China [grant number 12301377], and the NSF of Jiangxi Province [grant number 20232BAB211014], and National Natural Science Foundation of China Major Research Program [grant number 92358303], Major Research Plan on West-Pacific Earth System Multispheric Interactions [grant number 92358303], and Jiangxi Provincial Ganpo Talent Program [grant number 20243BCE51010]. Qi Kuang's research was supported by the Early-Career Young Scientists and Technologists Project of Jiangxi Province [grant number 20232BAB211014].

References

Atomsa, G. A., & Zhou, Y. (2023). Application of neural network to model rainfall pattern of Ethiopia. *Statistical Theory and Related Fields*, *7*(1), 69–84. https://doi.org/10.1080/24754269.2022.2136266

Bassett, G., & Koenker, R. (1978). Regression quantiles. *Econometrica*, *46*(1), 33–50. https://doi.org/10.2307/1913643

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Chaudhuri, P. (1991). Global nonparametric estimation of conditional quantile functions and their derivatives. *Journal of Multivariate Analysis*, *39*(2), 246–269. https://doi.org/10.1016/0047-259X(91)90100-G

Chen, X., & White, H. (1999). Improved rates and asymptotic normality for nonparametric neural network estimators. *IEEE Transactions on Information Theory*, *45*(2), 682–691. https://doi.org/10.1109/18.749011

Feng, X., He, X., Jiao, Y., Kang, L., & Wang, C. (2024). Deep nonparametric quantile regression under covariate shift. *Journal of Machine Learning Research*, *25*(385), 1–50.

He, X., Ng, P., & Portnoy, S. (2002). Bivariate quantile smoothing splines. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *60*(3), 537–550. https://doi.org/10.1111/1467-9868.00138

Jia, Y., & Jeong, J. (2022). Deep learning for quantile regression under right censoring: DeepQuantreg. *Computational Statistics & Data Analysis*, *165*, 107323. https://doi.org/10.1016/j.csda.2021.107323

Koenker, R. (2017). Quantile regression: 40 years on. *Annual Review of Economics*, *9*(1), 155–176. https://doi.org/10.1146/economics.2017.9.issue-1

Koenker, R., & Machado, J. A. F. (1999). Goodness of fit and related inference processes for quantile regression. *Journal of the American Statistical Association*, *94*(448), 1296–1310. https://doi.org/10.1080/01621459.1999.10473882

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lee, S. (2003). Efficient semiparametric estimation of a partially linear quantile regression model. *Econometric Theory*, *19*(1), 1–31. https://doi.org/10.1017/S0266466603191013

Lee, Y. K., Mammen, E., & Park, B. U. (2010). Backfitting and smooth backfitting for additive quantile models. *The Annals of Statistics*, *38*(5), 2857–2883. https://doi.org/10.1214/10-AOS808

Manski, C. F. (1975). Maximum score estimation of the stochastic utility model of choice. *Journal of Econometrics*, *3*(3), 205–228. https://doi.org/10.1016/0304-4076(75)90032-9

Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine Learning Research*, *7*(6), 983–999.

Powell, J. L. (1986). Censored regression quantiles. *Journal of Econometrics*, *32*(1), 143–155. https://doi.org/10.1016/0304-4076(86)90016-3

Shen, G., Jiao, Y., Lin, Y., Horowitz, J., & Huang, J. (2021). Deep quantile regression: Mitigating the curse of dimensionality through composition. arXiv preprint.
Shen, G., Jiao, Y., Lin, Y., Horowitz, J. L., & Huang, J. (2024). Nonparametric estimation of non-crossing quantile regression process with deep ReQU neural networks. *Journal of Machine Learning Research*, *25*(88), 1–75.
Utkin, L. V., & Konstantinov, A. V. (2022). Attention-based random forest and contamination model. *Neural Networks*, *154*, 346–359. https://doi.org/10.1016/j.neunet.2022.07.029
Wang, H., & Cao, R. (2025). Deep learning quantile regression for interval-valued data prediction. *Journal of Forecasting*, *44*(5), 1806–1825. https://doi.org/10.1002/for.v44.5
Wang, Z. (2023). *Research on Variable Importance Estimation Based on Deep Forest [D]*. Nanjing Medical University. https://doi.org/10.27249/d.cnki.gnjyu.2023.000487
White, H. (1992). Nonparametric estimation of conditional quantiles using neural networks. In C. Page & R. LePage (Eds.), *Computing Science and Statistics* (pp. 190–199). Springer.
Zhang, Y., Wang, L., Yu, M., & Shao, J. (2019). Quantile treatment effect estimation with dimension reduction. *Statistical Theory and Related Fields*, *4*(2), 202–213. https://doi.org/10.1080/24754269.2019.1696645
Zhou, Z., & Feng, J. (2018). Deep forest. *National Science Review*, *6*(1), 74–86. https://doi.org/10.1093/nsr/nwy108

