A Novel Differentiable Rank Learning Method Towards Stock Movement Quantile Forecasting

Chenyou Fan\textsuperscript{a}, Hengyang Lu\textsuperscript{b} and Aimin Huang\textsuperscript{c};

\textsuperscript{a}South China Normal University
\textsuperscript{b}Jiangnan University
\textsuperscript{c}Hangzhou Higgs Asset Management Co., Ltd.

ORCiD ID: Chenyou Fan https://orcid.org/0000-0002-9835-8507, Hengyang Lu https://orcid.org/0000-0001-5321-705X, Aimin Huang https://orcid.org/0000-0001-9895-3202

Abstract. We focus on Stock Movement Forecasting (SMF) using AI techniques to develop modern automated trading systems. Previous studies with deep-learning-based methodology have only considered binary up-or-down trends, ignoring the importance of fine-grained categorization of the stock movements to facilitate decision-making. However, the challenges of SMF arise from the randomness of the global market impacting cross-sectional stocks and the volatility of internal dynamics in each time series. To address these challenges, we present a novel end-to-end learning-to-rank framework that incorporates both market-level and stock-level dynamics. Specifically, we aim to identify cross-sectional stocks that exhibit notable movements at every time step and learn to rank steps with the most significant movements in the temporal dimension. We conduct extensive evaluations of our multi-task learning framework utilizing real-world market data, which demonstrate superior performance when compared to state-of-the-art methods, with improvements in the Gain and Sharpe Ratio by 5-15%.

1 Introduction

The prediction of market price movements is a crucial aspect of Fin-tech AI research, and is accomplished through the use of data mining \cite{6,15} and artificial intelligence techniques \cite{21,22}.

Price movement forecasting applies to many financial assets such as futures and options, carbon credits, commodities, and more. It can benefit individual investors by helping them anticipate market risks, and help policy makers to counter sudden spikes in consumer product prices, thereby promoting overall social welfare. An emerging area of interest within this field is the prediction of carbon credit prices, specifically within the leading carbon trading markets such as the European Union Emission Trading Scheme (EU ETS) \cite{13}. This can facilitate the achievement of a balance between economic growth and reduced greenhouse gas emissions, thus promoting sustainability.

This study focuses on the most common task of Stock Movement Forecasting (SMF), which can naturally extend to other financial assets. Many recent studies have simplified SMF \cite{4,7,19,21} as a binary classification task that focuses solely on prominent rising and falling steps, ignoring neutral movements. This was done to address the issue of class imbalance, where tiny movements within ±0.1% can account for up to 40% of per-minute trading scenarios. However, this approach is impractical as neutral predictions play a key role in realistic decision-making, and predicting only the binary up-or-down movement is too coarse for a practical trading strategy. We show that a finer-grained prediction is necessary to depict the distribution of movements and make adaptive decisions, such as determining the optimal trading quantities.

To this end, we focus on modeling SMF as a fine-grained Distributional Quantile Classification (DQC) task. Initially, we collect statistical data on historical stock movement, which is then partitioned into a series of finely divided quantile intervals to define the range of movement distribution. As in Fig. 1 this allows us to forecast the future movement distribution across the designated domain. The probability assigned to each quantile interval reflects the probability of the corresponding movement range occurring. Furthermore, utilizing the predicted fine-grained movements, we can carry out a realistic trading gain assessment to measure the quality of the forecasts.

In this paper, we propose a unique learning-to-rank framework that simultaneously explores the market-level dynamics and stock-level volatility. For modeling the market-level dynamics, we search for stocks with the most significant movements at each time step. For modeling the stock-level volatility, we learn to rank steps with the most significant movements in the temporal dimension. This learning-to-rank formulation integrates seamlessly into the movement quantile prediction as a multi-task learning objective. We evaluate our methods with practical financial metrics on realistic datasets and show their...
superior performance to existing state-of-the-art methods.

We summarize our contributions in this study as follows.

1. We propose a price movement forecasting framework with advanced deep-learning based sequence prediction methodology.
2. We integrate market-level contextual features from market indices and capture stock-level variations from individual stock patterns with our differentiable learning-to-rank methodology.
3. We show that our model outperforms state-of-the-art approaches by 5-15% by evaluating with practical financial metrics including trading gain and Sharpe ratio.
4. Our framework could improve social welfare and improve AI topics such as metric learning and recommendation systems.

