Cross-Sectional Analysis of Conditional Stock Returns: Quantile Regression with Machine Learning*

Haitao Li† Guoliang Ma‡ Cindy Yu§

April 2023

Abstract

We develop machine learning methods to forecast conditional quantiles of stock returns in the cross section through quantile regression. Machine learning makes it possible to capture highly nonlinear relations between conditional quantiles and a large number of return predictors. We adopt Bayesian optimization with Gaussian process that significantly improves the efficiency of hyperparameter tuning in machine learning. Simulation studies show that our methods accurately predict the conditional quantiles and consequently the whole conditional distributions of complicated data-generating processes. Empirical results show that our methods can identify stocks with extreme positive or negative returns and achieve superior performance in long-short investing.

Keywords— Cross-sectional return, Machine learning, Quantile regression, Conditional distribution, Bayesian optimization

*The computing support for the research reported in this paper in part was supported by the two National Science Foundation grants MRI1726447 and MRI2018594. The research also in part was funded by the Philip and Virginia Sproul Professorship at Iowa State University. All opinions, findings, and conclusions expressed in this papers are those of the authors.

†Cheung Kong Graduate School of Business, Beijing, China, 100738, Email: htli@ckgsb.edu.cn

‡Department of Statistics, Iowa State University, Ames, IA, USA, 50010, Email: glma@iastate.edu

§Department of Statistics, Iowa State University, Ames, IA, USA, 50010, Email: cindyyu@iastate.edu

1 Introduction

The modern asset pricing literature has been primarily focused on modeling expected stock returns in the cross section under the linear factor model framework of Sharpe (1963), Ross (1976), Fama and French (1993, 2015) and many others. A large number of firm characteristics have been identified in the literature that has predictive power for cross-sectional returns.[1] The important work of Gu et al. (2020) breaks the linear factor model tradition by developing machine learning models that allow for highly nonlinear relations between expected stock returns and a large number (up to 1,000) of return predictors. Gu et al. (2020) show that these machine learning models significantly outperform traditional linear factor models in forecasting future expected stock returns out of sample.

The literature has been less concerned about modeling the conditional distributions of future stock returns. However, conditional distributions contain much richer information than conditional expectations, given the well-known fact that stock returns tend to be highly skewed and exhibit fat tails and extreme values. For the 17,947 U.S. stocks included in our empirical analysis, Figure 1 provides the histograms of the skewness and kurtosis of each stock calculated using historical returns between 1987 and 2016.[2] Specifically, Figure 1a shows that most stocks are right-skewed with skewness ranging from about -5 to 15, while Figure 1b shows that most stocks exhibit fat tails with kurtosis ranging from 0 to 255.[3]

Figure 2 further illustrates potential advantages offered by conditional distributions in making investment decisions. Consider two firms whose monthly returns follow two de-meaned Gamma distributions, where $R_1 \sim -Gamma(5,1) + 5$ is left-skewed and $R_2 \sim Gamma(5,1) - 5$ is right-

[1] Green et al. (2013), Green et al. (2017), and Harvey et al. (2016) report more than 300 return predictive signals.

[2] Our empirical analysis uses the same data as that in Gu et al. (2020). Kindly made available by Professor Dacheng Xiu, it contains monthly returns of stocks listed on the NYSE, AMEX, and NASDAQ, 94 firm characteristics, and 12 macroeconomic variables between January 1978 and December 2016.

[3] Bai and Ng (2005) provide various tests for skewness, kurtosis, and normality for time series data.

skewed. While R_1 and R_2 both have zero means, Figures 2a and 2b show that R_1 has a median of 0.34%, while R_2 has a median of -0.34%. Some investors might prefer R_1 because it has a higher probability (50%) of exceeding 0.34% than R_2 (38%). Tail behaviors of return distributions become more crucial for investors who prefer stocks that can have extreme positive or negative returns. Figures 2c and 2d show that the 95% quantiles of R_1 and R_2 are 3.3% and 4.29%, respectively, whereas the 5% quantiles of R_1 and R_2 are -4.18% and -3.08%, respectively. It is clear that R_1 is more likely to exhibit extreme negative returns, and thus it could be preferred for short selling, while the opposite is true for R_2.

In this paper, we develop machine learning methods to forecast the conditional distributions of stock returns by forecasting the conditional quantiles through quantile regression. More specifically, denote $f(r_{i,t+1}|\boldsymbol{x}_{i,t})$ as the conditional density for firm i, where $r_{i,t+1}$ is the stock return between t and $t+1$ and $\boldsymbol{x}_{i,t}$ is a high dimensional vector of return predictors at t. For $\tau \in [0,1]$, let $q_\tau(\boldsymbol{x}_{i,t})$ denote the τ-th quantile function, which by definition is the inverse conditional cumulative distribution (*cdf*) function $F^{-1}(\tau|\boldsymbol{x}_{i,t})$, i.e., $q_\tau(\boldsymbol{x}_{i,t}) = F^{-1}(\tau|\boldsymbol{x}_{i,t})$. Machine learning makes it possible to model $q_\tau(\boldsymbol{x}_{i,t})$ as a highly nonlinear function of $\boldsymbol{x}_{i,t}$, and we estimate $q_\tau(\boldsymbol{x}_{i,t})$ using quantile regression based on large panel data to obtain $\hat{q}_\tau(\boldsymbol{x}_{i,t})$.[4] Therefore, if τ_j $(j = 1, 2, \ldots, J)$ are independently drawn from the uniform distribution on the unit interval, $\mathcal{U}_{[0,1]}$, then $\hat{q}_{\tau_j}(\boldsymbol{x}_{i,t}) = \hat{F}^{-1}(\tau_j|\boldsymbol{x}_{i,t})$ give legitimate random samples from the conditional density $f(r_{i,t+1}|\boldsymbol{x}_{i,t})$. Thus $\hat{q}_{\tau_j}(\boldsymbol{x}_{i,t})$ $(j = 1, 2, \ldots, J)$ can be used to form an empirical version of the conditional distribution and to calculate characteristics of the conditional distribution, such as mean, median, quantiles, tail probabilities, and higher moments (e.g., variance and skewness). In our empirical analysis, to simplify the numerical procedures, we consider $J = 100$ equally spaced grid points τ_j on the unit interval (ranging from 0.5% to 99.5%) and compute the corresponding $\hat{q}_{\tau_j}(\boldsymbol{x}_{i,t})$'s.

Conditional mean prediction in Gu et al. (2020) involves estimating linear and nonlinear ex-

[4]In our panel data analysis, the dependent and independent variables are, respectively, the τ-th quantiles and the return predictors of all the firms during the model training and validation period.

pected return functions of a large set of return predictors by minimizing the mean squared error. Similarly, conditional distribution prediction in this paper involves estimating linear and nonlinear conditional quantile functions of the set of predictors through quantile regression for each of the 100 τ_j's. While conditional mean prediction only requires one least squares regression, conditional distribution prediction is computationally more challenging because it requires 100 quantile regressions.

Gu et al. (2020) introduce various machine learning models to capture potential nonlinear dependence of the expected returns on the predictors. Given that the relationship between the conditional distributions and a large number of predictors could be even more complicated, we also adopt powerful machine learning models in our quantile regression analysis. In addition to the widely used linear models, such as principal component regression (PCR) and penalized linear regression (Lasso quantile regression), we also consider a three-layer perception neural network (NN3) and a tree-based boosting model, Light Gradient Boosting Machine (LightGBM), which has the potential to capture the complicated nonlinear dependence of quantiles on the predictors. While most existing studies on machine learning choose hyperparameters through grid search, we adopt Bayesian optimization with Gaussian process to tune the hyperparameters of LightGBM. The Bayesian optimization approach significantly improves the efficiency of hyperparameter tuning and leads to better model performance.

We conduct extensive simulation studies to examine our methods' ability to uncover complicated conditional distributions. The data-generating processes have highly skewed and heavy-tailed error terms in our simulation setup. The conditional mean can be either a linear or nonlinear function of the return predictors. And the error terms can be heterogeneous among all firm-time specific observations. For each setup, we generate 15 years of monthly returns for 200 firms, with 36,000 observations in total. For conditional distribution prediction, at the beginning of each year, we use the past nine years of data to train and validate machine learning models. Specifically, with chosen hyperparameters, we use the first seven years of data to train the models and the next two years of

data to validate their performance under the associated chosen hyperparameters. We then fix the model with the optimal hyperparameters and use the following twelve months of data to provide monthly out-of-sample forecasts of conditional distributions. We repeat this procedure until the end of the sample. We find that LightGBM has an excellent performance in capturing the conditional distributions for both the linear and nonlinear simulation setups, although the performance is slightly worse for the nonlinear case. NN3 and Lasso provide comparable forecasting accuracy in the linear case but degrade in the nonlinear setup. PCA cannot predict future return distributions, mainly due to the independence among simulated predictors.

We apply our methods to forecast the conditional distributions of the returns of all the U.S. stocks listed on the NYSE, AMEX, and NASDAQ between 1978 and 2016. Our empirical analysis follows the same tuning and training scheme used in the simulation studies. Since we need nine years of data for model training and validation, our out-of-sample forecasts start in January 1987 and continue monthly until December 2016.

We then evaluate the out-of-sample performance of the three machine learning models in three ways. First, we examine whether these models can accurately predict the central tendency of the returns, compared to the regression mean predictions, which are obtained by minimizing the mean squared error with the same machine learning methods. We also define *quantile mean* (q-mean hereinafter) as the average of the 100 predicted quantiles. We find that the regression mean, the q-mean, as well as the median (obtained from quantile regressions with $\tau = 0.5$) from the three machine learning methods perform equally well in predicting future realized returns with comparable forecasting errors.

Second, we evaluate the performance of the four machine learning models in forecasting the conditional densities using the likelihood-ratio (LR) statistical tests developed in Berkowitz (2001). The comparison shows that LightGBM produces the most accurate density forecasts, followed by NN3, PCR, and then Lasso. The proportions of firms that pass (at the 5% significance level) the Berkowitz's (2001) LR test are 80.32% for LightGBM, 70% for NN3, 70.20% for PCR, and 62.39%

for Lasso. In appendix C, we address the multiple comparisons problems that Chordia et al. (2020) stress.

Finally, we examine the performance of long-short portfolios constructed using regression mean, q-mean, and median from the three machine learning models. Stocks are sorted into long (short) portfolios if they have high (low) values of these measures. One of the most robust results is that the long-short portfolios sorted on median have higher cumulative returns and Sharpe ratios than those sorted on regression mean for all four models. Due to its robustness to outliers, the median could lead to lower in-sample over-fitting risk and better out-of-sample forecasts. Another result is based on q-mean. While the median has the highest profits for PCR and Lasso than q-mean and regression mean, q-mean has the best performance for LightGBM with significantly higher cumulative returns than the median and the regression mean. Given that q-mean essentially utilizes all the information of the conditional distributions to sort stocks, it is not surprising that q-mean works well for LightGBM, for it captures the conditional densities much better than PCR and Lasso.

We also consider other long-short portfolios sorted on other statistics derived from the predicted conditional distributions. These statistics include each single predicted quantile, pairs of predicted quantiles, and pairs of predicted tail probabilities of generating extreme positive and negative returns. Most long-short portfolios based on information of conditional distributions outperform the ones based on regression mean.

A huge amount of literature has been developed in economics and finance to model time-varying volatility and even the entire conditional distribution of economic and financial time series, such as the famous ARCH-GARCH models of Engle (1982) and Bollerslev (1986), and the regime-switching model of Hamilton (1989). Other models have considered non-Gaussian distributions, such as Student-t or Laplace distributions, for the error terms in time series models.[5] However, these

[5]See, for example, Kon (1984), Nelson (1991), Peiro (1994), Baixauli and Alvarez (2004), Kelly and Jiang (2014), and Hohberg et al. (2020), to cite just a few.

parametric models mainly focus on time series analysis. They could also be overly restrictive and fail to capture the true data-generating processes, as shown by Hong et al. (2004), Egorov et al. (2006), and Hong et al. (2007) in their studies of the performances of these types of models in out-of-sample density forecasts of spot interest rates, bond yields, and exchange rates. While several studies have adopted quantile regression in estimating conditional distributions, these studies typically (i) consider only linear quantile functions with a limited number of predictors and (ii) estimate the conditional distributions of a small number of time series[6].

Our paper makes important methodological and empirical contributions to the empirical asset pricing literature. Methodologically, we aaply machine learning methods for forecasting the conditional quantiles and distributions of the returns of thousands of individual stocks in the cross section. The approach is intuitive and can accommodate highly nonlinear relations between conditional quantiles and a large number of return predictors. We also introduce the Bayesian optimization approach based on Gaussian process, which significantly improves the computational efficiency of machine learning methods by automatically tuning the multivariate hyperparameters. Empirically, we provide strong evidence that conditional distributions contain much richer information than conditional means in forecasting future stock returns and can lead to much better results in long-short investing.

The rest of the paper is organized as follows. Section 2 introduces our methodology for forecasting conditional distributions based on quantile regression through machine learning. Section 3 provides simulation studies on the validity of our proposed methods. In Section 4, we conduct an empirical study that applies our methods to the U.S. stock market data between 1978 and 2016. Section 5 concludes.

[6]Melly (2005), Chernozhukov et al. (2013), and Callaway and Huang (2020) among others apply quantile regression to causal inference in labor economics, while Fortin et al. (2011) provide a review of distributional methods in economics. Machado and Mata (2005) applies linear quantile regression to predict conditional distributions using small panel data, while Zhao (2013) applies linear quantile regression to estimate conditional distributions using time series data. Instead of quantile regression, Foresi and Peracchi (1995) and Anatolyev and Baruník (2019) estimate conditional distributions in time series setting using odds ratios and hierarchical models, respectively.

2 Quantile Regression with Machine Learning

Suppose we observe stock returns on date $t \in \{1, \ldots, T\}$ for N_t listed firms, where the number of listed firms could change with time. Denote the return of firm i at time $t+1$ as $r_{i,t+1}$. The asset pricing literature has identified a long list of firm-level characteristics (see Green et al. (2017) and Gu et al. (2020) for a summary) that have potential predictive power for future stock returns, which we denote as a column vector $\boldsymbol{x}_{i,t}$ for firm i on date t.

Various methods can predict future stock returns for given $\boldsymbol{x}_{i,t}$. The simplest and most widely used one is linear regression. For example, we could estimate the expected returns using cross-sectional regression for each t. Then linear regression is equivalent to solving the following optimization problem at the specific t:

$$\hat{\boldsymbol{\beta}}_t = \min_{\boldsymbol{\beta} \in \mathbb{R}^p} \sum_{s \in \mathcal{T}_t} \sum_{i=1}^{N_s} (r_{i,s+1} - \boldsymbol{\beta}^\intercal \boldsymbol{x}_{i,s})^2, \tag{1}$$

where $\mathcal{T}_t$ contains the years used to train the model in order to forecast the returns in year $t+1$, $\boldsymbol{\beta}$ is a p-dimensional column vector of regression coefficients and the estimation of the expected return $\mathbb{E}_t(r_{i,t+1}|\boldsymbol{x}_{i,t})$ is $\hat{\boldsymbol{\beta}}_t^\intercal \boldsymbol{x}_{i,t}$.

To overcome the limitations of linear models, Gu et al. (2020) introduce nonlinear machine learning methods to forecast future stock returns by considering the following optimization problem:

$$\hat{f}_t = \min_{f \in \mathcal{F}} \sum_{s \in \mathcal{T}_t} \sum_{i=1}^{N_s} (r_{i,s+1} - f(\boldsymbol{x}_{i,s}))^2, \tag{2}$$

where $f(\boldsymbol{x})$ is potentially a nonlinear function in a function space $\mathcal{F}$, of $\boldsymbol{x}$, and the expected return $\mathbb{E}_t(r_{i,t+1}|\boldsymbol{x_{i,t}})$ is $\hat{f}_t(\boldsymbol{x}_{i,t})$.

However, both the linear and nonlinear models based on the least squares objective function focus on predicting the conditional means of future stock returns. Conditional means, however, fail

to capture the rich information contained in the conditional distributions of future returns.

One widely adopted way of modeling conditional distributions is to make parametric assumptions on the data-generating process. However, these parametric models could still be overly restrictive and may not be flexible enough to capture the highly complicated forms of conditional distributions. Moreover, the large number of predictors could result in a high dimensionality problem. We obtain conditional distributions of stock returns through quantile regression to overcome these challenges. We introduce machine learning methods to capture the complicated dependence of conditional quantiles on a large number of return predictors.

We present our quantile regression with machine learning methods as follows. In Section 2.1, we introduce the idea of quantile functions and quantile regression. In Section 2.2, we show how to assemble the estimated quantiles to obtain empirical forecasts of conditional distributions. In Section 2.3, we introduce the machine learning algorithms considered in this paper.

2.1 Quantile Regression

For a given $\tau \in (0,1)$, the conditional quantile function $q_\tau(\boldsymbol{x}_{i,t})$ is defined as a function satisfying $\Pr\left(r_{i,t+1} < q_\tau(\boldsymbol{x}_{i,t}) | \boldsymbol{x}_{i,t}\right) = \tau$. When $\tau = 0.5$, $q_{0.5}(\boldsymbol{x}_{i,t})$ represents the conditional median of $r_{i,t+1}$.

One easy way of understanding quantile regression estimation is by replacing the objective function in the least squares regression with other appropriate loss functions, and then we obtain estimates of conditional quantiles instead of the conditional mean. For example, if we assume a linear relation between quantiles and the predictors, i.e., $q_\tau(\boldsymbol{x}_{i,t}) = \boldsymbol{\beta}_{t,\tau}^\intercal \boldsymbol{x}_{i,t}$, we can estimate the median of future returns by minimizing the absolute deviation below:

$$\hat{\boldsymbol{\beta}}_{t,0.5} = \min_{\boldsymbol{\beta} \in \mathbb{R}^p} \sum_{s \in \mathcal{T}_t} \sum_{i=1}^{N_s} |r_{i,s+1} - \boldsymbol{\beta}^\intercal \boldsymbol{x}_{i,s}|. \tag{3}$$

Then the predicted median of $r_{i,t+1}$ is $\hat{\boldsymbol{\beta}}_{t,0.5}^\intercal \boldsymbol{x}_{i,t}$.

Suppose we are interested in estimating a generic τ-th quantile $q_{t,\tau}(\boldsymbol{x}_{i,t})$ instead of the median.