Appendix. Additional simulation results in Section 4

Table A1. The numerical performance of the simulated data of the multivariate single index model at the 0.1 quantile.

		sin			exp		
		Homo	Linear	Nonlinear	Homo	Linear	Nonlinear
L_τ	DQF	3.3726	3.3819	3.3562	2.1887	2.1779	2.1718
		(0.3253)	(0.3612)	(0.3431)	(0.0138)	(0.0141)	(0.0109)
	DQR	6.0627	5.9937	6.0894	4.9431	4.9298	4.9281
		(2.5054)	(1.9769)	(1.8297)	(0.1946)	(0.1875)	(0.1441)
	QRF	4.3225	4.3040	4.3183	2.3562	2.3505	2.3401
		(0.4923)	(0.4840)	(0.4871)	(0.0094)	(0.0109)	(0.0103)
	LQR	7.3059	7.3005	7.2979	4.3319	4.3254	4.3252
		(0.6351)	(0.6466)	(0.6457)	(0.1712)	(0.1582)	(0.1595)
R^2_τ	DQF	0.7177	0.7171	0.7193	0.8126	0.8137	0.8141
		(0.0006)	(0.0008)	(0.0006)	(0.0003)	(0.0003)	(0.0003)
	DQR	0.4923	0.4966	0.4881	0.5767	0.5775	0.5783
		(0.0127)	(0.0109)	(0.0087)	(0.0027)	(0.0031)	(0.0022)
	QRF	0.6381	0.6399	0.6386	0.7983	0.7989	0.7998
		(0.0006)	(0.0006)	(0.0007)	(0.0003)	(0.0003)	(0.0003)
	LQR	0.3831	0.3841	0.3841	0.6282	0.6289	0.6290
		(0.0029)	(0.0028)	(0.0028)	(0.0030)	(0.0029)	(0.0029)

Note: Sin function represents the variant v1; exp function represents the variant v2. The L_τ and R^2_τ are both recorded.

Table A2. The numerical performance of the simulated data of the additive model at the 0.1 quantile.

		sin			exp		
		Homo	Linear	Nonlinear	Homo	Linear	Nonlinear
L_τ	DQF	0.2738	0.2608	0.2615	0.1445	0.1177	0.1211
		(0.0002)	(0.0002)	(0.0002)	(0.0001)	(0.0000)	(0.0000)
	DQR	1.6044	1.6230	1.6035	2.0438	1.9938	2.0088
		(0.0433)	(0.0386)	(0.0502)	(0.0517)	(0.0664)	(0.0774)
	QRF	0.2896	0.2756	0.2767	0.1512	0.1278	0.1308
		(0.0003)	(0.0003)	(0.0003)	(0.0001)	(0.0000)	(0.0001)
	LQR	1.2604	1.2593	1.2555	1.3820	1.3785	1.3779
		(0.0188)	(0.0175)	(0.0180)	(0.0148)	(0.0163)	(0.0156)
R^2_τ	DQF	0.9158	0.9195	0.9194	0.9467	0.9567	0.9554
		(0.0001)	(0.0001)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
	DQR	0.5068	0.4990	0.5055	0.2451	0.2659	0.2586
		(0.0052)	(0.0053)	(0.0061)	(0.0125)	(0.0132)	(0.0165)
	QRF	0.9110	0.9150	0.9147	0.9442	0.9529	0.9518
		(0.0001)	(0.0001)	(0.0001)	(0.0000)	(0.0000)	(0.0000)
	LQR	0.6129	0.6117	0.6130	0.4902	0.4929	0.4924
		(0.0022)	(0.0022)	(0.0023)	(0.0037)	(0.0036)	(0.0038)

Note: Sin function represents the variant v1; exp function represents the variant v2. The L_τ and R^2_τ are both recorded.