2 Related Work

Stock Movement Forecasting (SMF) has received more attentions from economics, financial data engineering and machine learning communities. Recent studies [21,22] of SMF focus on modeling the temporal patterns from historical data for predicting the future and formulate SMF as a regression task [1,10,21] or a classification task [7,19,20]. We formulate SMF as a fine-grained classification task with distributional estimation which is robust to imbalanced data.

Recent SMF studies adopt Deep-Learning (DL) models to perform feature and pattern learning from data, such as Auto-Regression Neural Networks [16,18] and Recurrent Neural Networks (RNNs and LSTMs) [6,7,21]. The most recent Transformer [14] models also show superior performance in SMF. Ding et al. [4] and Lim et al. [10] improved Transformer attention with prior knowledge and interpretability respectively. Lim et al. [10] and Yoo et al. [20] combined Transformer with RNN structure to better encode past and future inputs. Our framework (Fig 2) follows the state-of-the-art Transformer encoder-decoder design. We also design a novel modal fusion module to combine market global context into individual stock prediction for better performance.

In this study, we employed a recent learning-to-rank (LTR) technique [3,17] based on Optimal Transport technique [2] to efficiently finding top elements of within a list of numbers. The pioneering work RSR [8] proposed to learn relations of stocks with a pairwise order loss. However, RSR couples the prediction of each stock with all other stocks, which requires using data of the entire market to predict an individual one. It is still unclear how to enable the rank of stocks to improve forecasting accuracy efficiently. Our method can explicitly predict the order of a dynamic stock list at any future step, improving forecasting accuracy and gains.

3 Tasks and Approaches

We first review the stock movement forecasting (SMF) task and related data and features. Then we demonstrate our learning framework and multi-task objectives to fulfill the task.

We consider a market of \( N \) common stocks and \( M \) stock market indices (e.g. S&P 500). We observe the past \( T_h \) steps \( T_h = \{1, 2, ..., T_h\} \) and predict the subsequent \( T_f \) future steps \( T_f = \{T_h + 1, ..., T_h + T_f\} \) as the SMF task. Depending on the data frequency, one step can be a day for daily trading data or a minute for per-minute trading data.

3.1 Input data and processed time-series features

**Numeric Features.** The stock market makes publicly available various price values (e.g. open, high, low, and close) and trading values

<table>
<thead>
<tr>
<th>No.</th>
<th>Features</th>
<th>Formulaion from raw data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>close</td>
<td>e.g., ( v_{\text{close}} = \text{close}<em>t / \text{close}</em>{t-1} - 1 )</td>
</tr>
<tr>
<td>2</td>
<td>open/high/low</td>
<td>e.g., ( v_{\text{open}} = \text{open}<em>t / \text{close}</em>{t-1} - 1 )</td>
</tr>
<tr>
<td>3</td>
<td>avg, k = 5, 10, ...</td>
<td>e.g., ( v_{\text{avg}} = \sum_{i=5}^{k} \text{close}_{t-1} / k - 1 )</td>
</tr>
<tr>
<td>4</td>
<td>vol/amtd</td>
<td>e.g., ( v_{\text{vol}} = \text{vol} / \text{vol}_{t-1} + 1 )</td>
</tr>
</tbody>
</table>

(e.g., trade number, volume, and amount) at each time step. We process the numeric data as previous works [7,20] as Table 1 shows.

First, the close price movement \( v_{\text{close}} \) is defined as the relative change of close price between step \( t \) and \( t + 1 \) such that \( v_{\text{close}} = (\text{close}_t / \text{close}_{t-1} - 1) \); Second, the open/high/low prices relative to close price are similar. Third, the moving average of close price. Last, the relative change of trade number, volume and amount.

Importantly, the close price movements \( v_{\text{close}} \) at future steps are the prediction targets of SMF. Note that all numeric features depend on real-time market changes thus are available only at historical steps, but not future steps.

**Categorical Features.** We convert the following discrete and time-related categorical inputs into numerical embeddings. The afternoon and minute (if available) indicate AM-or-PM and the minute no. in the 4-hour trading period per day, respectively. The date and weekday denote the order of trading date in a year (0-365) and a week (0-4), respectively. Note that these time-related inputs are available at both historical and future steps as they are sequential.