That is, we are interested in finding $\boldsymbol{\beta}_{t,\tau}$ such that $\Pr[r_{i,t+1} \leq \boldsymbol{\beta}_{t,\tau}^{\intercal} \boldsymbol{x}_{i,t} | \boldsymbol{x}_{i,t}] = \tau$, where $\tau \in [0,1]$. This can be achieved by minimizing the following absolute deviation loss (or check loss) function introduced by Koenker and Bassett (1978):

$$\rho_\tau(u) = u\left(\tau - \mathbb{1}(u < 0)\right). \tag{4}$$

It is interesting to note that $\rho_\tau(\cdot)$ is τ-specific, i.e., for a different τ, a specific check loss is used for estimation. The eight check loss functions for τ equally spanning from 0.01 to 0.99 in Figure 3 show that each objective function puts different weights on the data for the associated τ. Intuitively, when we are interested in the 15% quantile (the dashed orange curve in Figure 3), the slope is much steeper in the negative half, indicating smaller observations have more information about this small quantile than larger observations and should be given more weights in estimation. Similarly, for larger quantiles, say 85%, the objective function would put more weights on larger observations than smaller ones (as is shown by the dashed green curve in Figure 3).

Therefore, we could estimate the τ-th quantile by:

$$\begin{aligned}
\hat{\boldsymbol{\beta}}_{t,\tau} &= \min_{\boldsymbol{\beta} \in \mathbb{R}^p} \sum_{s \in \mathcal{T}_t} \left\{ \sum_{i=1}^{N_s} \rho_\tau(r_{i,s+1} - \boldsymbol{\beta}^{\intercal} \boldsymbol{x}_{i,s}) \right\}, \quad \text{or equivalently} \\
\hat{\boldsymbol{\beta}}_{t,\tau} &= \min_{\boldsymbol{\beta} \in \mathbb{R}^p} \sum_{s \in \mathcal{T}_t} \left\{ \sum_{i=1}^{N_s} (r_{i,s+1} - \boldsymbol{\beta}^{\intercal} \boldsymbol{x}_{i,s}) \left[\tau - \mathbb{1}(r_{i,s+1} < \boldsymbol{\beta}^{\intercal} \boldsymbol{x}_{i,s})\right] \right\},
\end{aligned} \tag{5}$$

and the predicted τ-th quantile of the conditional distribution of $r_{i,t+1}$ is $\hat{q}_{t,\tau}(\boldsymbol{x}_{i,t}) = \hat{\boldsymbol{\beta}}_{t,\tau}^{\intercal} \boldsymbol{x}_{i,t}$. The loss functions (4) and(5) are non-differentiable at 0. The simplex method and interior point are typical methods to minimize check losses.

In the above discussion, we assume a linear quantile function. In general, quantiles could depend on the predictors $\boldsymbol{x}_{i,t}$ nonlinearly. If we denote $q_{t,\tau}(\boldsymbol{x}_{i,t})$ as a general nonlinear function, then the τ-th quantile function can be estimated by minimizing the objective function below:

$$\begin{aligned}
\hat{q}_{t,\tau}(\cdot) &= \min_{q\in\mathcal{F}} \sum_{s\in\mathcal{T}_t} \left\{ \sum_{i=1}^{N_s} \rho_\tau \left(r_{i,s+1} - q(\boldsymbol{x}_{i,s}) \right) \right\} \\
\hat{q}_{t,\tau}(\cdot) &= \min_{q\in\mathcal{F}} \sum_{s\in\mathcal{T}_t} \left\{ \sum_{i=1}^{N_s} \left(r_{i,s+1} - q(\boldsymbol{x}_{i,s}) \right) \left[\tau - \mathbb{1}(r_{i,s+1} < q(\boldsymbol{x}_{i,s})) \right] \right\}.
\end{aligned} \tag{6}$$

The predicted τ-th quantile of the conditional distribution of $r_{i,t+1}$ is $\hat{q}_{t,\tau}(\boldsymbol{x}_{i,t})$.

2.2 From Quantiles to Conditional Distributions

Constructing conditional distributions from quantiles entails the inverse probability transformation. For $r_{i,t+1}$, denote its conditional *cdf* as $F(r_{i,t+1}|\boldsymbol{x}_{i,t})$. The τ-th theoretical quantile of $r_{i,t+1}$ conditional on $\boldsymbol{x}_{i,t}$ satisfies

$$\tau = \Pr\left[r_{i,t+1} < q_{t,\tau}(\boldsymbol{x}_{i,t}) | \boldsymbol{x}_{i,t} \right], \quad i.e., \tag{7}$$

$$\tau = F(q_{t,\tau}(\boldsymbol{x}_{i,t})|\boldsymbol{x}_{i,t}). \tag{8}$$

Applying the inverse of the conditional *cdf*, $F^{-1}(\cdot|\boldsymbol{x}_{i,t})$, on both sides, we obtain

$$F^{-1}(\tau|\boldsymbol{x}_{i,t}) = q_{t,\tau}(\boldsymbol{x}_{i,t}). \tag{9}$$

By the inversion probability transformation, we know that if $\{\tau_j\}_{j=1}^J$ is generated from $\mathcal{U}_{[0,1]}$, $\left\{ q_{t,\tau_j}(\boldsymbol{x}_{i,t}) \equiv F^{-1}(\tau_j|\boldsymbol{x_{i,t}}) \right\}_{j=1}^J$ are legitimate random samples drawn from the conditional density $f(\cdot|\boldsymbol{x}_{i,t})$. Thus for a large J, we can use $\left\{ \hat{q}_{t,\tau_j}(\boldsymbol{x}_{i,t}) \right\}_{j=1}^J$ to construct an empirical version of the conditional density. We can further use the forecasted empirical density to calculate characteristics

such as the probability that the next period return will exceed a given extreme cutoff r_c:

$$\Pr\left[r_{i,t+1} \geq r_c | \boldsymbol{x}_{i,t}\right] = \frac{1}{J}\sum_{j=1}^{J} \mathbb{1}(\hat{q}_{t,\tau_j}(\boldsymbol{x}_{i,t}) \geq r_c). \tag{10}$$

Another example is in parallel with the regression mean, $\mathbb{E}[r_{i,t+1}|\boldsymbol{x}_{i,t}]$ (i.e., conditional mean obtained from the least squares regression). Formally, we define the quantile mean (q-mean) as

$$\mathbb{E}_t^{QM}\left[r_{i,t+1}|\boldsymbol{x}_{i,t}\right] = \frac{1}{J}\sum_{j=1}^{J} \hat{q}_{t,\tau_j}(\boldsymbol{x}_{i,t}). \tag{11}$$

We can also define higher-order moments, including variance and skewness, from the predicted quantiles in a similar fashion. One drawback of random τ is that oftentimes a randomly generated τ_j from $\mathcal{U}_{[0,1]}$ could be very close to 0 or 1, causing numerical instability in estimation. Hence in practice, we consider 100 fixed grid points of τ_j's ranging from 0.5% to 99.5%. The validity of using fixed τ's to calculate equations (10) and (11) is theoretically justified in Appendix A.

These quantities calculated from predicted conditional distributions contain much richer information about future stock returns and provide investors with much richer choices in forming investment decisions. For example, while investors have been using expected return to sort stocks when forming long-short portfolios, median or q-mean could be used for the same purpose. The median is more robust to outliers than the mean, while q-mean in equation (11) aggregates all the information in all J quantiles. Moreover, conditional probability in equation (10) or quantile in its own right can help investors identify stocks that are likely to have extreme positive or negative returns. Stocks with the highest $\Pr[r_{i,t=1} \geq r_c|\boldsymbol{x}_{i,t}]$, for an appropriately chosen r_c, could be potential candidates to long, while stocks with the highest $\Pr[r_{i,t=1} \leq -r_c|\boldsymbol{x}_{i,t}]$ could be potential candidates to short. Similarly, for a given τ, we could choose stocks with the highest τ-th quantile to long and the lowest τ-th quantile to short. We could also consider a pair of quantiles (e.g., the 5% and the 95%), where we choose stocks with the highest 95% quantiles to long and those with

the lowest 5% quantiles to short. We will examine in our empirical analysis whether these new measures lead to better performances for long-short investing.

2.3 Machine Learning Algorithms

The important paper of Gu et al. (2020) demonstrates the power of sophisticated machine learning models in capturing nonlinear dependence of expected returns on return predictors. Given that the relation between conditional distributions and return predictors could be even more complicated, we also consider machine learning methods in our study to accommodate a large number of predictors $\boldsymbol{x}_{i,t}$ and to capture the potential nonlinear relations between conditional quantiles and the predictors. As equation (9) indicates, the quantile functions $q_{t,\tau}(\boldsymbol{x}_{i,t})$ uniquely determines the conditional *cdf* $F(r_{i,t+1}|\boldsymbol{x}_{i,t})$. Thus without any parametric restrictions on $q_{t,\tau}(\boldsymbol{x}_{i,t})$, our methods allow for very flexible shapes of the conditional distributions.

We consider four machine learning algorithms in our analysis. The first two are popular methods of dimension reduction that have been widely used in the literature to deal with large numbers of predictors. One is principal component regression (PCR), which exploits the structure of the covariance matrix of the predictors to reduce dimensionality. The other is Lasso, which is based on the idea of penalization. We also consider two non-linear algorithms, multilayer perception neural network and gradient boosting machines (trees). In particular, we consider the three-layer neural network and LightGBM, a version of the gradient boosting machine. All machine learning algorithms involve hyperparameters, and different values of hyperparameters could lead to very different model predictions. Below we first briefly introduce the three machine learning algorithms and then introduce the Bayesian optimization with Gaussian process for hyperparameter tuning. It is worth mentioning that for all four machine learning algorithms, for each year from 1987 to 2016, and each of the τ_j ($j = 1, 2, \ldots, 100$), we tune the hyperparameters to achieve the best out-of-sample forecast performance. In total, for each machine learning algorithm, we train $30 \times 100 = 3,000$

models and base our forecasting exercise on these models.

2.3.1 Principal Component Regression

The covariance matrix of the predictors $\boldsymbol{x}_{i,t}$ contains information about their comovements. Principal component analysis of the covariance matrix helps select combinations of $\boldsymbol{x}_{i,t}$ that explain the comovements. By replacing the original predictors $\boldsymbol{x}_{i,t}$ with their principal components in our quantile regression, we reduce the dimension of the covariates to at most m. Given a specific $\tau \in (0,1)$, for m principal components, the predicted quantile function takes the form

$$\hat{q}^{PCR}_{t,\tau}(\boldsymbol{x}_{i,t}) = \hat{\boldsymbol{\beta}}^{\intercal}_{t,\tau}(\boldsymbol{a}_1^{\intercal}\boldsymbol{x}_{i,t}, \ldots, \boldsymbol{a}_m^{\intercal}\boldsymbol{x}_{i,t})^{\intercal}, \tag{12}$$

where $\boldsymbol{a}$s are the principal loadings and $\hat{\boldsymbol{\beta}}_{t,\tau}$ is the vector of coefficients obtained by linear quantile regression as in equation (5) but with principal components as regressors. The only hyperparameter of PCR is the number of the principal components m.

2.3.2 Penalized Linear Methods

Belloni and Chernozhukov (2011) propose a uniformly consistent L_1-penalized estimator for quantile regression. Specifically, for a given τ, the penalized coefficients estimator is obtained as follows:

$$\hat{\boldsymbol{\beta}}_{t,\tau} = \min_{\boldsymbol{\beta}\in\mathbb{R}^p} \frac{1}{\sum_{s\in\mathcal{T}_t} N_s} \sum_{s\in\mathcal{T}_t}\sum_{i=1}^{N_s} \rho_\tau(r_{i,s+1} - \boldsymbol{\beta}^{\intercal}\boldsymbol{x}_{i,s}) + \frac{\lambda\sqrt{\tau(1-\tau)}}{\sum_{s\in\mathcal{T}_t} N_s} \sum_{k=1}^{p} \hat{\sigma}_k|\beta_k|, \tag{13}$$

where N_s and $r_{i,s+1}$ have been defined in the previous sections, p is the dimension of the covariates, $\hat{\sigma}_k^2 \coloneqq \frac{1}{\sum_{s\in\mathcal{T}_t} N_s}\sum_{s\in\mathcal{T}_t}\sum_{i=1}^{N_s} x_{i,s,k}^2$, and $x_{i,s,k}$ is the k-th element of $\boldsymbol{x}_{i,s}$. The final predicted quantile function is then

$$\hat{q}^{Lasso}_{t,\tau}(\boldsymbol{x}_{i,t}) = \hat{\boldsymbol{\beta}}^{\intercal}_{t,\tau}\boldsymbol{x}_{i,t}. \tag{14}$$

With the penalty term in equation (13), some components of $\hat{\boldsymbol{\beta}}_{t,\tau}$ will be shrunk to 0. Consequently, the effective dimension of the covariates could be reduced. The only hyperparameter in quantile regression with Lasso is λ in equation (13).

2.3.3 Neural Networks

We implement classic multilayer perception (MLP) networks with three hidden layers. In MLP, each layer contains a fixed number of cells that perform two transformations on the previous cells. Starting with the input layer (original $\boldsymbol{x}_{i,t}$), the cells in the first hidden layer (1) combine $\boldsymbol{x}_{i,t}$ linearly into $c_l^{(1)} \equiv b_1 + \boldsymbol{w}^{(1)}\boldsymbol{x}_{i,t}$, where b_1 and $\boldsymbol{w}^{(1)}$ are parameters and then (2) apply a non-linear activation function on the linear combination $\sigma(c_l^{(1)})$. The index l ranges from 1 to L, the total number of cells in the first layer. After the transformation, the output serves as the input to the second layer, and so forth. In the last layer, after linear combination, instead of applying yet another activation function, the outputs directly enter the check loss.

2.3.4 Tree-Based Methods

While PCR and Lasso can solve the problem of high-dimensionality, they are still linear models. To capture the potential nonlinear dependence of conditional quantiles on return predictors, we consider a tree-based method, Light Gradient Boosting Machine.

Recall that at time t, our goal is to find the best function $q(\cdot)$ in a function space $\mathcal{F}$ by

$$\hat{q}_{t,\tau} = \min_{q \in \mathcal{F}} \sum_{s \in \mathcal{T}_t} \sum_{i=1}^{N_s} \rho_\tau(r_{i,s+1} - q(\boldsymbol{x}_{i,s})). \tag{15}$$

Due to the high-dimensionality of $\boldsymbol{x}_{i,t}$ and the flexibility of $q(\cdot)$, it is very difficult to accomplish the above minimization in one step. Instead, boosting trees start with a simple although non-optimal function and iteratively improve this function until reaching a final prediction function that aggregates the initial one and all the subsequent improvements.

Specifically, for a given τ, we start by estimating a simple binary tree $\hat{q}^{[0]}_{t,\tau}(\boldsymbol{x}_{i,t})$ that minimizes the quantile loss $\rho_\tau(\cdot)$ based on the original data $\{r_{i,t+1}, \boldsymbol{x}_{i,t}\}_{t=1,\ldots,T;i=1,\ldots,N_t}$. This naive binary tree, however, certainly cannot make perfect predictions. From this initial tree, we obtain the "τ-residual"s from all the observations $\{r_{i,t+1} - \hat{q}^{[0]}_{t,\tau}(\boldsymbol{x}_{i,t})\}$. These residuals contain information on how far the naive tree is from optimal. In the next iteration, gradient boosting machine works on the τ-residuals by building another tree from the derived data $\{r_{i,t+1} - \hat{q}^{[0]}_{t,\tau}(\boldsymbol{x}_{i,t}), \boldsymbol{x}_{i,t}\}$. This new tree, denoted as $\hat{q}^{[1]}_{t,\tau}(\boldsymbol{x}_{i,t})$, captures the incapability of $\hat{q}^{[0]}_{t,\tau}$. Continuing in this fashion for S iterations, more binary trees are built to exploit the information contained in the original data. Aggregating all the trees, we obtain the final prediction function as

$$\hat{q}^{LightGBM}_{t,\tau}(\boldsymbol{x}_{i,t}) = \sum_{s=0}^{S} \hat{q}^{[s]}_{t,\tau}(\boldsymbol{x}_{i,t}), \tag{16}$$

where S is the number of boosting iterations.

Several improvements and various implementations are developed based on the idea of naive boosting trees. For example, gradient boosting trees shorten the computing time by approximating the loss function (the check loss ρ_τ in our context) with second-order Taylor expansion. XGBoost employs histogram-based splitting that reduces the number of possible splits of each tree to reduce computing time. It also adopts more regularizations to control over-fitting. LightGBM is by far the fastest and stale implementation of gradient boosting trees. It accelerates the computing through histogram-based sampling, gradient-based one-side sampling, and exclusive feature bundling. For a detailed discussion of LightGBM, see Ke et al. (2017). Although the descendants of boosting trees differ in various aspects, their differences are mainly in the technical implementations. In our paper, we use LightGBM to represent the tree-based method. The hyperparameters we consider for LightGBM are the learning rate, the maximum depth of a tree, and the minimum number of observations on a leaf.

For all four machine learning algorithms, we obtain the 100 predicted quantiles for each firm in

each month from 1987 to 2016 and calculate several interesting summary statistics, including the tail probabilities of equation (10) and q-mean of equation (11). We then use these quantities to construct long-short portfolios.

2.3.5 Hyperparameter Tuning

Hyperparameters govern the structure of machine learning models, and different hyperparameters could lead to very different models. As a result, hyperparameter tuning, i.e., selecting the optimal hyperparameters to construct models for forecasting purposes, is crucial for the performance of machine learning models.

In both simulation studies and empirical analysis below, we use nine years of in-sample monthly data to tune the hyperparameters of all three machine learning algorithms. Specifically, we first fix a hyperparameter and use the first seven years of data to train the model and get the estimate of the model parameters. We then use the next two years of data for validation. That is, based on the model parameters estimated using the first seven years of data, we choose the hyperparameters that give the best fit using the recent two years of data. Then based on the selected hyperparameters, we retrain the model using the entire nine years of data. After this exercise, we end up with a set of hyperparameters as well as the model parameter estimates that jointly determine a machine learning model that has the best fit for the nine years of in-sample data. We then use this model to make monthly out-of-sample forecasts of conditional distributions for the next twelve months.

Grid search is the standard approach for hyperparameter tuning. That is, one would search through the space of the hyperparameters point-by-point and find the values that give the best out-of-sample fit. Grid search is relatively straightforward for PCR and Lasso, which have only one hyperparameter (m for PCR and λ for Lasso). For example, when validating PCR, we would evaluate the model 50 times for each $m \in \{1, 2, \ldots, 50\}$ and find the one that gives the best fit. While grid search would also work for quantile Lasso, we adopt the suggestion by Belloni and Chernozhukov (2011) to choose the best in-sample hyperparameter λ (see their equation (2.7) for

explicit computation steps) for computational efficiency.

When the number of hyperparameters increases, however, the computational burden of grid search increases exponentially, and the tuning process becomes extremely time-consuming. The main reason is that grid search is, in essence, enumeration. It just blindly searches over the hyperparameter space without any guidance on the locations that are more likely to contain better hyperparameter values.