Table A3. The numerical performance of the simulated data of the deep interaction model at the 0.1 quantile.

		sin			exp		
		Homo	Linear	Nonlinear	Homo	Linear	Nonlinear
L_τ	DQF	0.3957	0.3876	0.3834	0.2534	0.2355	0.2359
		(0.0006)	(0.0004)	(0.0004)	(0.0002)	(0.0002)	(0.0002)
	DQR	0.8048	0.7925	0.7752	0.7139	0.6934	0.7132
		(0.0069)	(0.0069)	(0.0072)	(0.0101)	(0.0098)	(0.0084)
	QRF	0.4153	0.4057	0.4051	0.2777	0.2605	0.2618
		(0.0005)	(0.0005)	(0.0004)	(0.0003)	(0.0003)	(0.0002)
	LQR	0.7286	0.7147	0.7159	0.5487	0.5314	0.5326
		(0.0040)	(0.0046)	(0.0048)	(0.0016)	(0.0016)	(0.0017)
R^2_τ	DQF	0.7588	0.7634	0.7660	0.8282	0.8410	0.8405
		(0.0003)	(0.0003)	(0.0003)	(0.0003)	(0.0002)	(0.0002)
	DQR	0.5094	0.5162	0.5272	0.5159	0.5316	0.5172
		(0.0030)	(0.0032)	(0.0032)	(0.0063)	(0.0061)	(0.0054)
	QRF	0.7467	0.7524	0.7528	0.8119	0.8242	0.8230
		(0.0004)	(0.0003)	(0.0003)	(0.0003)	(0.0002)	(0.0002)
	LQR	0.5554	0.5635	0.5631	0.6280	0.6408	0.6394
		(0.0023)	(0.0024)	(0.0024)	(0.0017)	(0.0018)	(0.0017)

Note: Sin function represents the variant v1; exp function represents the variant v2. The L_τ and R^2_τ are both recorded.

Table A4. The numerical performance of the simulated data of the multivariate single index model at the 0.9 quantile.

		sin			exp		
		Homo	Linear	Nonlinear	Homo	Linear	Nonlinear
L_τ	DQF	4.6017	4.6193	4.6109	2.3238	2.3158	2.3168
		(0.4719)	(0.5723)	(0.5210)	(0.0249)	(0.0265)	(0.0264)
	DQR	19.4920	19.3134	19.2341	5.7727	5.7827	5.8084
		(27.9829)	(27.8151)	(29.0941)	(0.3826)	(0.4124)	(0.4401)
	QRF	5.9691	6.0034	5.9653	2.4814	2.4847	2.4898
		(0.5744)	(0.6737)	(0.6154)	(0.0225)	(0.0263)	(0.0238)
	LQR	35.3857	35.4023	35.4051	4.8393	4.8448	4.8470
		(18.7247)	(18.7258)	(18.6377)	(0.2801)	(0.2847)	(0.2843)
R_τ^2	DQF	0.9223	0.9221	0.9222	0.8282	0.8286	0.8285
		(0.0001)	(0.0001)	(0.0001)	(0.0004)	(0.0004)	(0.0004)
	DQR	0.6774	0.6807	0.6816	0.5731	0.5722	0.5702
		(0.0032)	(0.0029)	(0.0036)	(0.0035)	(0.0035)	(0.0036)
	QRF	0.8990	0.8984	0.8990	0.8165	0.8162	0.8157
		(0.0002)	(0.0002)	(0.0002)	(0.0004)	(0.0004)	(0.0004)
	LQR	0.4009	0.4006	0.4005	0.6419	0.6411	0.6409
		(0.0053)	(0.0053)	(0.0053)	(0.0026)	(0.0027)	(0.0027)

Note: Sin function represents the variant v1; exp function represents the variant v2. The L_τ and R_τ^2 are both recorded.