**Feature Fusion.** At historical steps, we concatenate numeric inputs and categorical embeddings, then project to a hidden feature of dimension \( d_{\text{model}} \) as historical inputs \( X^h = [x_1; ...; x_{T_h}] \in \mathbb{R}^{T_h \times d_{\text{model}}} \). At future steps, we only project categorical embeddings to \( d_{\text{model}} \) as future inputs \( X^f = [x_{T_h+1}; ...; x_{T_h+T_f}] \in \mathbb{R}^{T_f \times d_{\text{model}}} \).

3.2 Division of movement intervals as classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Move ( A_P ) (%)</th>
<th>Percentage</th>
<th>M-Quantile</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>&lt;0.1</td>
<td>10%</td>
<td>0.05</td>
</tr>
<tr>
<td>1</td>
<td>0.1-0.2</td>
<td>10%</td>
<td>0.15</td>
</tr>
<tr>
<td>2</td>
<td>0.2-0.3</td>
<td>10%</td>
<td>0.25</td>
</tr>
<tr>
<td>3</td>
<td>&gt;0.3</td>
<td>10%</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>0.3-0.4</td>
<td>10%</td>
<td>0.75</td>
</tr>
<tr>
<td>5</td>
<td>0.4-0.5</td>
<td>10%</td>
<td>0.85</td>
</tr>
<tr>
<td>6</td>
<td>&gt;0.5</td>
<td>10%</td>
<td>0.95</td>
</tr>
</tbody>
</table>

In order to make precise predictions about forthcoming price changes, we slice a collection of quantile intervals to serve as the range of movement distribution.

Initially, we gather price movement statistics from the training data and partition the range of movements into \( C = 7 \) intervals to create a \( C \)-class space \( C = \{0, 1, ..., 6\} \). Table 2 presents each interval along with its corresponding empirical data percentage and the approximate quantile of the median movement (M-Quantile) for that specific interval.

Notably, the Class-3 stands for neutrality of small movements that fall within the \([-0.1\%, 0.1\%]\) range, which account for approximately 40% of historical movements. These small movements are flanked by two distinct movement classes: Class-0 represent downward movement intervals, while Class-4-6 encompass upward movement intervals. Both downward and upward movement classes have roughly the same distribution, with each accounting for approximately 10% of total movements. At the two extremes of the movement spectrum, Class-0 and Class-6 exhibit large movements of less than -0.3% and greater than 0.3%, respectively, providing an effective method of estimating sharp fluctuations.
We show the designed DQC-Rank learning framework in Fig. 2. There
where
Thus, the output
Then we use each stock feature
features such as future date and step, details in Sec. 3.1) and decode
market. The per-step outputs are shown in Fig. 2(b).

We design a new approach to modeling market-level and stock-level
dynamics using a learning-to-rank method. Our methodology involves
two important steps.
The first step involves defining the inter-stock ranking, wherein we
identify the largest moving future stocks across the entire market
to obtain market-level insights. The second step involves defining the
intra-stock ranking, wherein we identify the largest moving future
time steps for each individual stock to uncover its internal patterns at
the stock-level.

Notations. Suppose we have a list of \( N \) scalar numbers and a
predefined \( K \) with \( 0 < K \leq N \). With a binary class space \( B = \{0, 1\} \), we label the top-\( K \) smallest elements in the list as class 0,
while the rest \( N - K \) elements are in class 1. Let \( I_N \) be an all-ones
vector of dimension \( N \).

4.1 Inter-Stock Rank Learning (Inter-Rank)

The objective of Inter-Rank is to identify the top-\( K \) stocks that are
rising or falling, within a pool of \( N \) cross-sectional stocks in
the market, in order to capture the market-level context that may impact groups of time series. Fig.3 illustrates the top rising and two falling instances. To achieve this, we formulate a task that involves distinguishing the top-\(K\) changing time-series from the remaining \((N - K)\) time-series.

Given the quantile logit prediction \(\hat{\psi}\) as in \(3\), we can estimate the expected movement by \(g_i = \sum_{i,v} v_i \sigma(\psi(v_i))\), where \(\sigma\) is softmax operation, \(v_i\) is median value of the discretized movement interval. A large positive/negative \(g_i\) indicates rapid rising/falling trend. Let \(\mathcal{G} = \{g_i\}_{i=1}^N\) be expected movements for all \(N\) stocks at the same time step.