To overcome this challenge, we adopt Bayesian optimization with Gaussian process for hyperparameter tuning and apply it to NN3 and the LightGBM model, each of which has three hyperparameters[7]. Bayesian optimization, first proposed by Močkus (1975), is an optimization algorithm that finds the optimal value of a function by iteratively searching through its domain. Unlike gradient-based optimization algorithms, Bayesian optimization does not require any information on derivatives and therefore is more suitable for optimization problems where the objective functions are not known in closed form.

In the context of tuning hyperparameters, Bergstra et al. (2011), Snoek et al. (2012), and Turner et al. (2021) have shown that Bayesian optimization offers an effective approach to hyperparameter tuning, which is essentially an optimization problem. That is, the goal is to choose hyperparameters that minimize the validation loss[8]:

$$\min_{\boldsymbol{\xi}\in\boldsymbol{\Xi}} \nu(\boldsymbol{\xi}), \tag{17}$$

where $\boldsymbol{\xi}$ represents the hyperparameters, $\boldsymbol{\Xi}$ is the domain of the hyperparameters, and ν represents the validation loss function. The main challenges we face are that ν is generally not known with an explicit analytic formula and is computationally expensive to evaluate.

The basic approach of Bayesian optimization is to first assign a prior distribution to the unknown $\nu(\cdot)$ function. Then evaluate $\nu(\cdot)$ at a randomly chosen hyperparameter $\boldsymbol{\xi}$. Based on this new

[7]Dropout, initial learning rate, and decay for NN3 and learning rate, minimal points on each leaf, and maximum depth of a tree for LightGBM.

[8]Bayesian optimization typically finds the maximal; we use minimization for illustrative purposes.

information, we update the posterior distribution to obtain a posterior distribution using the Bayes rule. The posterior distribution would provide guidance on where to choose the next hyperparameter for the next-round evaluation. This process will continue until convergence is reached. When choosing the prior distributions of the (typically unknown) objective function, Gaussian process is widely used due to its computational convenience.

In our implementation, the prior of $\nu(\cdot)$, denoted as $\nu^{(0)}$, is assumed to follow a Gaussian process, with pre-specified mean and covariance structures. Specifically, at any $\boldsymbol{\xi} \in \boldsymbol{\Xi}$, $\nu^{(0)}(\boldsymbol{\xi})$ follows a Gaussian distribution with constant mean $\mu \in \mathbb{R}$ and for any two points $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2 \in \boldsymbol{\Xi}$, the covariance between the two points $\nu^{(0)}(\boldsymbol{\xi}_1)$ and $\nu^{(0)}(\boldsymbol{\xi}_2)$ is determined by a Gaussian kernel

$$\text{Cov}\left(\nu(\boldsymbol{\xi}_1), \nu(\boldsymbol{\xi}_2)\right) \equiv \frac{1}{2\pi\sigma^2} \exp\left\{-\frac{\|\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2\|_2}{2\sigma^2}\right\}, \tag{18}$$

where $\|\cdot\|_2$ denotes the Euclidean norm and σ^2 is the prior variance parameter.

Once the prior is specified, we randomly sample a point $\boldsymbol{\xi}^{(0)}$ and evaluate $\nu(\boldsymbol{\xi}^{(0)})$. Then the posterior of $\nu(\cdot)$, denoted as $\nu^{(1)}(\cdot)$, can be easily updated given the new information obtained from $\nu(\boldsymbol{\xi}^{(0)})$ and the Gaussian nature of the prior distribution $\nu^{(0)}(\cdot)$. Based on the updated information, we choose a new hyperparameter $\boldsymbol{\xi}^{(1)}$ that is most likely to offer the smallest out-of-sample forecasting error. Next, we can obtain a newer posterior $\nu^{(2)}(\cdot)$ based on the prior $\nu^{(0)}(\cdot)$ and the evaluation at the two selected hyperparameters $\nu(\boldsymbol{\xi}^{(0)})$ and $\nu(\boldsymbol{\xi}^{(1)})$. We then select a newer hyperparameter $\boldsymbol{\xi}^{(2)}$ that is most likely to offer the smallest out-of-sample forecasting error. Continuing in this fashion, we update the posterior and select the hyperparameters M times. At last, the best hyperparameter $\boldsymbol{\xi}^{opt}$ is the one that gives the minimum value of $\left\{\nu(\boldsymbol{\xi}^{(0)}), \nu(\boldsymbol{\xi}^{(1)}), \ldots, \nu(\boldsymbol{\xi}^{(M)})\right\}$. We choose $M = 20$ in our empirical analysis. Note that in this process, we do not require that $\nu(\cdot)$ is known with an explicit analytic formula. Instead we only require that $\nu(\cdot)$ can be evaluated numerically.

One main advantage of Bayesian optimization is that it requires only a few iterations to converge

due to its guided updating nature. For example, if a randomly chosen new hyperparameter $\boldsymbol{\xi}$ leads to a good fit of the data, then those hyperparameters that exhibit high covariance in the Gaussian process with $\boldsymbol{\xi}$ are also likely to be good choices and vice versa. An acquisition function precisely determines the next-round hyperparameters. In our analysis, we use expected improvement as the acquisition function. We refer interested readers to Shahriari et al. (2015) for details on other choices of acquisition functions. As a result, the algorithm only needs to search those promising regions, which greatly enhances the efficiency of hyperparameter tuning. In contrast, grid search, by completely discarding the information contained in earlier searches and going over the hyperparameter domain blindly, has to evaluate the objective function unnecessarily many times. This makes grid search practically infeasible when the number of hyperparameters is large.

3 Simulation

This section presents our simulation analysis to investigate the efficacy of the four machine learning models for quantile regression in forecasting conditional distributions. Our covariates generating process is similar to that of Gu et al. (2020) but we consider conditional mean functions that exhibit both linear and nonlinear dependence on return predictors. The linear case has independently and identically distributed (i.i.d.) asymmetric random errors, whereas the nonlinear case has heterogeneous asymmetric random errors.

In the following, we first introduce the simulation setup, including the predictors and the return generating process in Section 3.1. We then present the forecasting results from the four machine learning methods in Section 3.2.

3.1 Return Generating Processes

In our simulation, we consider a balanced panel of N firms with T months of observations. For time $t \in \{1, \ldots, T\}$ and firm $i \in \{1, \ldots, N\}$, stock returns are obtained from the following data-generating process:

$$\begin{aligned} r_{i,t+1} &= g^*(\boldsymbol{x}_{i,t}) + \boldsymbol{\beta}_{t+1}^\intercal \boldsymbol{x}_{i,t} + \varepsilon_{i,t+1}, \\ \boldsymbol{\beta}_{t+1} &\overset{iid}{\sim} \mathcal{N}\left(\mathbf{0}, \sigma^2 \boldsymbol{I}_3\right), \\ \varepsilon_{i,t+1} &\text{ is the random error to be specified,} \end{aligned} \tag{19}$$

where $\boldsymbol{x}_{i,t}$ is a 3×1 vector representing the return predictors, $g^*(\boldsymbol{x}_{i,t})$ is the conditional mean for firm i at time t, $\boldsymbol{\beta}_{t+1}$ is a 3×1 vector independent of $\boldsymbol{x}_{i,t}$, $\boldsymbol{\beta}_{t+1}^\intercal \boldsymbol{x}_{i,t}$ represents the time effects, and $\boldsymbol{I}_3$ is a 3×3 identity matrix. For each t, we actually simulate a 50-dimensional vector of predictors to represent the high dimensionality in the real data. However, instead of using all 50 predictors to generate the returns, we choose three of them, treating the remaining ones as spurious factors. Therefore, in the return-generating process, only three predictors are effective, while in our forecasting exercises, we use all 50 predictors as the input of the machine learning models.

The return predictors are generated according to the following data-generating process.[9] The k-th element of $\boldsymbol{x}_{i,t}$ is simulated as:

$$x_{i,t,k} = \frac{2}{N+1} \text{rank}_{\varsigma_{i,t,k}} - 1, \tag{20}$$

where

$$\begin{aligned} \varsigma_{i,t,k} &= \rho_k \varsigma_{i,t-1,k} + \epsilon_{i,t,k}, \\ \epsilon_{i,t,k} &\overset{iid}{\sim} \mathcal{N}\left(0, 1 - \rho_k^2\right), \text{ where } \rho_k \overset{iid}{\sim} \mathcal{U}_{[0.9,1]}. \end{aligned} \tag{21}$$

Also note in equation (21), for each $k \in \{1, 2, \ldots, 50\}$, we let ρ_k's be i.i.d. and $\epsilon_{i,t,k}$ be i.i.d. for all

[9] We rank the predictors for each t, so the transformed predictors are homogeneous across times, and their values are always between -1 and 1.

i and t. We set the initial values of $\varsigma_{i,t,k}$ to be zero but discard the first 180 months of observations as burn-in.

We consider two functional forms of $g^*(\cdot)$:

$$g^*(\boldsymbol{x}_{i,t}) = 0.3 + 0.3x_{i,t,1} + 0.2x_{i,t,2} + w_t x_{i,t,3}, \tag{22a}$$

$$g^*(\boldsymbol{x}_{i,t}) = 1 + 0.4x_{i,t,1} + 2x_{i,t,1}x_{i,t,2}^2, \tag{22b}$$

where w_t represents a market-wide effect and follows an AR(1) process:,

$$w_t = \rho w_{t-1} + u_t, u_t \overset{iid}{\sim} \mathcal{N}\left(0, 1-\rho^2\right), \text{ for } \rho = 0.9. \tag{23}$$

For the first setup (22a), which assumes a linear relationship between expected return and the covariates, we adopt a skewed t-distribution error term $\varepsilon_{i,t+1} \sim t_5(0, \sigma_\varepsilon^2)$, where $\sigma_\varepsilon = 0.05$ is the scale parameter. The second setup (22b) assumes a nonlinear relation between the expected return and the covariates, and we adopt a random error that follows heterogeneous one-parameter asymmetric Laplacian distributions (ALD). That is, $\varepsilon_{i,t+1} \sim ALD(\kappa_i)$, where $\kappa_i \sim \mathcal{N}(1, 0.15)$. ALD is known for its asymmetric shape with heavy tails (see Linden (2001) for example). For the one-parameter case as in our simulation, $\kappa > 0$ is the asymmetry parameter. $- < \kappa < 1$ results in an ALD with a heavy right tail, and $\kappa > 1$ results in an ALD with a heavy left tail. Another interesting feature about ALD is that unless for $\kappa = 1$, which represents the symmetric distribution, the mean of ALD is not 0. In our simulation, we guarantee that the simulated κ_i's are all greater than 0, and all the error terms are de-meaned. The second setup is more difficult to estimate because of the nonlinear relationship and the more complicated structure of the noise terms.

For each setup, we simulate 15 years (180 months) of data, with 200 firms in each month, totaling 36,000 observations. At the beginning of each year, we use the past nine years of data to

completely train a machine learning model: the first seven years for model training and the next two years for model validation. Then we use the trained model to make monthly out-of-sample forecasts of conditional return distributions for the next twelve months. Effectively, we have only six years of data with 14,400 firm-level observations for out-of-sample evaluation. Whereas Gu et al. (2020) focuses on conditional mean prediction, we are more interested in forecasting conditional distributions.

Table 1 reports the decomposition of the total variations of the simulated returns for both the linear and nonlinear models. We find that the simulated returns of the nonlinear model are much more volatile than that of the linear model. For example, the total variance of the simulated returns in the nonlinear model (2.6470) is about ten times that in the linear model (0.2671). The biggest contribution to such a huge difference is the firm-specific risk generated by the extreme values of the ALD error term. The firm-specific risk accounts for about 80% (90%) of the total variance (standard deviation) of the simulated returns in the nonlinear model. While the variance of the conditional mean and the time effect in the nonlinear model are larger than that in the linear model, their contributions to the total variations of the simulated returns are much smaller. In general, it should be much more difficult to forecast the conditional distributions of the nonlinear model, which has a much more complicated model structure, more time variations in model coefficients, and much higher total and firm-specific risks.

3.2 Results

We first examine model performance by comparing certain conditional quantiles (at 20%, 50%, and 80%) of the true conditional distributions with the corresponding ones predicted by the three machine learning methods. For a given set of predictors, the conditional mean of the true distribution is known. In order to obtain the true conditional quantiles, we conduct a Monte Carlo simulation of size 500 from the true return generating process to capture the time-varying effects

and firm-specific risks.

Figure 4 compares the true 20%, 50%, and 80% quantiles obtained from Monte Carlo simulation with the quantiles predicted by the fully trained PCR, Lasso, NN3, and LightGBM models for the linear setup in (22a). The horizontal axis is the true quantile, and the vertical axis is the forecasted quantile. Each dot represents a firm-time observation. The red lines are at 45 degrees and pass the origin. From top to bottom, we report the results of PCR, Lasso, NN3, and LightGBM, respectively. We note that Lasso, NN3, and LightGBM can accurately predict the true quantiles for the linear case with relatively small return variation. In contrast, PCR fails to accurately predict the true quantiles. This is mainly because the simulation setup does not impose any specific structures on the covariance matrix of the predictors, which makes it hard to take advantage of the power of PCR.

Figure 5 compares the true 20%, 50%, and 80% quantiles obtained from Monte Carlo simulation with the quantiles predicted by the three fully trained machine learning models for the nonlinear setup in (22b). From top to bottom, we report the results of PCR, Lasso, NN3, and LightGBM, respectively. It is easily seen that LightGBM produces the closest prediction among the three. Although the band is wider than that in Figures 4j, 4k, and 4l, the LightGBM predictions still closely match the truth. The Lasso and NN3 predictions capture the trend but with wider bands. They both fail at lower and higher ranks of quantiles. For example, the Lasso prediction tails are much shorter than those of the true distributions, suggesting Lasso misses the information in tails of returns. PCR predictions is the worst among the four, and can only reveal the general location of each quantile. Given the more complicated data-generating process in equation (22b), the above results illustrate the power of LightGBM in capturing highly nonlinear and complicated patterns in noisy data.

Next, we compare the true conditional distributions of returns with the conditional distributions predicted by the three machine learning methods through quantile regression. Out of the 14,400 out-of-sample observations, we randomly select three firm-time-specific conditional distributions.

Given the set of predictors of a chosen firm, we obtain the true distributions through Monte Carlo simulation from the return-generating processes and the predicted conditional distributions through quantile regression estimations for each machine learning method.

Figure 6 compares the three randomly selected firm-time specific true conditional densities with the corresponding predicted conditional densities of each machine learning method for the linear setup in (22a). From top to bottom, we report the results of PCR, Lasso, NN3, and LightGBM. In each subplot, the gray histogram indicates the simulated true density, while the yellow histogram shows the predicted density. Similar to Figure 4, Lasso, NN3, and LightGBM can accurately predict the conditional densities, while PCR fails to do so in this simplified linear setting.

Figure 7 provides similar results on density comparisons for the nonlinear setup in (22b). Consistent with the results on quantile predictions, NN3 and LightGBM can accurately predict the true conditional distributions of the three randomly selected firms. While Lasso and PCR have reasonable performance for one of the firms, they fail badly for the others. In the results not shown here, we have compared many more firm-time observations and found that LightGBM consistently provides excellent predictions while the other two methods work well only for a small portion of the firms. These results further demonstrate the superiority of the gradient tree-based method in capturing the complicated patterns in the data.

In summary, our simulation results show that NN3 and LightGBM produce the most consistent and robust performances in predicting both quantiles and the whole conditional distributions for relatively simple as well as highly complex return-generating processes. However, even though NN3 shows similar performance to LightGBM in our simulation studies, it took 30 times more computing time, which renders full convergence of the algorithm nonfeasible in empirical analysis. As we present in the real data application, even allowed for much more computing time than LightGBM, NN3 can hardly beat LightGBM. While Lasso has similar performances for the linear case (22a), its performance deteriorates significantly for the nonlinear case. PCR has the worst performance among the three models for both linear and nonlinear cases, partly due to the lack

of structure in the covariance matrix in the simulation setup. We include all four methods in our empirical analysis for comparison.

4 Empirical Analysis

In this section, we apply the four machine learning methods for quantile regression to forecast the conditional distributions of returns of individual stocks listed in the U.S. stock market. We first introduce the data used in our empirical analysis in Section 4.1, which is very similar to that used in Gu et al. (2020). We then discuss the out-of-sample performance of our methods in predicting realized returns and conditional distributions in Section 4.2. Finally, we examine the performance of long-short investment portfolios constructed based on distributional information in Section 4.3.

4.1 Data Description

Our empirical analysis focuses on the monthly returns of stocks listed on the NYSE, AMEX, and NASDAQ between January 1978 and December 2016. With the addition of NASDAQ firms after 1971, the number of firms changed dramatically in 1973. We decide to start our sample in 1978, leaving 60 months for the market to stabilize.

Since we require nine years of data for model training and validation, we can make out-of-sample forecasts only for the years between 1987 and 2016. The forecasting sample contains 17,947 firms with 1,809,418 firm-year-month observations. At the beginning of 1987, using a model trained between 1978 and 1984 and validated between 1985 and 1986, we make monthly forecasts of the conditional distributions of stock returns for each month in 1987. That is, for each month and each firm, we predict the corresponding 100 quantiles of the returns for 100 equally spaced grid points ranging from $\tau = 0.5\%$ to $\tau = 99.5\%$. We repeat the process one year at a time until 2016, the final year of the sample. We report our tuned optimal hyperparameters for all the quantile

models in all the years in Appendix B. For comparison purposes, we also make monthly forecasts of expected stock returns following the same training scheme, except that the machine learning models are trained and validated with the least-squares loss.

While we adopt almost the same set of firm characteristics of Green et al. (2017) and Gu et al. (2020) and macroeconomic variables of Goyal and Welch (2008) as return predictors, several important differences exist in our analysis. First, we exclude the cross-sectional premium in Goyal and Welch (2008) since it contains too many missing values, although we still keep market capitalization, long-term yield, risk-free rate, inflation, and high-yield bond yield as macroeconomic variables. Second, we drop the interactions between firm characteristics and macroeconomic variables to avoid over-fitting. Finally, we exclude firms from the financial industry ($60 \leq$SIC≤ 67) and convert the two-digit SIC industry classification to Fama-French 10 industry classification to mitigate the sparsity of the industry index matrix. Following Gu et al. (2020), all the remaining return predictors have been standardized through rank transformation. For detailed construction of the security-specific predictors, see Green et al. (2017) and Gu et al. (2020).