Table A5. The numerical performance of the simulated data of the additive model at the 0.9 quantile.

		sin			exp		
		Homo	Linear	Nonlinear	Homo	Linear	Nonlinear
L_τ	DQF	0.2879	0.2768	0.2763	0.1443	0.1241	0.1209
		(0.0002)	(0.0002)	(0.0002)	(0.0001)	(0.0000)	(0.0000)
	DQR	1.6446	1.6468	1.6517	2.0037	2.0141	1.9269
		(0.0342)	(0.0234)	(0.0237)	(0.0944)	(0.0999)	(0.1159)
	QRF	0.3044	0.2935	0.2948	0.1507	0.1330	0.1299
		(0.0004)	(0.0003)	(0.0003)	(0.0001)	(0.0000)	(0.0000)
	LQR	1.3055	1.2959	1.2993	1.3977	1.3880	1.3902
		(0.0122)	(0.0133)	(0.0130)	(0.0261)	(0.0272)	(0.0261)
R_τ^2	DQF	0.9081	0.9118	0.9120	0.9415	0.9495	0.9507
		(0.0001)	(0.0001)	(0.0001)	(0.0000)	(0.0000)	(0.0000)
	DQR	0.4752	0.4755	0.4736	0.1865	0.1760	0.2143
		(0.0053)	(0.0042)	(0.0043)	(0.0239)	(0.0305)	(0.0269)
	QRF	0.9029	0.9065	0.9061	0.9389	0.9459	0.9471
		(0.0001)	(0.0001)	(0.0001)	(0.0000)	(0.0000)	(0.0000)
	LQR	0.5834	0.5871	0.5859	0.4313	0.4332	0.4313
		(0.0023)	(0.0024)	(0.0024)	(0.0096)	(0.0099)	(0.0102)

Note: Sin function represents the variant v1; exp function represents the variant v2. The L_τ and R_τ^2 are both recorded.

Table A6. The numerical performance of the simulated data of the deep interaction model at the 0.9 quantile.

		sin			exp		
		Homo	Linear	Nonlinear	Homo	Linear	Nonlinear
L_τ	DQF	0.3711	0.3628	0.3614	0.2709	0.2508	0.2512
		(0.0004)	(0.0004)	(0.0004)	(0.0005)	(0.0003)	(0.0003)
	DQR	0.8363	0.8148	0.8348	0.9106	0.9143	0.9220
		(0.0120)	(0.0103)	(0.0128)	(0.0175)	(0.0170)	(0.0126)
	QRF	0.3944	0.3859	0.3860	0.3112	0.2953	0.2962
		(0.0003)	(0.0004)	(0.0004)	(0.0004)	(0.0004)	(0.0004)
	LQR	0.6294	0.6199	0.6207	0.5927	0.5671	0.5700
		(0.0039)	(0.0035)	(0.0036)	(0.0027)	(0.0021)	(0.0026)
R^2_τ	DQF	0.8042	0.8083	0.8091	0.7508	0.7660	0.7655
		(0.0003)	(0.0003)	(0.0003)	(0.0006)	(0.0007)	(0.0007)
	DQR	0.5600	0.5707	0.5609	0.1571	0.1407	0.1345
		(0.0033)	(0.0028)	(0.0030)	(0.0266)	(0.0325)	(0.0255)
	QRF	0.7921	0.7962	0.7963	0.7137	0.7248	0.7240
		(0.0003)	(0.0003)	(0.0003)	(0.0007)	(0.0007)	(0.0006)
	LQR	0.6686	0.6731	0.6729	0.4516	0.4680	0.4653
		(0.0013)	(0.0011)	(0.0011)	(0.0073)	(0.0070)	(0.0075)

Note: Sin function represents the variant v1; exp function represents the variant v2. The L_τ and R^2_τ are both recorded.