We choose a proper positive number \(\eta > 0\) such that \(g_i\) is roughly within \((-\eta, \eta)\), e.g., a stable choice is the range of the neutrality class. Now we build two cost matrices \(C_r, C_f \in \mathbb{R}^{N \times 2}\) for rising and falling cases respectively such that:

\[
C_r = \begin{bmatrix}
(g_1 - \eta)^2, (g_1 + \eta)^2 \\ 
\vdots \\ 
(g_N - \eta)^2, (g_N + \eta)^2
\end{bmatrix},
\]

while \(C_f\) is built by swapping the two columns of \(C_r\).

The first and second column of \(C_r\) indicates the difference of \(g_i\) to \(\eta\) and \(-\eta\) respectively. The following discussion will focus on the task of finding the top-\(K\) (out of \(N\)) rising stocks, which can extend to the falling case symmetrically.

We formulate finding the top-\(K\) elements as an Optimal Transport (OT) \(\text{(12)}\) task, which seeks to assign each stock to class 0 (if top-\(K\)) or 1 (if non-top-\(K\)) with costs \(C_r\). The minimal cost assignment \(S \in \mathbb{R}^{N \times 2}\) satisfies

\[
S^* = \arg\min_{S \geq 0} \langle S, C_r \rangle - \lambda H(S),
\]

s.t. \(\sum_{i,j} S_{i,j} \log S_{i,j}\) is an entropy term. The Cond-1 ensures each row of \(S\) sums to 1 (each stock is either assigned class 0 or 1), and Cond-2 ensures class 0 and 1 have \(K\) and \(N - K\) elements respectively. The optimal solution \(\{S^*_i\}_{i=1}^N\) is less than \((g_1 + \eta)^2\) (2nd column) which assigns it as class-1 (non-top-\(K\)) with positive \(\eta\). Conversely, a negative \(g_i\) favors assignment of class-1.

**Lemma 1** The OT plan \(S^*_i\) of Problem \(7\) with entropy weight \(\lambda = 0\) provides the exact top-\(K\) highest movement \(g_i\) out of all \(N\) movements \(\mathcal{G}\), indexed by

\[
A^*_i = \{A^*_1, \ldots, A^*_N\} = S^*_i \cdot [0, 1]^T,
\]

s.t. \(A^*_i = 0\), if \(g_i\) is a top-\(K\) element, 

\(1\), if \(g_i\) is a non-top-\(K\) element,

in which \(A^*_i\) indicates the class of each stock \(i\).

**Lemma 2** shows that a special case of task \(7\) with \(\lambda = 0\) can yield optimal top-\(K\) selection, i.e., the solution \(S^*_i \in \mathbb{R}^{N \times 2}\) indicates the optimal assignment of each stock to be top-\(K\) (class 0) or not (class 1). Proof details are similar in \(17\). However, solving Problem \(7\) with \(\lambda = 0\) is linear programming \(\text{(12)}\) with a high time complexity of \(O(N^3 \log N)\) and is not differentiable.

Fortunately, we can solve the convex relaxation of Problem \(7\) with \(\lambda > 0\) in a reduced complexity of \(O(N^2 \log N)\). We adapt Sinkhorn Algorithm \(\text{(2)}\) to solve \(\hat{S}_i^*\) and the related \(A^* = \hat{S}^*_i \cdot [0, 1]^T\) in an iterative and differentiable way. We can extend to finding top-\(K\) falling stocks by replacing \(C_r\) with \(C_f\) in Problem \(7\) which yields the solution \(S_f^*\) and \(A_f^*\).

Once we have the estimated top-\(K\) rising and falling stock indicator \(\hat{A}_r^*\) and \(\hat{A}_f^*\), we can formulate a loss minimization procedure to optimize the Inter-Stock rank learning during training stage as follows.

We first retrieve the true top-\(K\) rising and falling stock list to build the ground-truth indicator \(A_r^*\) and \(A_f^*\) respectively. We thus can minimize the L2-distance between the estimated indicators against the true indicators as:

\[
\ell^f(S_i, I_r) = \frac{1}{2N \cdots ||A_r^* - A'_f||_2^2, \text{ top rising} }.
\]

\[
\ell^r(S_i, I_f) = \frac{1}{2N \cdots ||A_f^* - A'_r||_2^2, \text{ top falling} }.
\]

With the chain rule on Eq.(6)-(8), the predicted ranking \(\hat{A}_r^*\) and \(\hat{A}_f^*\) depend on movement predictions \(\hat{g}_i\) for each stock. Thus, minimizing \(\ell^r\) and \(\ell^f\) help optimize the movement predictions towards their true values which align with the correct ranking of top moving time series.