4.2 Conditional Return Distributions

In this section, we first examine the ability of the four machine learning methods to capture the central tendency of future stock returns. Specifically, we use regression mean, median, and q-mean (the average of the 100 imputed quantiles) of each model predicted by each machine learning algorithm to forecast realized returns and report the summary statistics of the forecasting errors in Table 2. We denote $|r_{i,t+1} - \hat{r}_{i,t+1}|$ as the absolute forecast error, where $\hat{r}^{QM}$, $\hat{r}^{0.5}$, and $\hat{r}^{reg}$ represent predictions made by q-mean, median, and regression mean, respectively. The summary statistics of the absolute forecast errors clearly show that for each machine learning method, median, q-mean, and regression mean have similar abilities in forecasting realized returns. Therefore, the advantage of quantile regression does not seem to be reflected in forecasting the center of return distributions.

We then evaluate the performance of the machine learning methods in forecasting the conditional distributions of stock returns. Diebold et al. (1998) propose to evaluate density forecasts based on the idea of generalized residuals introduced in Rosenblatt (1952). For each firm i, the conditional *cdf* of stock returns given predictors $\boldsymbol{x}_{i,t}$ at time $t+1$, $\hat{F}(r_{i,t+1}|\boldsymbol{x}_{i,t})$, is called a generalized residual of the conditional density. According to Rosenblatt (1952) and Diebold et al. (1998), if the forecasted conditional density $\hat{F}(\cdot|\boldsymbol{x}_{i,t})$ captures the true return generating process, the generalized residuals, $\hat{F}(r_{i,t+1}|\boldsymbol{x}_{i,t}),\ t = 0, \ldots, T-1$, should follow i.i.d. $\mathcal{U}_{[0,1]}$. Different methods have been developed to evaluate density forecasts by testing the hypothesis of i.i.d. $\mathcal{U}_{[0,1]}$.

The well-known Kolmogorov-Smirnov (KS) test measures the distance between the *cdf*s of two random variables. Under the null hypothesis, the two *cdf*s should be the same, implying a small KS distance. In our case, the KS distance is between the firm-level empirical distribution of the generalized residuals and the $\mathcal{U}_{[0,1]}$. Berkowitz (2001) points out several disadvantages of the KS test and proposes to transform the generalized residuals with the inverse *cdf* of the standard normal distribution. Under the null hypothesis, the transformed generalized residuals should follow the standard normal distribution, which can be tested using the Shapiro normality test.

While the above two tests mainly focus on testing the shape property of the *cdf* of the generalized residuals, Berkowitz (2001) develop a likelihood ratio (LR) test that jointly tests the serial dependence and the shape of the generalized residuals.[10] Whereas the shape property measures whether a given model captures the true conditional density at each t, the serial independence measures whether the predicted conditional distributions fully absorb the dynamic properties of the data-generating processes. Specifically, the LR test assumes the transformed generalized residuals follow an AR(1) process and tests jointly whether the AR(1) coefficient is zero and whether the error term follows the standard normal distribution. We apply all three tests (KS, Shapiro, and LR) in our empirical analysis for a complete discussion.

[10] Hong and Li (2005) develop nonparametric tests that jointly test whether the generalized residuals follow the uniform distribution and are independently distributed.

Table 3 reports the summary statistics of the p-values of the KS, Shapiro, and LR tests. Large p-values suggest that we do not have enough evidence to reject the null hypothesis that the machine learning methods can capture the conditional distributions of returns. Consistent with our simulation evidence, all three tests show that LightGBM can better capture complex conditional distributions than the other two methods. For example, LightGBM has higher median and mean p-values for all three tests than NN3, PCR, and Lasso: the median and mean p-values of the LR tests for LightGBM are 0.3003 and 0.3623, respectively, followed by 0.2331 and 0.3216 for NN3, 0.2018 and 0.3044 for PCR, and 0.1272 and 0.2525 for Lasso. Moreover, at least 80% of the firms under LightGBM have p-values for all three tests greater than 5%, suggesting that LightGBM can accurately forecast the conditional distributions of 80% of the firms in our sample. The proportion lowers to about 70%, 70%, and 60% for NN3, PCR, and Lasso, respectively.

4.3 Long-Short Portfolios

The above analysis confirms that the machine learning methods, particularly the LightGBM model, can forecast the conditional distributions of stock returns for most firms in our sample accurately. In this section, we examine whether conditional distributions offer additional information beyond expected returns in forming long-short portfolios for investment purposes. Although the forecast is implemented annually, we rebalance at a monthly frequency to construct long and short portfolios. We form decile portfolios to exercise long-short investing.

4.3.1 Portfolios Sorted on q-Mean, Median, and Mean

In addition to the regression mean (i.e., expected return), which has been widely used as a sorting variable in long-short investing, we also use q-mean, constructed from the forecasted conditional distributions as in equation (11), and median to sort stocks into long and short portfolios. Figure 8 reports the cumulative returns of the long and short portfolios based on the three sorting variables

for the four machine learning methods from 1987 to 2016. From top to bottom, the four panels represent the results for PCR, Lasso, NN3, and LightGBM, respectively. The solid lines represent the cumulative returns of the long portfolios. In contrast, the dashed lines represent the short portfolios. Different colors represent the results for the three sorting variables under each model.

One of the most consistent results in Figure 8 is that under all machine learning models, the median leads to better results in long-short investing than the regression mean. Median-sorted cumulative long-short returns over the 30 years under PCR, Lasso, NN3, and LightGBM are 474%, 828%, 739%, and 1036%, respectively, while the corresponding results for regression mean are -45%, 68%, 113%, and 360%, respectively. The Median, which is more robust to outliers than the mean, could lower the risk of in-sample fitting and thus leads to better out-of-sample performance. In contrast, one of the reasons that regression means under PCR leads to negative long-short returns could be due to the incompetency of linear methods since we do not include explicit interactions between return predictors as in Gu et al. (2020). Another reason for the poor performance of PCR is due to the rank transformation, which hinders the method from successfully extracting the potential variance-covariance structure from the original predictors. Although the transformed data work well in Gu et al. (2020), our samples have different coverages, as discussed in Section 4.1.

Figure 8 also shows that q-mean under LighGBM leads to the best results in long-short investing with a cumulative return of 1970% over the 30 years, which is significantly higher than that of all other sorting variables under all four machine learning methods. We defer the discussion of the superior performance of LightGBM q-mean to the end of Section 4, after presenting other metrics of portfolio performance.

Table 5 reports the annualized Sharpe ratios of the cumulative long-short portfolios constructed using q-mean, median, and regression mean. Although q-mean does not lead to higher cumulative long-short returns than the median under PCR and Lasso, Table 5 shows that q-mean leads to higher Sharpe ratios than all other sorting variables under all machine learning models. Consistent

with previous results, the regression mean has the lowest Sharpe ratios among the three sorting variables.

Figure 9 reports the average monthly returns of the decile portfolios sorted by q-mean, median, and regression mean. It is interesting to see that the average monthly returns of the decile portfolios sorted by q-mean and median exhibit a clear monotonic pattern, from the lowest to the highest, although the return spread is generally much higher for q-mean under the LightGBM model. In contrast, the average monthly returns of the decile portfolios sorted by regression mean do not exhibit a clear monotonic pattern, and the return spreads are not as significant either.

The above results suggest that conditional distributions contain much richer information than regression means about future stock returns. Moreover, consistent with simulation results, LightGBM has the best performance, given its flexibility in capturing the potential nonlinear patterns of the returns for given predictors.

4.3.2 Portfolios Sorted on Conditional Probabilities and Quantiles

To take full advantage of the information contained in conditional distributions, we consider three additional sorting variables for long-short portfolios. First, given a cutoff $r_c \geq 0$, we select stocks with the highest $\Pr[r_{i,t+1} \geq r_c|\boldsymbol{x_{i,t}}]$ to long, and stocks with the highest $\Pr[r_{i,t+1} \leq -r_c|\boldsymbol{x}_{i,t}]$ to short. The latter probability is defined similarly as in equation (10). Second, for a given τ_j, we choose stocks with the highest $\hat{q}_{t,\tau_j}(\boldsymbol{x}_{i,t})$ to long and the lowest $\hat{q}_{t,\tau_j}(\boldsymbol{x}_{i,t})$ to short. Finally, we sort stocks with pairs of quantiles. For $\tau_j \in \{0.5\%, \ldots, 49.5\%\}$, we select firms into the short (and long) portfolios when their τ_j (and $1-\tau_j$) quantiles are the lowest (and highest). For example, we short stocks with the lowest 0.05% quantiles and long stocks with the highest 99.5% quantiles). When we face ties in firms sorted by the above three variables, we prefer firms with higher empirical skewness estimated from the 100 quantiles $\hat{q}_{t,\tau_j}(\boldsymbol{x}_{i,t})$.

Figures 10 and 11 report the cumulative returns and annualized Sharpe ratios of the long-short portfolios sorted by the above three measures. For comparison purposes, we also report the same

information for long-short returns sorted by q-mean, median, and regression mean. When the sorting variable is conditional probabilities $\Pr[r_{i,t+1} \geq r_c|\boldsymbol{x}_{i,t}]$ and $\Pr[r_{i,t+1} \leq -r_c|\boldsymbol{x}_{i,t}]$ (as in the left panels), the horizontal axis represents the cutoff level r_c, which ranges from 0 to 10%, with an increment of 0.2%, i.e., we form 50 probability-based long-short portfolios. When the sorting variable is a single conditional quantile $\hat{q}_{\tau_j,t}(\boldsymbol{x}_{i,t})$ (as in the middle panels), the horizontal axis shows the 100 grid points $\tau_j \in \{0.5\%, \ldots, 99.5\%\}$. When the sorting variable is a pair of quantiles (as in the right panels), the horizontal axis represents the smaller grid points $\tau_j \in \{0.5\%, \ldots, 49.5\%\}$.

When we sort stocks using conditional probabilities, as the cutoff value r_c increases, the long and short portfolios will select stocks that are more likely to have extreme positive and negative returns. If our machine learning methods can accurately forecast conditional probabilities, the returns of their long-short portfolios should be an increasing function of r_c. The left panels of Figure 10 confirm this pattern for LightGBM and Lasso, whose cumulative long-short returns are monotonically increasing in r_c. Although the long-short returns of PCR and NN3 do not exhibit a clear monotonic relationship with r_c, the left panels of Figure 11 show that the Shape ratios of the four machine learning methods generally increase with r_c.

One interesting observation is that for portfolios sorted on single quantiles when τ_j is greater than 50%, the cumulative long-short returns and the Sharpe ratios start to decline (see the middle panel of Figure 10). This could be due to inaccurate predictions of quantiles for large τ_j's. As noted in Section 1, stock returns tend to be right-skewed, resulting in longer right tails. The sparse observations on the right tail make the predictions more volatile. Using pairs of quantiles partly resolves this drawback, which leads to more stable long-short returns and Sharpe ratios, as is shown in the right panels of Figure 11.

The last observation we make is specific to the neural network. Both the cumulative return and Sharpe ratio exhibit fluctuations around the regression mean. More often than not, the alternative sorting variables provide better returns and Sharpe ratios than regression means. However, it is hard to conclude that such superior performance is robust. One explanation is that due to the

overlong training time[11], we stop the algorithm far beyond its convergence. We still provide the results for completeness of discussion, even though it can provide only limited results.

In general, the above three sorting variables (conditional probabilities, single quantiles, and pairs of quantiles) lead to higher long-short returns and Sharpe ratios than regression mean, highlighting the incremental information contained in conditional distributions beyond conditional means. As a result, LightGBM generally has higher long-short returns and Sharpe ratios for the three sorting variables. Moreover, q-mean under LightGBM has the highest long-short returns and Sharpe ratios among all the sorting variables and machine learning methods we consider. We investigate possible explanations in the following section.

4.3.3 Advantages of Sorting Using q-Mean

One of the most intriguing findings from the above analysis is that q-mean under the LightGBM model leads to much higher cumulative returns and Sharpe ratios for long-short investing than all the other sorting variables under all four machine learning methods. Below we provide a more in-depth analysis of the advantages of q-mean as a sorting variable.

For simplicity, we consider long-short portfolios sorted by the average of *two* conditional quantiles obtained from the empirical prediction. Suppose we have $\tau_1 = 11.5\%$ and $\tau_2 = 88.5\%$, for each stock i, we can compute $\hat{q}_{t,\tau_1}(\boldsymbol{x}_{i,t})$ and $\hat{q}_{t,\tau_2}(\boldsymbol{x}_{i,t})$ and sort stocks into long and short portfolios by $\left[\hat{q}_{t,\tau_1}(\boldsymbol{x}_{i,t}) + \hat{q}_{t,\tau_2}(\boldsymbol{x}_{i,t})\right]/2$. In Figure 12, the main plot shows the distribution of $\hat{q}_{t,\tau_1}(\boldsymbol{x}_{i,t})$ (orange) and $\hat{q}_{t,\tau_2}(\boldsymbol{x}_{i,t})$ (gray), and the subplot on the upper right corner provides the distribution of $\left[\hat{q}_{t,\tau_1}(\boldsymbol{x}_{i,t}) + \hat{q}_{t,\tau_2}(\boldsymbol{x}_{i,t})\right]/2$ (purple).

Suppose we select firms with top 1% (we use 1% instead of 10% to highlight the distinct regions in Figure 12) of $\left[\hat{q}_{t,\tau_1}(\boldsymbol{x}_{i,t}) + \hat{q}_{t,\tau_2}(\boldsymbol{x}_{i,t})\right]/2$ to long and bottom 1% of $\left[\hat{q}_{t,\tau_1}(\boldsymbol{x}_{i,t}) + \hat{q}_{t,\tau_2}(\boldsymbol{x}_{i,t})\right]/2$ to short. In Figure 12, we first highlight the top and bottom 1% regions in the distribution of $\left[\hat{q}_{t,\tau_1}(\boldsymbol{x}_{i,t}) + \hat{q}_{t,\tau_2}(\boldsymbol{x}_{i,t})\right]/2$. Then we use two green arrows to show the regions in the distributions

[11] Longer than one month on an Nvidia H100 GPU for the 3,000 models.

of $\hat{q}_{t,\tau_1}(\boldsymbol{x}_{i,t})$ and $\hat{q}_{t,\tau_2}(\boldsymbol{x}_{i,t})$, from which the top 1% portfolio firms come. Similarly, we use two red arrows to indicate the regions in the distributions of $\hat{q}_{t,\tau_1}(\boldsymbol{x}_{i,t})$ and $\hat{q}_{t,\tau_2}(\boldsymbol{x}_{i,t})$, from which the bottom 1% portfolio firms come.

It is interesting to see that the top 1% firms typically have extremely high $\hat{q}_{t,\tau_2}(\boldsymbol{x}_{i,t})$ and middle-range $\hat{q}_{t,\tau_1}(\boldsymbol{x}_{i,t})$. Firms with high $\hat{q}_{t,\tau_2}(\boldsymbol{x}_{i,t})$ have the highest 88.5% quantiles among all the firms and therefore are more likely to achieve extreme positive returns. On the other hand, firms with middle-range $\hat{q}_{t,\tau_1}(\boldsymbol{x}_{i,t})$ do not have the lowest 11.5% quantiles and are not likely to have extreme negative returns. As a result, the top 1% firms in the long portfolio are more likely to have extreme positive returns but are unlikely to have extreme negative returns. These firms are good choices to long because they have high upside potential but limited downside risk.

Similarly, the bottom 1% firms typically have extremely low $\hat{q}_{t,\tau_1}(\boldsymbol{x}_{i,t})$ and low $\hat{q}_{t,\tau_2}(\boldsymbol{x}_{i,t})$. Firms with low $\hat{q}_{t,\tau_1}(\boldsymbol{x}_{i,t})$ have the lowest 11.5% quantiles among all the firms and therefore are more likely to achieve extreme negative returns. On the other hand, firms with low $\hat{q}_{t,\tau_2}(\boldsymbol{x}_{i,t})$ have the lowest 88.5% quantiles and are not likely to have extreme positive returns. As a result, the bottom 1% firms in the short portfolio are more likely to have extreme negative returns but are unlikely to have extreme positive returns. These firms are good choices to short because they have high downside risk but limited upside potential.

Therefore, averaging over $\hat{q}_{t,\tau_1}(\boldsymbol{x}_{i,t})$ and $\hat{q}_{t,\tau_2}(\boldsymbol{x}_{i,t})$ provides a convenient way to select stocks with high upside potential and limited downside risk to long as well as stocks with high downside risk and limited upside potential to short. The result will be even more robust if the averaging is done over the 100 quantiles, which can potentially diversify away the risks in the averaging of individual pairs of quantiles.

5 Conclusion

We extend the empirical asset pricing literature by providing a cross-sectional analysis of conditional stock returns through the lens of conditional quantiles and distributions. Compared to expected returns, conditional distributions contain much richer information for formulating investment decisions. We develop machine learning methods to forecast the conditional quantiles of stock returns in the cross section through quantile regression. Machine learning can capture potential nonlinear dependence of the conditional quantiles on a large number of return predictors. We adopt Bayesian optimization with Gaussian process to improve the efficiency of hyperparameter tuning in machine learning. Extensive simulation studies show that our methods can accurately forecast complicated data-generating processes' conditional quantiles and distributions. Empirical results from the U.S. data show that measures constructed from conditional distributions can identify stocks with extreme positive or negative returns and achieve superior performance in long-short investing.

References

Anatolyev, S. and Baruník, J. (2019). Forecasting dynamic return distributions based on ordered binary choice. International Journal of Forecasting, 35(3):823–835.

Bai, J. and Ng, S. (2005). Tests for skewness, kurtosis, and normality for time series data. Journal of Business & Economic Statistics, 23(1):49–60.

Baixauli, J. S. and Alvarez, S. (2004). Analysis of the conditional stock-return distribution under incomplete specification. European Journal of Operational Research, 155(2):276–283.

Belloni, A. and Chernozhukov, V. (2011). ℓ1-penalized quantile regression in high-dimensional sparse models. The Annals of Statistics, 39(1):82–130.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. Journal of the Royal Statistical Society. Series B, 57(1):289–300.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for hyper-parameter optimization. Advances in Neural Information Processing Systems, 24.

Berkowitz, J. (2001). Testing density forecasts, with applications to risk management. Journal of Business & Economic Statistics, 19(4):465–474.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. Journal of Econometrics, 31(3):307–327.

Bonferroni, C. (1936). Teoria statistica delle classi e calcolo delle probabilita. Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze, 8:3–62.

Callaway, B. and Huang, W. (2020). Distributional effects of a continuous treatment with an application on intergenerational mobility. Oxford Bulletin of Economics and Statistics, 82(4):808–842.

Chernozhukov, V., Fernández-Val, I., and Melly, B. (2013). Inference on counterfactual distributions. Econometrica, 81(6):2205–2268.

Chordia, T., Goyal, A., and Saretto, A. (2020). Anomalies and false rejections. The Review of Financial Studies, 33(5):2134–2179.

Diebold, F. X., Gunther, T. A., and Tay, A. S. (1998). Evaluating density forecasts with applications to financial risk management. International Economic Review, 39(4):863–883.