Finally, we average the inter-loss over all \(T_f\) future steps as the complete **Inter-Stock Ranking Loss** as:

\[
L_{\text{inter}} = \frac{1}{2T_f} \sum_{t=1}^{T_f} (\ell^r_t + \ell^f_t).
\]

### 4.2 Intra-Stock Rank Learning (Intra-Rank)

![Figure 4: Demo of intra-stock ranking by finding the top-K (k= 2) rising and falling future steps of one stock.](image)

In complementary to Inter-Stock, we further propose the **Intra-Rank** which aims to explore the internal volatility of each time-series by learning to identify its top-\(K\) rising and falling steps over all future steps, as shown in Fig.4.

Let \(\mathcal{G}' = \{g_{r,h+i}\}_{i=1}^v\) be movement predictions of a stock over future steps. With a similar choice of \(\eta\) as \(\mathcal{G}\), we build the cost matrix \(D_r \in \mathbb{R}^{T_f \times 2}\) for rising case such as:

\[
D_r = \begin{bmatrix}
(g_{r+1} - \eta)^2, (g_{r+1} + \eta)^2 \\ 
\vdots \\ 
(g_{r+T_f} - \eta)^2, (g_{r+T_f} + \eta)^2
\end{bmatrix},
\]

while the falling case cost matrix \(D_f\) can be built by swapping the two columns of \(D_r\).
We formulate the task of finding the top-\(K'\) rising (out of \(T\)) future steps as an Optimal Transport task which seeks to assign each step to class 0 (top-\(K'\)) or 1 (non-top-\(K'\)) with costs \(D_{\pi}\). The minimal cost assignment \(P^* \in \mathcal{P}^{T/2 \times 2}\) satisfies
\[
P^* = \arg \min_{P^{T/2 \times 2}} \langle P, D_{\pi} \rangle - \lambda \cdot H(P),\]
s.t. \(P_{12} = 1_{T/2} = [K', T - K']^T\).

The optimal solution \((P^*)_t, t \in [1, T]\) indicates the best assignment of each step \(t\) to be top-\(K'\) (class 0) while \((P^*)_t[1, T] = (0, 1)\) indicates non-top-\(K'\).

**Lemma 2** The optimal plan \(P^*_t\) of Problem (12) with entropy weight \(\lambda = 0\) provides the exact top-\(K'\) highest \(g_{t, n+1}\) out of \(Q\), indexed by
\[
E_t^c = \{E_t^{c_1}, \ldots, E_t^{c_{K'}}\}^T = P^*_t \cdot [0, 1]^T,
\]
s.t. \(E_t^c = \begin{cases} 0, & \text{if } g_t \text{ is a top-}K'\text{ element}, \\ 1, & \text{if } g_t \text{ is a non-top-}K'\text{ element}, \end{cases}\)
in which \(E_t^c\) indicate the class of each time step \(t\).

Similar to discussions under Lemma[1] we formulate Problem (12) with \(\lambda > 0\) as its convex relaxation and estimate the optimal solution \(P^*_t\) and its depending top-\(K'\), rising step indicator \(E_t^r\) and top-\(K'\) falling step indicator \(E_t^f\).

In training, the true top-\(K'\) rising and falling steps are taken as the the ground-truth indicator \(\hat{E}_t^r\) and \(\hat{E}_t^f\) respectively. We minimize the L2-distance from the estimated step indicator \(\hat{E}_t^r\) and \(\hat{E}_t^f\) to the ground-truth indicator as:
\[
E_t^r(P_t, I_t) = \frac{1}{2T} \lVert \hat{E}_t^r - E_t^r \rVert_2, \quad \text{top rising} ;
\]
\[
E_t^f(P_t, I_t) = \frac{1}{2T} \lVert \hat{E}_t^f - E_t^f \rVert_2, \quad \text{top falling} .
\]

Finally, we average the intra loss for each of \(N\) stocks as the complete Intra-Stock Ranking Loss as follows:
\[
L_{\text{intra}} = \frac{1}{2 \cdot N} \sum_{t=1}^{N} (E_t^r + E_t^f).\]

**5.1 Datasets**

KDD-17[21] has daily prices of 50 top performing US stocks from 10 sectors, which splits into training (2007-01 to 2015-07), validation (2015-08 to 2015-09), and testing (2015-10 to 2015-12).

ACL-18[19] contains daily prices of 98 US stocks with top capital sizes, which splits into training (year 2007-2014), validation (year 2015), and testing (year 2016). For both datasets on US market, we use S&P-500 (SPY) as the stock index. We take 5 days as history and predict the closing prices at next 5 days.

CSI-21 (ours) is self-collected from 800 China A-shares from 2018-2021. The stock set consists of the first 300 and the next 500 top performing stocks in the market, corresponding to the CSI-300 and CSI-500 indices. We collect the per-minute stock prices at 240 trading steps in a 4-hour trading day. We take 5 minutes as history and predict the closing prices at next 5 minutes.

To better evaluate our methods, we set three rolling-based dataset splits on CSI-21 as follows. Each split includes a test set with stock price (1) from 2021-04 to 2021-06 (downtrend market), (2) from 2021-01 to 2021-03 (fluctuating market), and (3) from 2020-10 to 2020-12 (uptrend market), respectively. For each split, all data ahead of the test set is further divided for training and validation. We will release this dataset and our splits.

**5.2 Our Methods and Baselines**

DQC-Plain is our proposed Quantile Classification framework with proposed Encoder-MMF-Decoder design and optimizes with DQC loss (Sec. 5.3). DQC-Rank further improves DQC-Plain by incorporating our proposed Rank learning framework in Sec. 5 which utilizes inter- and intra-stock rank learning objectives to regularize the model.

We compare our methods with the following studies. LightGBM[9] is the gradient boosting tree with a multi-class learning objective to predict the movement quantiles of future steps. ALSTM[7] uses LSTM model with temporal attention and adversarial training to model price sequence. RSR[8] uses GNN and pairwise ranking to explore stock relations. TFT[10] combines LSTM and Transformer with a modified interpretable multi-head attention design. DTM[9] couples all stocks by combining their features as global context. In contrast, we perform inter-stock ranking to learn the market context only in training stage. This makes our model robust to single stock variation and much faster for single-stock inference.

**5.3 Evaluation Metrics**

Accuracy(\(\uparrow\)). We evaluate the 7-way quantile classification accuracy which denotes as \(\text{QAcc}\). To evaluate the performance against class-imbalance issue, we also report the average per-class quantile accuracy denoted as \(\text{PAcc}\).

MCC(\(\uparrow\)) (Matthews Correlation Coefficient) is commonly used[19] to evaluate binary classification (positive or negative), which ranges from \([-1, 1]\).

Gain(\(\uparrow\)). We evaluate the trading gains based on the buy-hold-sell strategy[3] and our fine-grained quantile predictions. We buy 2 stocks with predicted class-5/6, buy 1 stock with class-4, hold with neutral class-3, and short-sell 2 stocks with class-0/1 and 1 stock with class-2, respectively. The Gain accumulates the one-step price change times with current position at each future step.

MDD(\(\downarrow\)). The Maximum DrawDown (MDD) metric calculates the maximum gap of the return rates from peaks to consecutive troughs within the test period. The lower MDD the better model is, as it measures the risk level.
SR\(^{↑}\). The Sharpe Ratio (SR) is a common metric in financial study to measure the return rate adjusted with volatility such as \(\text{SR} = \frac{\text{Gain}}{\sigma} \), in which \(\sigma\) is standard deviation. A higher SR indicates a greater return relative to its risk taken, thus the better strategy.