Egorov, A. V., Hong, Y., and Li, H. (2006). Validating forecasts of the joint probability density of bond yields: Can affine models beat random walk? Journal of Econometrics, 135(1):255–284.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. Econometrica, pages 987–1007.

Fama, E. F. and French, K. R. (1993). Common risk factors in the returns on stocks and bonds. Journal of Financial Economics, 33(1):3–56.

Fama, E. F. and French, K. R. (2015). A five-factor asset pricing model. Journal of Financial Economics, 116(1):1–22.

Foresi, S. and Peracchi, F. (1995). The conditional distribution of excess returns: An empirical analysis. Journal of the American Statistical Association, 90(430):451–466.

Fortin, N., Lemieux, T., and Firpo, S. (2011). Chapter 1 - decomposition methods in economics. volume 4 of Handbook of Labor Economics, pages 1–102. Elsevier.

Goyal, A. and Welch, I. (2008). A comprehensive look at the empirical performance of equity premium prediction. The Review of Financial Studies, 21(4):1455–1508.

Green, J., Hand, J. R., and Zhang, X. F. (2013). The supraview of return predictive signals. Review of Accounting Studies, 18(3):692–730.

Green, J., Hand, J. R. M., and Zhang, X. F. (2017). The characteristics that provide independent information about average U.S. monthly stock returns. The Review of Financial Studies, 30(12):4389–4436.

Gu, S., Kelly, B., and Xiu, D. (2020). Empirical asset pricing via machine learning. The Review of Financial Studies, 33(5):2223–2273.

Hamilton, J. D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. Econometrica, pages 357–384.

Harvey, C. R., Liu, Y., and Zhu, H. (2016). ... and the Cross-Section of Expected Returns. The Review of Financial Studies, 29(1):5–68.

Hohberg, M., Pütz, P., and Kneib, T. (2020). Treatment effects beyond the mean using distributional regression: Methods and guidance. PloS one, 15(2):e0226514.

Hong, Y. and Li, H. (2005). Nonparametric specification testing for continuous-time models with applications to term structure of interest rates. The Review of Financial Studies, 18(1):37–84.

Hong, Y., Li, H., and Zhao, F. (2004). Out-of-sample performance of discrete-time spot interest rate models. Journal of Business & Economic Statistics, 22(4):457–473.

Hong, Y., Li, H., and Zhao, F. (2007). Can the random walk model be beaten in out-of-sample density forecasts? evidence from intraday foreign exchange rates. Journal of Econometrics, 141(2):736–776.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 30:3146–3154.

Kelly, B. and Jiang, H. (2014). Tail risk and asset prices. The Review of Financial Studies, 27(10):2841–2871.

Koenker, R. and Bassett, G. (1978). Regression quantiles. Econometrica, 46(1):33–50.

Kon, S. J. (1984). Models of stock returns—a comparison. The Journal of Finance, 39(1):147–165.

Linden, M. (2001). A model for stock return distribution. International Journal of Finance & Economics, 6(2):159–169.

Machado, J. A. and Mata, J. (2005). Counterfactual decomposition of changes in wage distributions using quantile regression. Journal of Applied Econometrics, 20(4):445–465.

Melly, B. (2005). Decomposition of differences in distribution using quantile regression. Labour Economics, 12(4):577–590.

Močkus, J. (1975). On bayesian methods for seeking the extremum. In Optimization Techniques IFIP Technical Conference, pages 400–404.

Nelson, D. B. (1991). Conditional heteroskedasticity in asset returns: A new approach. Econometrica, pages 347–370.

Peiro, A. (1994). The distribution of stock returns: International evidence. Applied Financial Economics, 4(6):431–439.

Rosenblatt, M. (1952). Remarks on a multivariate transformation. The Annals of Mathematical Statistics, 23(3):470–472.

Ross, S. A. (1976). The arbitrage theory of capital asset pricing. Journal of Economic Theory, 13(3):341–360.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. (2015). Taking the human out of the loop: A review of bayesian optimization. Proceedings of the IEEE, 104(1):148–175.

Sharpe, W. F. (1963). A simplified model for portfolio analysis. Management Science, 9(2):277–293.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. Advances in Neural Information Processing Systems, 25.

Storey, J. D. (2003). The positive false discovery rate: a bayesian interpretation and the q-value. The Annals of Statistics, 31(6):2013–2035.

Turner, R., Eriksson, D., McCourt, M., Kiili, J., Laaksonen, E., Xu, Z., and Guyon, I. (2021). Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In NeurIPS 2020 Competition and Demonstration Track, pages 3–26.

Zhao, Y. (2013). Forecasting the stock return distribution using macro-finance variables. Job Market Paper.

Figures

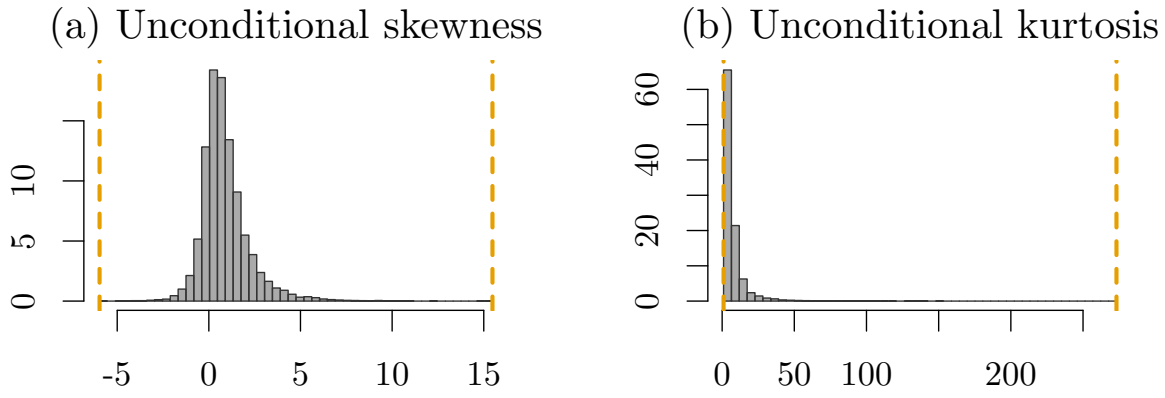

Figure 1: **Skewness and Kurtosis of Individual Stock Returns**

This figure reports the histograms of the skewness and kurtosis of individual stock returns of the 17,947 firms considered in our sample between January 1978 and December 2016. The vertical axes are in percentage. The two vertical dotted lines in each panel indicate the minimum and maximum values.

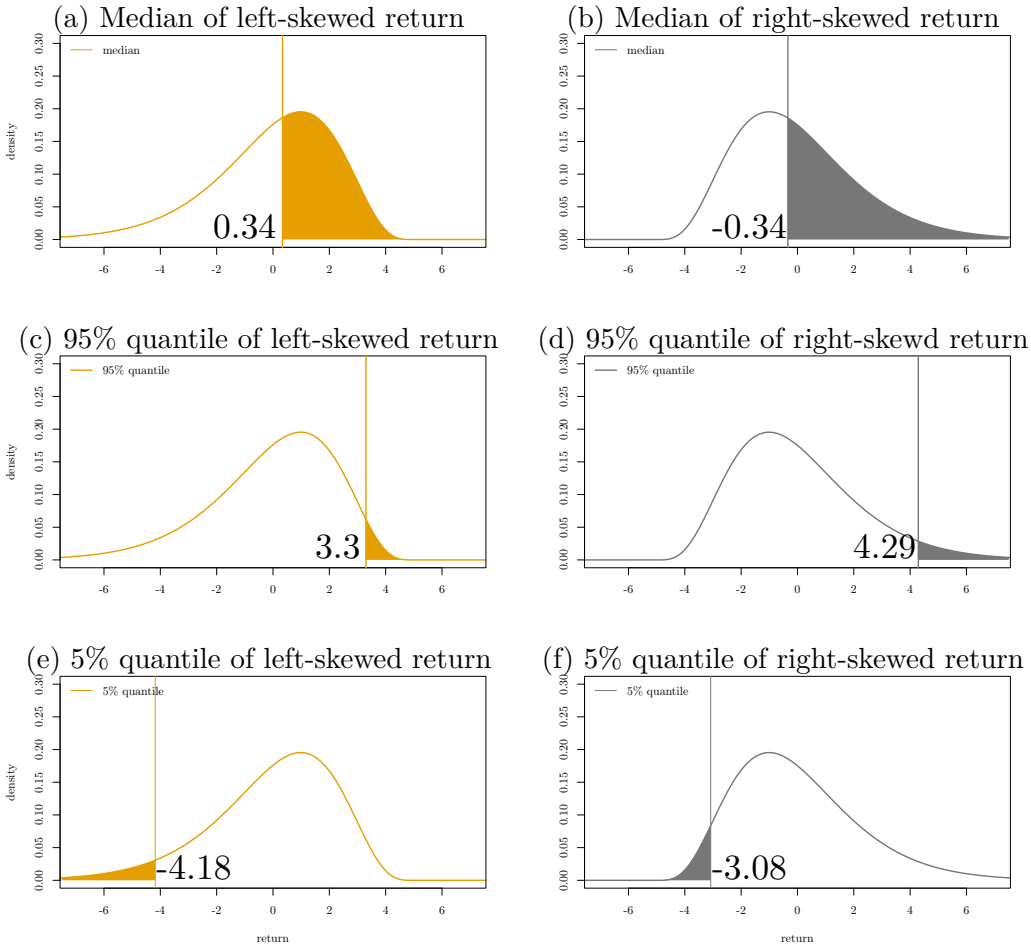

Figure 2: **Left- and Right-skewed Stock Return Distributions**

This figure provides a comparison of left- and right-skewed return distributions. The left column contains the probability density of the left-skewed distribution, which follows $-Gamma(5,1)+5$. The right column contains the probability density of the right-skewed distribution, which follows $Gamma(5,1)-5$. The vertical lines in the top, middle, and bottom subplots are the medians, the 95% quantiles, and the 5% quantiles calculated from Monte Carlo samples, respectively.

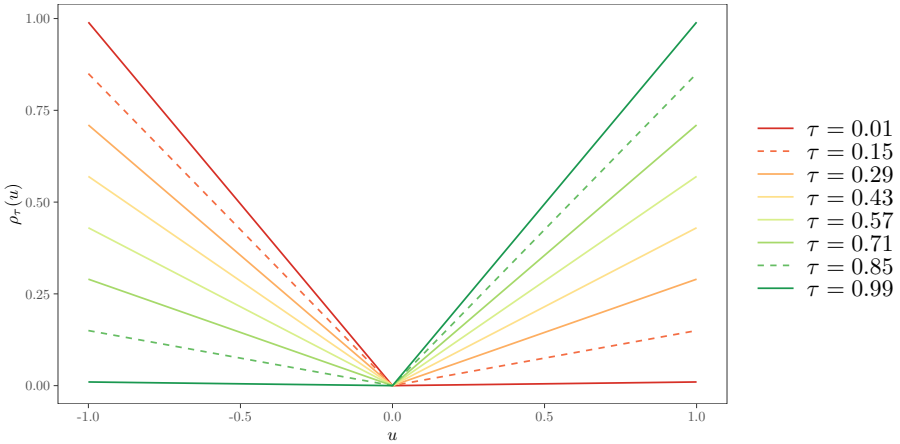

Figure 3: **Check Loss Functions for Quantile Regression**

The figure reports the check loss functions $\rho_\tau(u) = u(\tau - \mathbb{1}(u \leq 0))$ used in quantile regression for eight different values of τ equally spaced between 0.01 and 0.99.

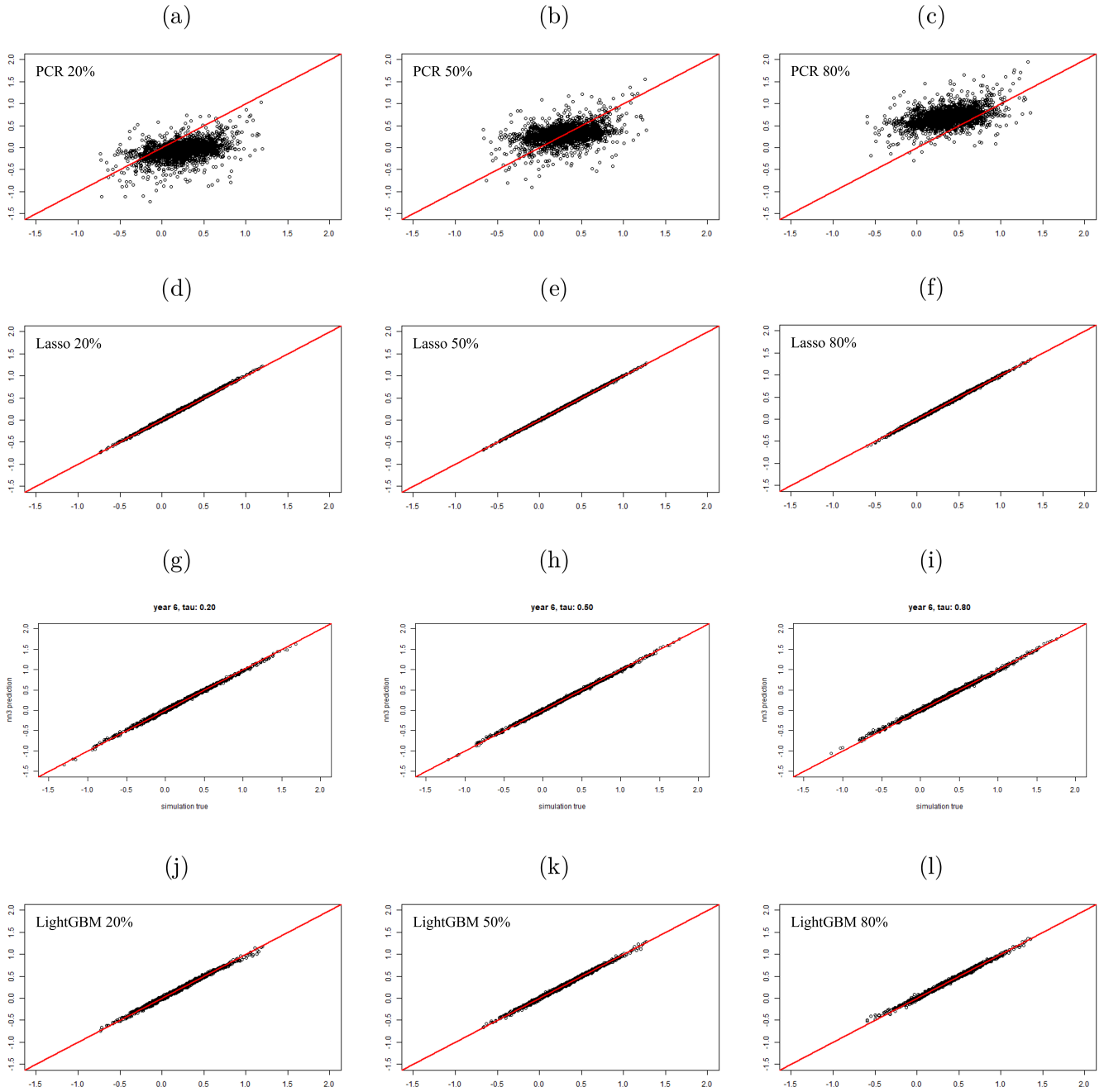

Figure 4: **True and Predicted Conditional Quanitles for Linear Data-Generating Process**

This figure provides a comparison of the true quantiles of the linear data-genearting process and the predicted quantiles from the three machine learning models. The true model has the following conditional mean function that is linear in the predictors (22a):

$$g^*(\boldsymbol{x}_{it}) = 0.3 + 0.3x_{it1} + 0.2x_{it2} + w_t x_{it3}.$$

From top to bottom, each row represents the results for PCR, Lasso, and LightGBM, respectively. From left to right, each row represents the results for the 20%, 50%, and 80% quantile, respectively. In each subplot, the horizontal axis represents the true quantiles, the vertical axis represents the predicted quantiles, and the red line represents the 45%-degree line passing through the origin. Each dot shows a firm-time specific observation.

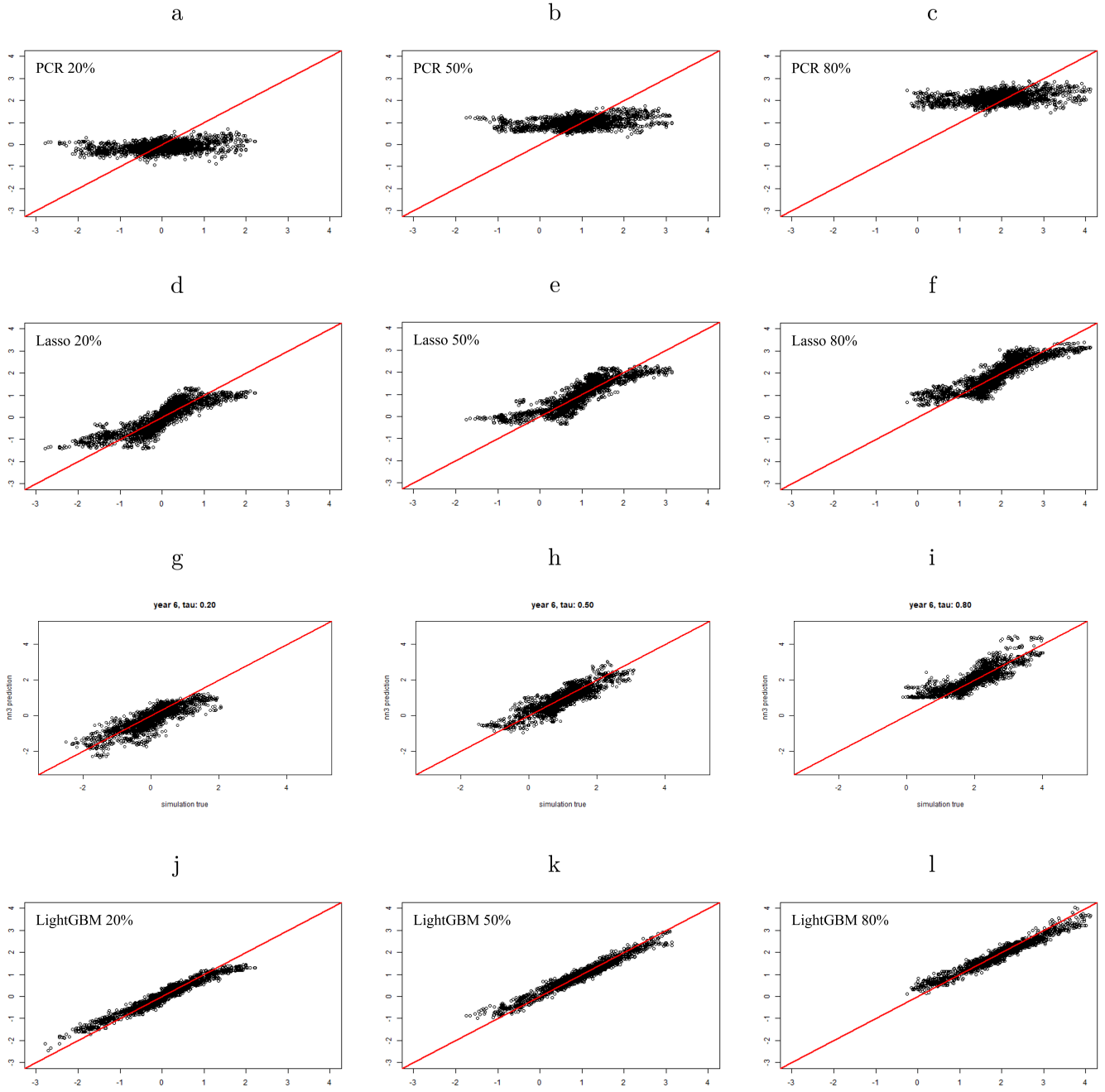

Figure 5: **True and Predicted Conditional Quanitles for Nonlinear Data-Generating Process**

This figure provides a comparison of the true quantiles of the nonlinear data-genearting process and the predicted quantiles from the three machine learning models. The true model has the following conditional mean function that is nonlinear in the predictors in model (22b):

$$g^*(\boldsymbol{x}_{it}) = 1 + 0.4x_{it1} + 2x_{it1}x_{it2}^2.$$

From top to bottom, each row represents the results for PCR, Lasso, and LightGBM, respectively. From left to right, each row represents the results for the 20%, 50%, and 80% quantile, respectively. In each subplot, the horizontal axis represents the true quantiles, the vertical axis represents the predicted quantiles, and the red line represents the 45%-degree line passing through the origin. Each dot shows a firm-time specific observation.