### 5.4 Results and Analysis

#### Table 3: Average result on CSI-21 with three rolling splits.

<table>
<thead>
<tr>
<th>Method \ Setting</th>
<th>QAcc (%)↑</th>
<th>PAcc (%)↑</th>
<th>MCC (%)↑</th>
<th>MDD (%)↓</th>
<th>Gain (%)↑</th>
<th>SR (%)↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALSTM [7]</td>
<td>44.2</td>
<td>21.2</td>
<td>-8.34</td>
<td>2.20</td>
<td>8.07</td>
<td>0.88</td>
</tr>
<tr>
<td>TFT [10]</td>
<td>31.4</td>
<td>32.0</td>
<td>-2.38</td>
<td>3.82</td>
<td>14.8</td>
<td>1.09</td>
</tr>
<tr>
<td>DTLML [20]</td>
<td>33.6</td>
<td>32.7</td>
<td>-1.76</td>
<td>3.79</td>
<td>12.7</td>
<td>1.10</td>
</tr>
<tr>
<td>RSR [8]</td>
<td>34.5</td>
<td>32.6</td>
<td>-6.71</td>
<td>2.98</td>
<td>12.9</td>
<td>1.07</td>
</tr>
<tr>
<td>DQC-Plain (ours)</td>
<td>35.2</td>
<td>34.1</td>
<td>-6.22</td>
<td>3.32</td>
<td>18.1</td>
<td>1.19</td>
</tr>
<tr>
<td>DQC-Rank (ours)</td>
<td>36.1</td>
<td>34.2</td>
<td>-4.94</td>
<td>3.05</td>
<td>21.4</td>
<td>1.26</td>
</tr>
</tbody>
</table>

Results of CSI-21. We show the results on CSI-21 in Table 3, which collects per-minute high-frequency trading data in China market. We summarize the results as follows.

- **DQC-Rank outperforms other baselines in Gain and SR**, leading the second place DQC-Plain by 14.4% in Gain (21.4 vs. 18.7) and 5.9% in SR (1.26 vs. 1.19), at a cost of a larger MDD (retraction).
- The overall trends of the three splits are downward, oscillating and upward, respectively. The Gain of each split is 7.8%, 23.4% and 33.0%, respectively, contributing to an averaged 21.4% in Gain (last row, col. Gain in Table 3).
- Due to the strong class-imbalance issue, LightGBM and ALSTM (shaded grey) had a seemingly high overall Acc (QAcc) (39.2% and 44.2%) while got extremely low in per-class (PAcc) (26.1% and 21.2%) and SR (0.16 and 0.88).
- We visualize the predictions in Fig. 5 and observe that DQC-Rank better fits to the true movements than baselines.

#### Results of ACL-18 and KDD-17. We show results on KDD-17 and ACL-18 in Table 4 and observe the following trends.

<table>
<thead>
<tr>
<th>ACL-18</th>
<th>QAcc (%)↑</th>
<th>PAcc (%)↑</th>
<th>MCC (%)↑</th>
<th>MDD (%)↓</th>
<th>Gain (%)↑</th>
<th>SR (%)↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>LightGBM [9]</td>
<td>24.9</td>
<td>15.0</td>
<td>-41.4</td>
<td>1.96</td>
<td>-4.47</td>
<td>-0.10</td>
</tr>
<tr>
<td>ALSTM [7]</td>
<td>42.9</td>
<td>30.6</td>
<td>36.3</td>
<td>1.07</td>
<td>25.1</td>
<td>0.61</td>
</tr>
<tr>
<td>TFT [10]</td>
<td>41.0</td>
<td>39.2</td>
<td>37.8</td>
<td>1.17</td>
<td>37.2</td>
<td>1.74</td>
</tr>
<tr>
<td>DTLML [20]</td>
<td>41.6</td>
<td>38.2</td>
<td>30.9</td>
<td>1.12</td>
<td>36.8</td>
<td>2.02</td>
</tr>
<tr>
<td>RSR [8]</td>
<td>41.1</td>
<td>39.7</td>
<td>37.8</td>
<td>1.15</td>
<td>35.9</td>
<td>2.21</td>
</tr>
<tr>
<td>DQC-Plain (ours)</td>
<td>41.3</td>
<td>39.7</td>
<td>36.8</td>
<td>1.18</td>
<td>38.9</td>
<td>1.82</td>
</tr>
<tr>
<td>DQC-Rank (ours)</td>
<td>46.4</td>
<td>42.4</td>
<td>39.0</td>
<td>1.09</td>
<td>39.8</td>
<td>2.33</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KDD-17</th>
<th>QAcc (%)↑</th>
<th>PAcc (%)↑</th>
<th>MCC (%)↑</th>
<th>MDD (%)↓</th>
<th>Gain (%)↑</th>
<th>SR (%)↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>LightGBM [9]</td>
<td>34.5</td>
<td>17.3</td>
<td>-38.6</td>
<td>2.62</td>
<td>3.57</td>
<td>0.09</td>
</tr>
<tr>
<td>ALSTM [7]</td>
<td>42.1</td>
<td>14.3</td>
<td>3.15</td>
<td>1.80</td>
<td>5.94</td>
<td>0.18</td>
</tr>
<tr>
<td>TFT [10]</td>
<td>34.9</td>
<td>31.1</td>
<td>10.9</td>
<td>2.61</td>
<td>18.4</td>
<td>1.34</td>
</tr>
<tr>
<td>DTLML [20]</td>
<td>34.8</td>
<td>29.8</td>
<td>12.6</td>
<td>2.48</td>
<td>17.6</td>
<td>1.31</td>
</tr>
<tr>
<td>RSR [8]</td>
<td>34.8</td>
<td>33.7</td>
<td>13.3</td>
<td>2.50</td>
<td>19.9</td>
<td>1.47</td>
</tr>
<tr>
<td>DQC-Plain (ours)</td>
<td>32.6</td>
<td>32.4</td>
<td>12.5</td>
<td>2.55</td>
<td>18.7</td>
<td>1.50</td>
</tr>
<tr>
<td>DQC-Rank (ours)</td>
<td>30.7</td>
<td>35.0</td>
<td>10.6</td>
<td>1.56</td>
<td>24.5</td>
<td>1.59</td>
</tr>
</tbody>
</table>