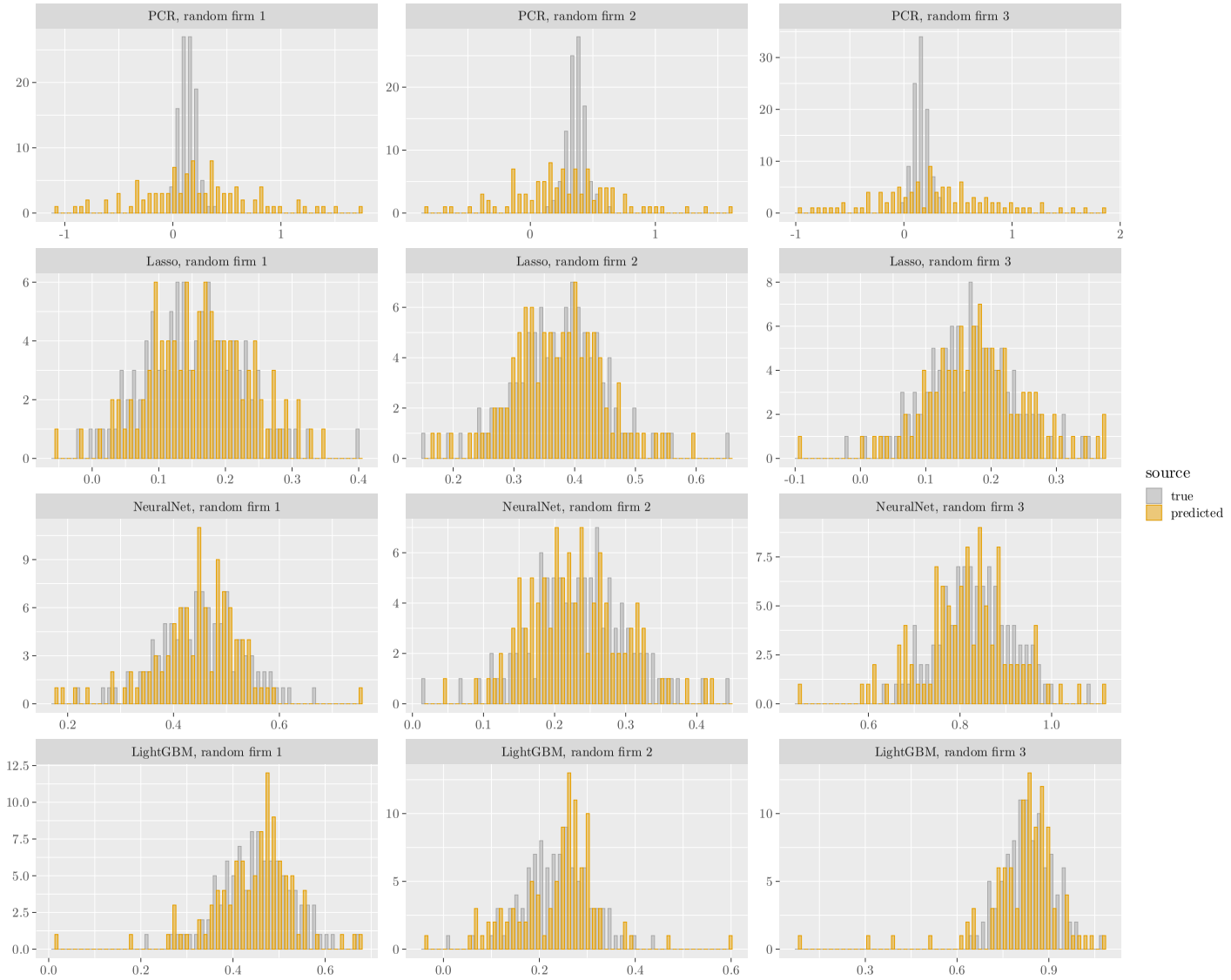

Figure 6: **True and Predicted Conditional Distributions for Linear Data-Generating Process**

This figure provides a comparison of the true distributions of the linear data-genearting process and the predicted distributions from the three machine learning models. The true model has the following conditional mean function that is linear in the predictors in model (22a):

$$g^*(\boldsymbol{x}_{it}) = 0.3 + 0.3x_{it1} + 0.2x_{it2} + w_t x_{it3}.$$

From top to bottom, each row represents the results for PCR, Lasso, and LightGBM, respectively, for three randomly selected firms in our sample. The yellow histograms represent the true densities, while the gray histograms represent the predicted densities.

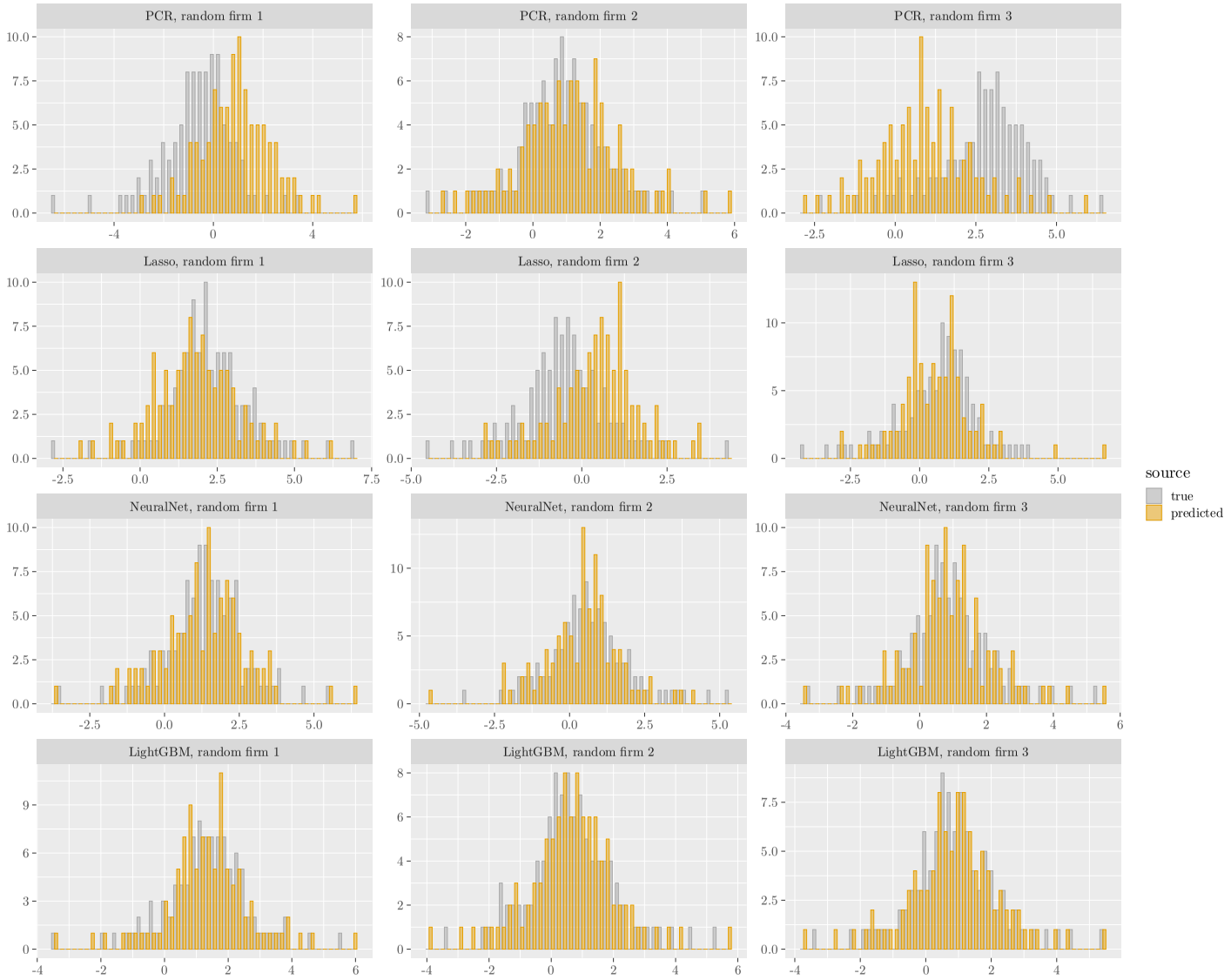

Figure 7: **True and Predicted Conditional Distributions for Nonlinear Data-Generating Process**

This figure provides a comparison of the true distributions of the nonlinear data-genearting process and the predicted distributions from the three machine learning models. The true model has the following conditional mean function that is nonlinear in the predictors in model (22b):

$$g^*(\boldsymbol{x}_{it}) = 1 + 0.4x_{it1} + 2x_{it1}x_{it2}^2.$$

From top to bottom, each row represents the results for PCR, Lasso, and LightGBM, respectively, for three randomly selected firms in our sample. The yellow histograms represent the true densities, while the gray histograms represent the predicted densities.

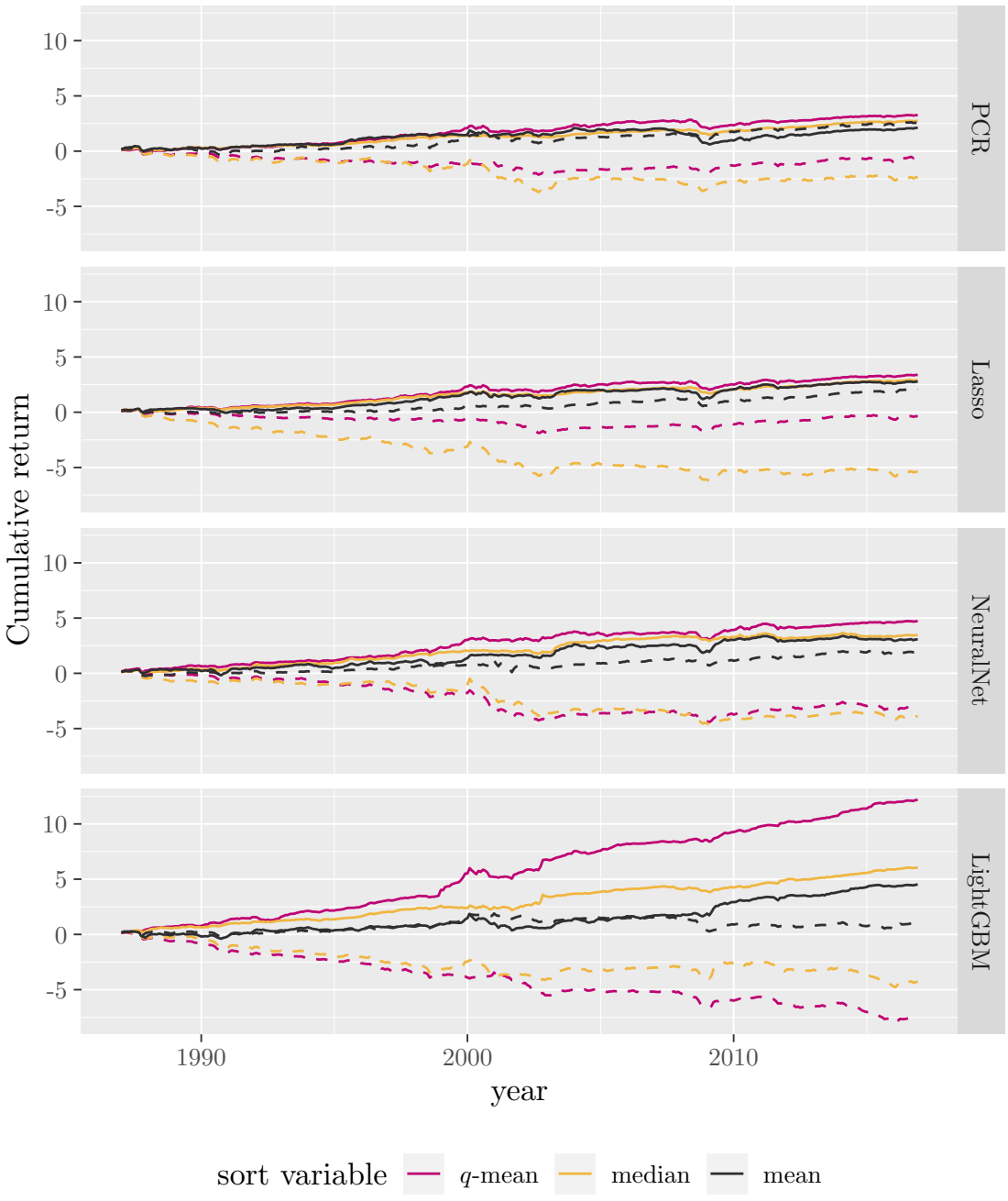

Figure 8: **Cumulative Returns of Long and Short Portfolios**

The figure provides time series plots of the cumulative returns of the long and short portfolios sorted by q-mean, median, and regression mean under PCR, Lasso, and LightGBM, from top to bottom, respectively. The solid curves represent the cumulative returns of the long portfolios, while the dashed curves represent that of the short portfolios. In each subplot, the violet, orange, and black curves represent the portfolios sorted by q-mean, median, and regression mean, respectively.

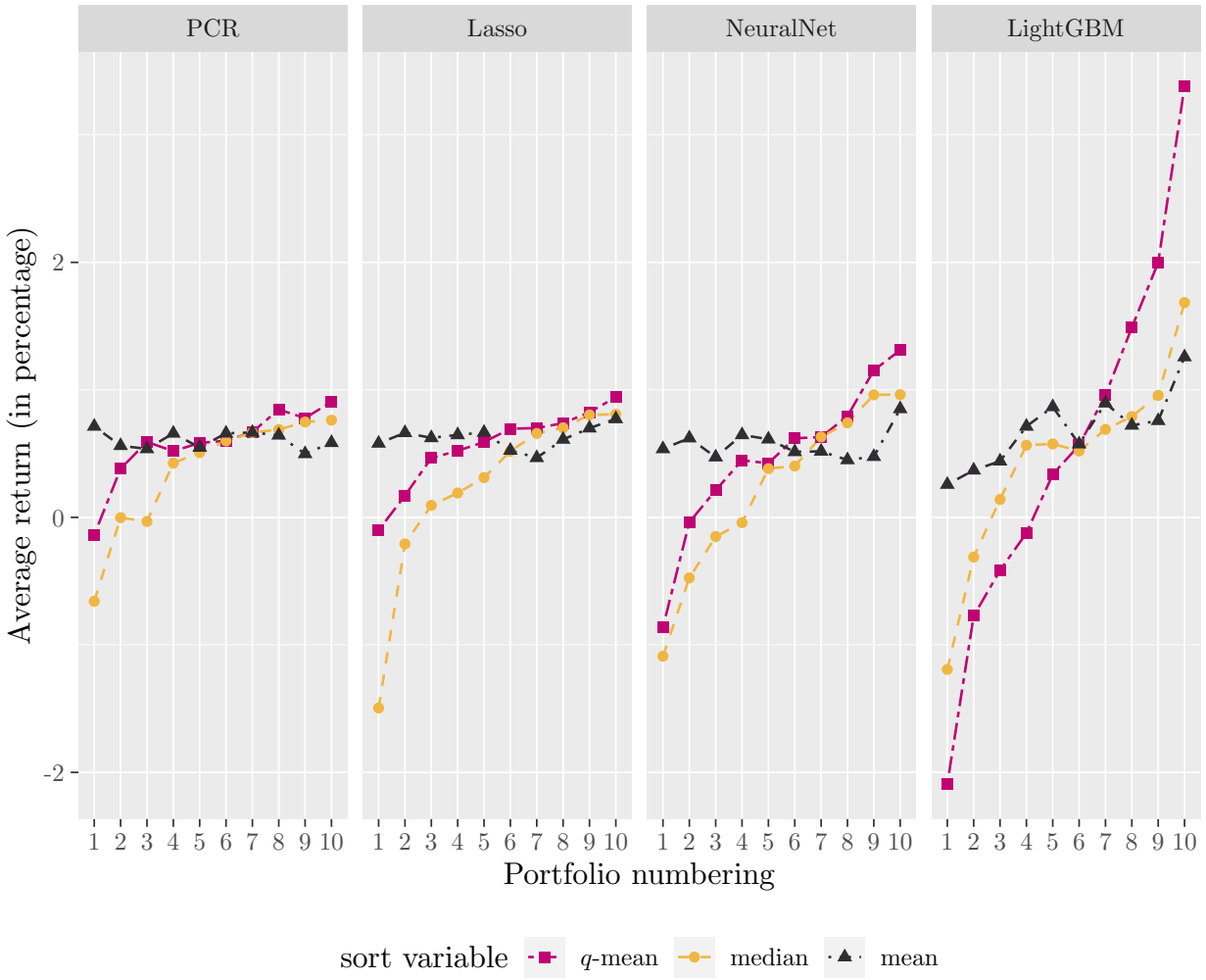

Figure 9: **Average Monthly Returns of Decile Portfolios**

The figure provides time-series averages of monthly returns of decile portfolios sorted by q-mean, median, and regression mean under PCR, Lasso, and LightGBM, from left to right, respectively. In each subplot, the horizontal axes represent the rank of the decile portfolios from 1 (short) to 10 (long); the common vertical axis represents the predicted average monthly returns; and the violet, orange, and black curves represent the average monthly returns for decile portfolios sorted by q-mean, median, and regression mean, respectively.