- **On both datasets, DQC-Rank yields a highest Sharpe Ratio (SR)** with its better profit-to-volatility feature. E.g., on ACL-18 DQC-Rank has a 5% increase of SR compared with best baseline RSR (2.33 vs. 2.21); on KDD-17, DQC-Rank has a 8.2% better SR over RSR (1.59 vs. 1.47) as well.
- **DQC-Rank yields the best Gain**, leading the RSR and DTML by more than 5% relatively on ACL-18. DQC-Rank has a 28% higher SR than DQC-Plain (2.33 vs. 1.82), as its rank learning better regularizes model training and reduces volatility. Similar trends are also with KDD dataset.
- **LightGBM yields the lowest Gains** due to their lacked capacity of performing complex temporal learning.

#### Figure 5: Visualization of distributional quantile prediction.

5.5 Ablation studies

Effects of Rank Learning (RKL). We study the individual contribution of \(L_{\text{inter}}\) in Eq. (10) and \(L_{\text{intra}}\) in Eq. (15). On CSI-21 dataset, DQC-Plain with \(L_{\text{inter}}\) and \(L_{\text{intra}}\) individually achieves a relative Gain boost of 3.2% and 5.5% resp., while they together (a.k.a., DQC-Rank) achieve 9.6% boost. \(L_{\text{intra}}\) contributes more as it explores the individual temporal pattern which matters more in the per-minute
interval. On the other, with the ACL-18 dataset of daily data, $L_{inter}$ contributes slightly more (inter $+3.6\%$ v.s. intra $+2.5\%$) as the inter-market context is more informative in the daily scale. Overall, both losses are critical and complementary.

**Choice of $K$ and $K'$.** The hyper-parameter $K$ in Inter-Rank learning controls number of stocks to consider with the largest movements in each of up and down direction. To search for its optimal value, we try (5, 10, 20, ..., 50)-percent of total $N = 88$ stocks of ACL-18 dataset, with resulting SRs as (0.72, 1.19, 1.23, 0.86, 0.77, 0.89). Thus, we default setting $K = \text{round}(0.2 \cdot N)$ in our experiments. The hyper-parameter $K$ controls how many future steps we explore as most significant moves per direction, which we set $K' = 1$ out of the next $T_f = 5$ steps to predict.

**Transaction cost.** We consider adding a 0.05% trading fee for each traded stock. The Gain of CSI-21 of DQC-Rank dropped from 20.5% to 6.1%, still outperforming the second place DQC-Plain 4.2% and DTML 2.6%. The Gain of ACL-18 dropped from 39.8% to 36.4%, outperforming the second place DQC-Plain 35.4% and TFT 33.7%.

### 6 Conclusion

We study stock movement forecasting as a fine-grained quantile classification task. We formulate learning-to-rank tasks to explore global context of the market and internal moving patterns of an individual stock. Our model achieves significant improvement on realistic datasets with various evaluation metrics.

In future work, we can apply our work to other financial assets such as futures and options, carbon credits, commodities, and more. Thus it can benefit individual investors by helping them anticipate market risks and minimize losses, as well as policy makers who can take early action based on the market prices of agricultural products to promote social welfare. Furthermore, we can utilize our learning-to-rank method to improve a broad range of critical AI topics, such as metric learning and recommendation systems.

### References