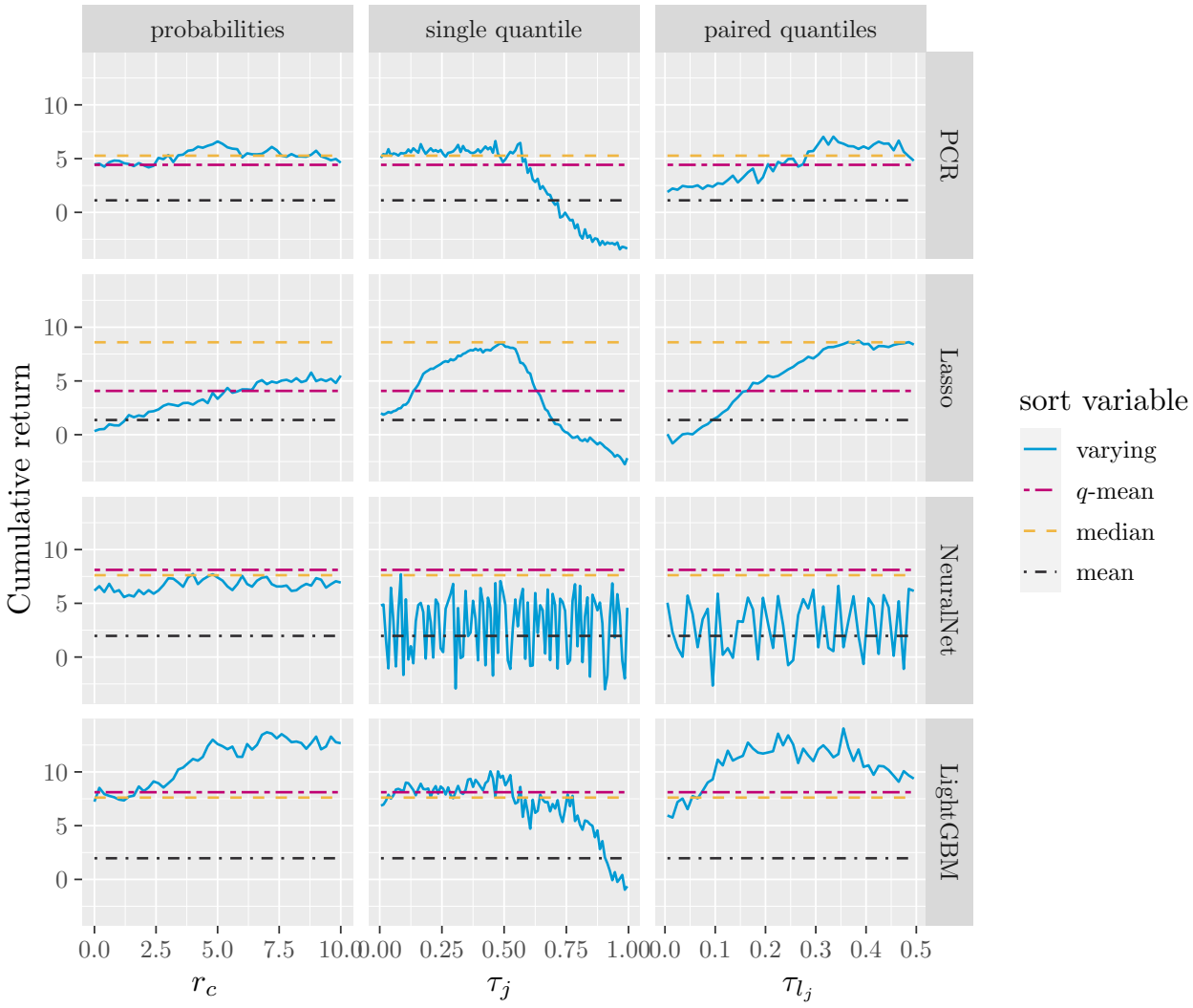

Figure 10: **Cumulative Returns of Long-Short Portfolios Sorted by Conditional Probabilities and Conditional Quantiles**

This figure reports cumulative returns of long-short portfolios sorted by conditional probabilities and conditional quantiles for PCR, Lasso, and LightGBM, from top to bottom, respectively. In the left column, for a given r_c, we select stocks with the highest $\Pr[r_{i,t+1} \geq r_c|\boldsymbol{x}_{i,t}]$ to long, and stocks with the highest $\Pr[r_{i,t+1} \leq -r_c|\boldsymbol{x}_{i,t}]$ to short; in the middle column, for a given τ_j, we choose stocks with the highest $\hat{q}_{\tau_j,t}(\boldsymbol{x}_{i,t})$ to long and the lowest $\hat{q}_{\tau_j,t}(\boldsymbol{x}_{i,t})$ to short; and in the right column, for $\tau_{l_j} \in \{0.5\%, \ldots, 49.5\%\}$, we select firms into the short (and long) portfolios when their τ_{l_j} (and 1-τ_j) quantiles are the lowest (and highest). For comparison, we also report cumulative long-short returns for portfolios sorted by q-mean, median, and regression mean, represented by the violet, orange, and black lines, respectively.

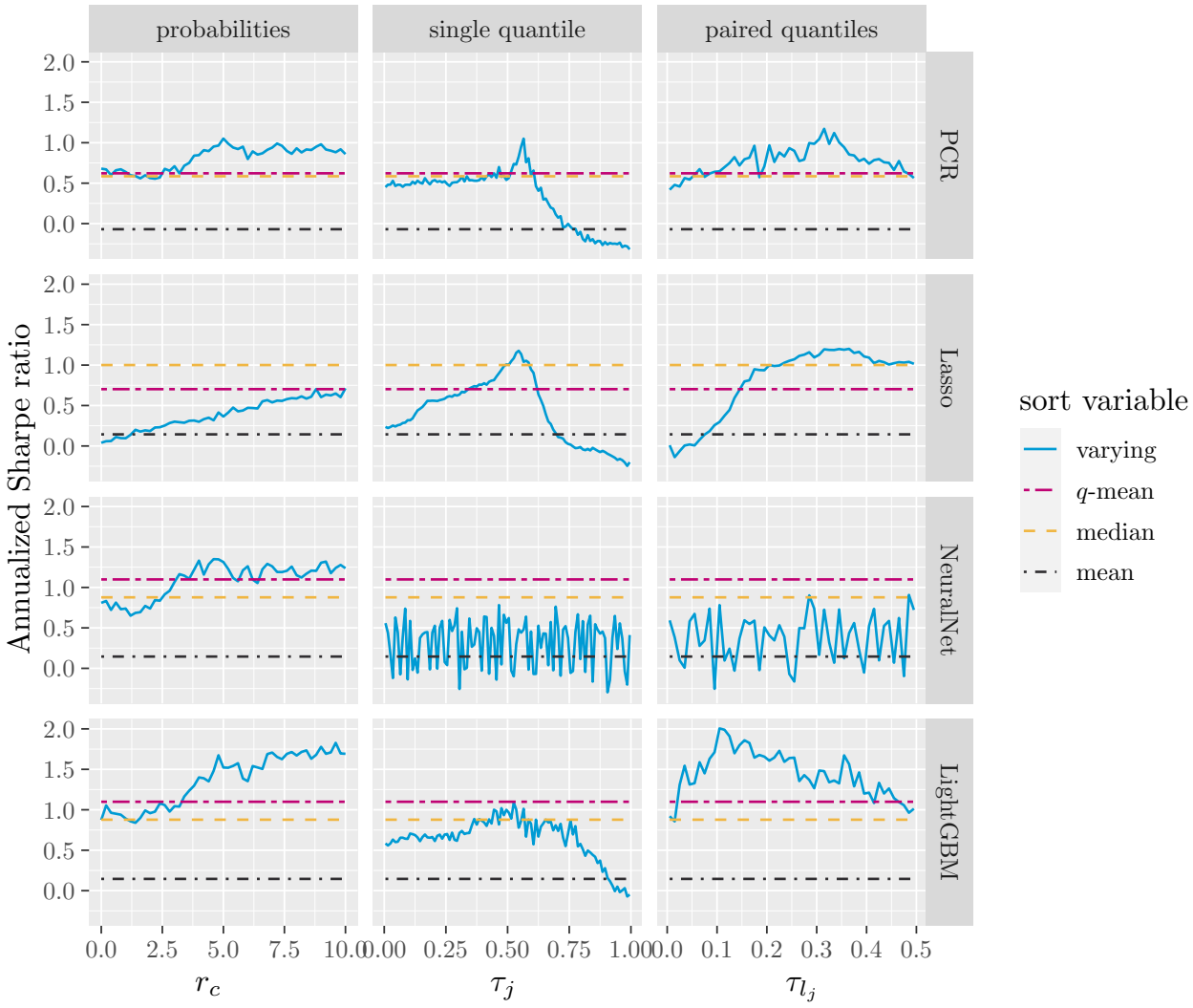

Figure 11: **Sharpe Ratios of Long-Short Portfolios Sorted by Conditional Probabilities and Conditional Quantiles**

This figure reports annualized Sharpe ratios of long-short portfolios sorted by conditional probabilities and conditional quantiles for PCR, Lasso, and LightGBM, from top to bottom, respectively. In the left column, for a given r_c, we select stocks with the highest $\Pr[r_{i,t+1} \geq r_c|\boldsymbol{x}_{i,t}]$ to long, and stocks with the highest $\Pr[r_{i,t+1} \leq -r_c|\boldsymbol{x}_{i,t}]$ to short; in the middle column, for a given τ_j, we choose stocks with the highest $\hat{q}_{\tau_j,t}(\boldsymbol{x}_{i,t})$ to long and the lowest $\hat{q}_{\tau_j,t}(\boldsymbol{x}_{i,t})$ to short; and in the right column, for $\tau_{l_j} \in \{0.5\%, \ldots, 49.5\%\}$, we select firms into the short (and long) portfolios when their τ_{l_j} (and 1-τ_j) quantiles are the lowest (and highest). For comparison, we also report the Sharpe ratios of long-short portfolios sorted by q-mean, median, and regression mean, represented by the violet, orange, and black lines, respectively.

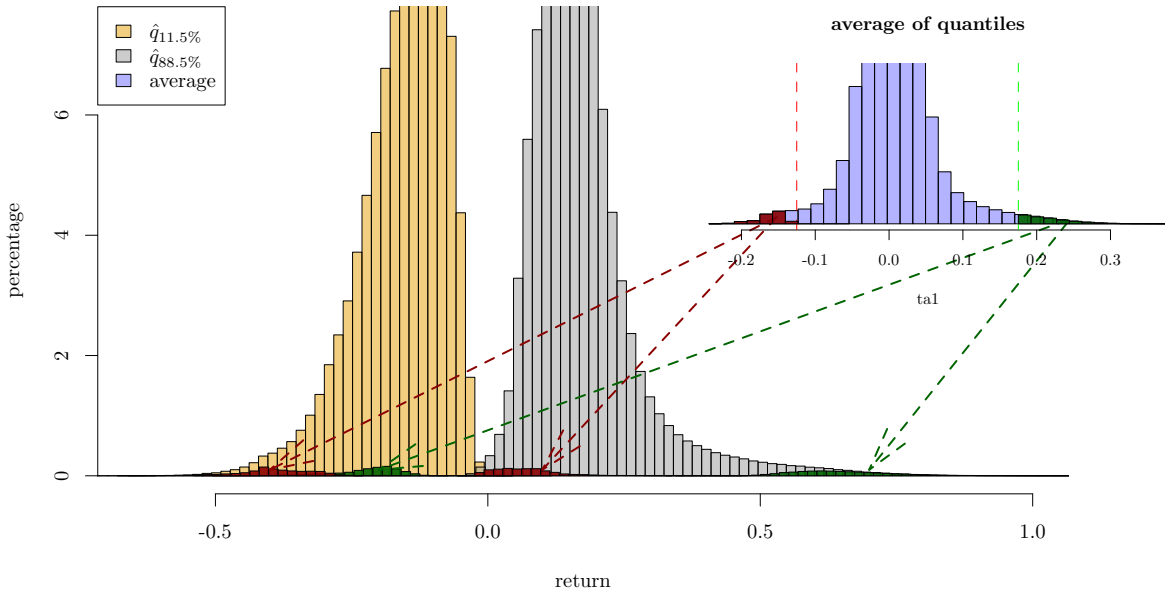

Figure 12: **Advantages of q-Mean for Long-Short Investing**

The figure provides an illustration of the advantages of q-mean as a sorting variable for long-short investing. Given $\tau_1 = 11.5\%$ and $\tau_2 = 88.5\%$, for each stock i, we obtain $\hat{q}_{\tau_1,t}(\boldsymbol{x}_{i,t})$ and $\hat{q}_{\tau_2,t}(\boldsymbol{x}_{i,t})$. We sort stocks using $[\hat{q}_{\tau_1,t}(\boldsymbol{x}_{i,t}) + \hat{q}_{\tau_2,t}(\boldsymbol{x}_{i,t})]/2$ and choose the top (bottom) 1% of firms to long (short). The histograms of $\hat{q}_{\tau_1,t}(\boldsymbol{x}_{i,t})$, $\hat{q}_{\tau_2,t}(\boldsymbol{x}_{i,t})$, and $[\hat{q}_{\tau_1,t}(\boldsymbol{x}_{i,t}) + \hat{q}_{\tau_2,t}(\boldsymbol{x}_{i,t})]/2$ are colored as orange, gray, and purple, respectively. The green (red) region of the purple distribution represent the top (bottom) 1% of the firms sorted by $[\hat{q}_{\tau_1,t}(\boldsymbol{x}_{i,t}) + \hat{q}_{\tau_2,t}(\boldsymbol{x}_{i,t})]/2$. The green regions in the orange and gray distributions suggest that the top 1% firms have high $\hat{q}_{\tau_2,t}(\boldsymbol{x_{i,t}})$ and middle-level $\hat{q}_{\tau_1,t}(\boldsymbol{x_{i,t}})$. These firms, which are likely to have extreme positive returns but unlikely to have negative returns, are good candidates to long. The red regions in the orange and gray distributions suggest that the bottom 1% firms have low $\hat{q}_{\tau_1,t}(\boldsymbol{x}_{i,t})$ and low $\hat{q}_{\tau_2,t}(\boldsymbol{x}_{i,t})$. These firms, which are likely to have extreme negatives returns but unlikely to have positive returns, are good candidates to short. Therefore, q-mean can identify stocks with high upside potential but limited downside risk to long and stocks with high downside risk but limited upside potential to short.

Tables

Table 1: **Decomposition of Variations of Simulated Returns**

This table provides a decomposition analysis of the variations of the simulated returns from model (22a): $g^*(\boldsymbol{x}_{it}) = 0.3 + 0.3x_{it1} + 0.2x_{it2} + w_t x_{it3}$ (left panel) and model (22a): $g^*(\boldsymbol{x}_{it}) = 1 + 0.4x_{it1} + 2x_{it1}x_{it2}^2$ (right panel) Variation is measured in both standard deviation (as in the column named "std") and variance (as in the column named "variance"). From top to bottom, the four rows represent the conditional mean function, the time effect, the residual noise, and the simulated returns, respectively.

	Model (22a)			Model (22b)		
	mean	std	variance	mean	std	variance
$g^\star$	0.30	0.51	0.26	0.99	0.72	0.52
$\boldsymbol{\beta}^\intercal \boldsymbol{v}$	0.00	0.05	0.00	0.00	0.20	0.04
ϵ	-0.00	0.06	0.00	0.00	1.45	2.10
$r_{i,t+1}$	0.30	0.52	0.27	0.99	1.63	2.65

Table 2: **Prediction Accuracy of Realized Returns**

This table compares the accuracy of regression mean, median, and quantile mean in predicting future realized returns of the 17,947 firms listed on NYSE, AMEX, and NASDAQ from 1987 to 2016 under PCR, Lasso, and LightGBM, from top to bottom, respectively. Rows with $|r - \hat{r}^{QM}|$ provide summary information of the absolute prediction errors of quantile mean. Rows with $|r - \hat{r}^{0.5}|$ provide summary information of the absolute prediction errors of median. Rows with $|r - \hat{r}^{reg}|$ provide summary information of the absolute prediction errors of regression mean.

ML	source	min	Q1	median	Q3	max	mean
PCR	$\lvert r - \hat{r}^{QM}\rvert$	0.00	0.03	0.08	0.16	18.99	0.12
	$\lvert r - \hat{r}^{0.5}\rvert$	0.00	0.03	0.08	0.15	19.01	0.12
	$\lvert r - \hat{r}^{reg}\rvert$	0.00	0.03	0.08	0.16	19.00	0.12
Lasso	$\lvert r - \hat{r}^{QM}\rvert$	0.00	0.04	0.08	0.16	18.98	0.12
	$\lvert r - \hat{r}^{0.5}\rvert$	0.00	0.04	0.08	0.15	19.01	0.12
	$\lvert r - \hat{r}^{reg}\rvert$	0.00	0.03	0.08	0.16	18.99	0.12
NeuralNet	$\lvert r - \hat{r}^{QM}\rvert$	0.00	0.03	0.08	0.16	19.00	0.13
	$\lvert r - \hat{r}^{0.5}\rvert$	0.00	0.03	0.08	0.16	19.02	0.13
	$\lvert r - \hat{r}^{reg}\rvert$	0.00	0.03	0.08	0.16	19.02	0.12
LightGBM	$\lvert r - \hat{r}^{QM}\rvert$	0.00	0.03	0.08	0.15	18.85	0.12
	$\lvert r - \hat{r}^{0.5}\rvert$	0.00	0.03	0.08	0.15	19.00	0.12
	$\lvert r - \hat{r}^{reg}\rvert$	0.00	0.03	0.08	0.16	18.99	0.12

Table 3: **Evaluation of Conditional Density Forecasts**

This table reports the evaluations of the conditional density forecasts using the Kolmogorov-Smirnov (KS) test, the Shapiro normality test, and the Berkowitz (2001) likelihood ratio (LR) test for PCR, Lasso, and LightGBM, from top to bottom, respectively. The last column, denoted as $P(> 0.05)$, reports the proportion of firm-time observations that pass the corresponding test at the 5% significance level, i.e. $p > 0.05$

ML	test	min	Q1	median	Q3	max	mean	$P(> 0.05)$
PCR	KS	0.00	0.07	0.23	0.52	1.00	0.31	0.79
	Shapiro	0.00	0.07	0.25	0.52	1.00	0.32	0.80
	LR	0.00	0.03	0.20	0.53	1.00	0.30	0.70
Lasso	KS	0.00	0.03	0.17	0.46	1.00	0.27	0.70
	Shapiro	0.00	0.07	0.25	0.53	1.00	0.33	0.79
	LR	0.00	0.01	0.13	0.43	1.00	0.25	0.62
NeuralNet	KS	0.00	0.06	0.21	0.48	1.00	0.29	0.76
	Shapiro	0.00	0.08	0.25	0.50	1.00	0.32	0.80
	LR	0.00	0.04	0.23	0.55	1.00	0.32	0.70
LightGBM	KS	0.00	0.08	0.26	0.53	1.00	0.33	0.81
	Shapiro	0.00	0.11	0.30	0.57	1.00	0.36	0.86
	LR	0.00	0.08	0.30	0.61	1.00	0.36	0.80

Table 4: **Annualized Sharpe Ratios of Long-Short Portfolios**

This table reports the annualized Sharpe ratios of the long-short portfolios sorted by q-mean, median, and regression mean under PCR, Lasso, and LightGBM, from top to bottom, respectively.

	q-mean	median	mean
PCR	0.62	0.59	-0.07
Lasso	1.00	0.70	0.15
NeuralNet	1.11	0.88	0.23
LightGBM	1.76	1.09	0.47

Table 5: **Alpha against benchmark models**

This table reports alpha against Fama and French (1993) and Fama and French (2015). The rows labeled FF3 is alpha against Fama and French (1993) and FF5 Fama and French (2015).

	Benchmark	Median	q-mean	Mean
PCR	FF3	1.78	1.26	-0.20
	FF5	0.92	1.07	-0.40
Lasso	FF3	2.64	1.13	0.24
	FF5	1.93	0.98	0.43
NeuralNet	FF3	0.21	2.20	0.20
	FF5	1.31	1.60	0.53
LightGBM	FF3	3.38	6.00	1.11
	FF5	3.10	6.31	1.25

Appendices

A Approximation to Expectation with Fixed τ's

We prove the validity of approximating expectations with the average of functions of quantiles obtained from a fixed grid. For a random variable X with *cdf* F (marginal or conditional) on the probability space $(\Omega, \mathcal{F}, P)$, where P is a probability measure, its induced measure on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ is $P_X(B) = P(X^{-1}(B))$ for any measurable $B \in \mathcal{B}(\mathbb{R})$, the Borel sigma-algebra on $\mathbb{R}$.

The expectation of X is defined as

$$\mathbb{E}[X] \coloneqq \int_{\mathbb{R}} x dP_X = \int_{\Omega} X(\omega) dP. \tag{A.1}$$

Apply a change of variable $\tau = F(x)$ and define $F^{-1}(\tau) = \inf\{x : \tau \leq F(x)\}$. We have

$$\begin{aligned} &\int_{\mathbb{R}} x dP_X \\ =&\int_{[0,1]} F^{-1}(\tau) dP_X F^{-1} \\ =&\int_{[0,1]} F^{-1}(\tau) d\tau, \end{aligned} \tag{A.2}$$

where the last equality holds if $P_X F^{-1}(\cdot) \equiv P_X\left(F^{-1}(\cdot)\right)$ is the Lebesgue measure on $[0,1]$.

To show this, pick $0 \leq a < b \leq 1$,

$$\begin{aligned} &P_X F^{-1}([a,b]) \\ =&P_X\left(\left[F^{-1}(a), F^{-1}(b)\right]\right) \\ =&P\left(F^{-1}(a) \leq X \leq F^{-1}(b)\right) \\ =&P\left(X \leq F^{-1}(b)\right) - P\left(X \leq F^{-1}(a)\right) \\ =&b - a, \end{aligned} \tag{A.3}$$

where the first equality is by the non-decreasing monotonicity of $F(\cdot)$ and the last equality holds by the definition of F^{-1}. Note that $F^{-1}(\cdot) \equiv q(\cdot)$ is exactly the quantile function (Its conditional counterpart for a given $\boldsymbol{x}_{i,t}$ is $F^{-1}(\tau|\boldsymbol{x}_{i,t}) \equiv q(\tau|\boldsymbol{x}_{i,t})$, which is denoted as $q_\tau(\boldsymbol{x}_{i,t})$ in the main text. Here we use τ as the argument to highlight its central role in this proof).

For any measurable function $g(\cdot)$, the expectation is thus

$$\mathbb{E}\left[g(X)\right] = \int_{\mathbb{R}} g(x) dP_X = \int_{[0,1]} g(F^{-1}(\tau)) dP_X F^{-1} = \int_{[0,1]} g(F^{-1}(\tau)) d\tau. \tag{A.4}$$

The conditional expectations can be derived in a similar manner.

For a partition of the unit interval $0 = \tau_0 < \tau_1 < \tau_2 < \cdots < \tau_J = 1$, the integral in equation (A.4) can be approximated by the Riemann sum
(when $\max\{\tau_1 - \tau_0, \tau_2 - \tau_1, \ldots, \tau_J - \tau_{J-1}\} \to 0$)

$$\sum_{j=1}^{J} g(q(\tau_j^*))[\tau_j - \tau_{j-1}], \tag{A.5}$$

where $\tau_j^* \in [\tau_{j-1}, \tau_j]$. In our implementation, we partition the unit interval into 100 sub-intervals of length 0.01 and choose the middle point as τ_j^*. In such cases,

$$\begin{aligned} &\sum_{j=1}^{J} g(q(\tau_j^*))[\tau_j - \tau_{j-1}] \\ =&\frac{1}{J}\sum_{j=1}^{J} g(q(\tau_j^*)) \\ =&\frac{1}{100}\sum_{j=1}^{100} g(q(\tau_j)). \end{aligned} \tag{A.6}$$

Specifically, for $g(r_{i,t+1}) = \mathbb{1}(r_{i,t+1} \geq r_c)$, we have $\mathbb{E}\left[g(r_{i,t+1})|\boldsymbol{x}_{i,t}\right] = P(r_{i,t+1} \geq r_c|\boldsymbol{x}_{i,t})$ as in equation (10). Similarly, for $g(r_{i,t+1}) = r_{i,t+1}$, we have $\mathbb{E}\left[g(r_{i,t+1})|\boldsymbol{x}_{i,t}\right] = \mathbb{E}\left[r_{i,t+1}|\boldsymbol{x}_{i,t}\right]$ as in equation (11).

B Optimal Hyperparameters

We summarise the tuned hyperparameters for the three machine learning methods. Figure B.1 shows the hyperparameter (the optimal number of principal components m) of PCR. The horizontal axis shows the rank of quantiles, ranging from 0.5% to 99.5%. The vertical axis shows the chosen optimal number of principal components. The solid curve shows the average of the 30 optimal hyperparameters for the annual quantile regression models. The band is one time-series standard deviation of the 30 annual optimal hyperparameters for each τ away from the corresponding average. The solid curve shows a smile shape: first decreasing with τ and then increasing, showing that it is easier to extract information from for the center of the conditional distributions than for the tails where data are more sparse.

Figure B.2 shows the distribution of the 30 penalty coefficients λ used in the Lasso quantile regression model. Following Belloni and Chernozhukov (2011), in each year, we compute one λ and use it to train the model for all quantiles.

Figure B.3 shows the hyperparameters of LightGBM. From top to bottom, the pictures show the learning rate, maximum depth of a tree, and the minimum number of observations on a leaf. In each plot, the horizontal axis shows the rank of the quantiles, τ. The vertical axis shows the value of the corresponding hyperparameters. The solid curve shows the time series average for each quantile model over the 30 years. The band indicates the fluctuation of one time-series standard deviation.

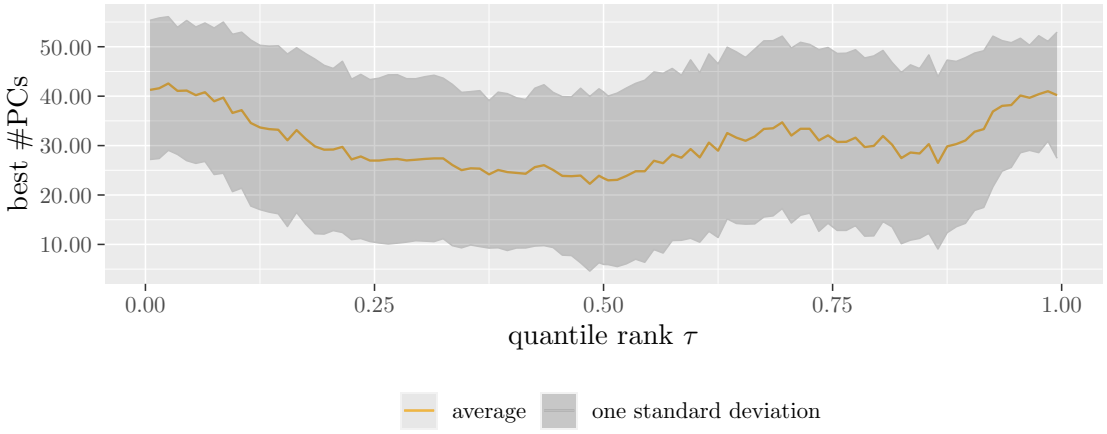

Figure B.1: **Distributions of m for PCR for All Quantile Models**

The figure provides the distributions of the hyperparameters, the optimal number of principal components (m) of PCR. The horizontal axis shows the indicator of the 100 quantiles, ranging form 0.5% to 99.5%. The vertical axis shows the optimal m. The yellow curve shows the time series average of the optimal hyperparameters over the 30 years. The band shows the corresponding plus (minus) the time-series standard deviation from the average.

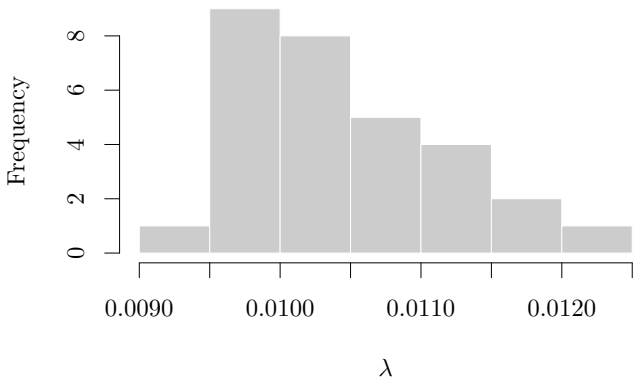

Figure B.2: **Distribution of λ for Lasso Models from 1987 to 2016**

The figure provides the histogram of the 30 λ's used to train the Lasso quantile regression. We adopt the suggested in-sample optimal value in Belloni and Chernozhukov (2011) instead of tuning directly for computational efficiency. The vertical axis shows the frequency.

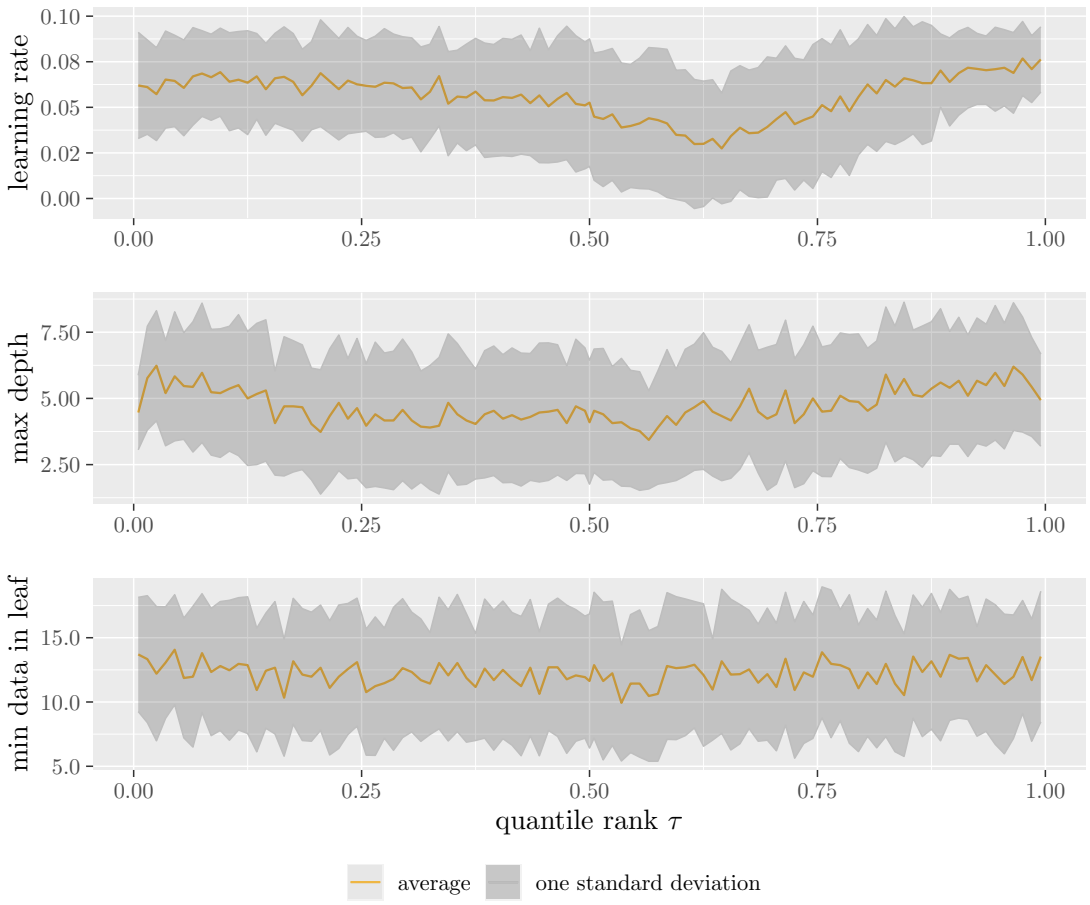

Figure B.3: **Distributions of Hyperparameters of LighGBM for All Quantile Models**

The figure presents the time-series averages and bands of one standard deviation of the three hyperparameters used in LightGBM. From up to bottom, the pictures show the time-series averages for learning rate, maximum depth, and minimum number of observations on a leaf. In each plot, the horizontal axis shows the indicator for the 100 quantiles, ranging from 0.5% to 99.5%. The vertical axis shows the hyperparameter value. The thicker curves show the time-series averages of the hyperparameter across the 30 years. The bands show the plus (minus) one standard deviations of the hyperparameters from the corresponding time-series averages.

C Solution to the Multiple Comparisons Problem with q Value

Since we perform hypotheses tests for all the firms simultaneously, we need address the multiple comparisons problem. The multiple comparisons problem refers to when one tests several hypotheses simultaneously, she might reject some hypotheses by mistake. Popular corrections for the multiple comparisons problem include Bonferroni (1936) method and its modifications that control the family wise error rate (FWER), Benjamini and Hochberg (1995) that controls the false discovery rate (FDR) and Storey (2003) that controls positive false discovery rate (pFDR). The Bonferroni (1936) method is the most strict but lack practical value when we compare thousands of hypotheses. The Benjamini and Hochberg (1995) approach works on controlling the proportion of incorrect rejections out of all the hypotheses considered. Storey (2003) is based on Benjamini and Hochberg (1995) and control the incorrect rejection rate over only the ones that are actually rejected. We adopt the more conservative Storey (2003) approach and use their proposed q-value to control the pFDR[12].

The definition of q-value is quite similar to that of p-value. For instance, for a two-sided single hypothesis test with a z statistic, the p-value is defined as

$$\Pr(|Z| \geq z_0 | H_0 \text{ is true}), \tag{C.7}$$

where Z is the test statistic and z_0 is its observation and H_0 is the null hypothesis. Similarly, q-value is defined as

$$\Pr(H_0 \text{ is true} || Z| \geq z_0). \tag{C.8}$$

Intuitively, the q-value measures the probability of an incorrect rejection, given a significant test statistic. The formal proof of the validity of q-value relies on the equivalence between pFDR and $\Pr(H_0 \text{ is true} || Z| \geq z)$, which is shown in Theorem 1 of Storey (2003). With smaller q-values (5% in our paper), we are confident to believe that the rejected density forecasts are in fact inaccurate. Moreover, the identical distribution of KS, Shapiro, and LR tests satisfy the requirement of Storey (2003).

Specifically, LightGBM still offers the most accurate density forecasts among the three machine learning methods. The median and mean q-values for LightGBM of the LR tests are 0.3334 and 0.3101; those for PCR are 0.1794 and 0.1896, while for Lasso, the numbers decrease to 0.0850 and 0.1115. The proportions of q-values being greater than 5% are reported in the last column of Table

[12]The research is burgeoning in controlling both the Type I and Type II errors. For example, see Harvey, Liu, and Zhu (2015) and Harvey and Liu (2020). We note that in a large bootstrap analyzing 18,000 tests, Harvey and Liu's (2020) results show that Storey (2003) is the most powerful, in that it offers the most strict control of the Type II error rate. In our context, a Type II error occurs when we falsely claim that one of the density forecasts is accurate. Our adoption of the q-value thus warrants our goal of controlling false claims of accurate density forecasts.

C.1. More than 90% of the LightGBM forecasts have larger q-values greater than 5%, showing that LightGBM has the strongest predictive power, followed by PCR with about 80% of firm passing the tests. Lasso produces about 70% accurate forecasts.

Table C.1: **Evaluation of Conditional Density Forecasts Using q-values**

This table reports the evaluation of the conditional density forecasts using the Kolmogorov-Smirnov (KS) test, the Shapiro normality test, and the Berkowitz (2001) likelihood ratio (LR) test for PCR, Lasso, and LightGBM with the Storey (2003) q-value approach. From top to bottom, the q-values are obtained for PCR, Lasso, and LightGBM predictions. The last column, denoted as $P(> 0.05)$ shows the proportion of firm-time observations that pass the corresponding test at 0.05 level, i.e. $q > 0.05$.

ML	test	min	Q1	median	Q3	max	mean	$P(> 0.05)$
PCR	KS	0.0000	0.0821	0.1475	0.2169	0.3148	0.1498	0.8596
	Shapiro	0.0001	0.1075	0.1806	0.2540	0.3665	0.1823	0.9224
	LR	0.0000	0.0535	0.1794	0.3156	0.4444	0.1896	0.7584
Lasso	KS	0.0000	0.0314	0.0853	0.1543	0.2501	0.0958	0.6606
	Shapiro	0.0000	0.1119	0.2049	0.2934	0.4136	0.2044	0.8951
	LR	0.0000	0.0143	0.0850	0.1921	0.3340	0.1115	0.6041
LightGBM	KS	0.0004	0.0970	0.1593	0.2194	0.3097	0.1580	0.9090
	Shapiro	0.0005	0.1914	0.2564	0.3222	0.4233	0.2529	0.9789
	LR	0.0000	0.1822	0.3334	0.4503	0.5552	0.3101	0.9183