ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230328

A Novel Differentiable Rank Learning Method Towards Stock Movement Quantile Forecasting

Chenyou Fan[a], Hengyang Lu[b] and Aimin Huang[c;*]

[a]South China Normal University
[b]Jiangnan University
[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 [6, 15] and artificial intelligence techniques [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) [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 [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 $\pm 0.1\%$ 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.

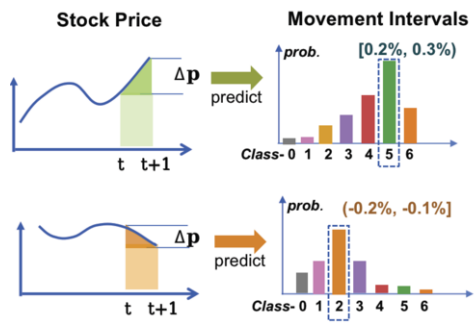

Figure 1: Illustration of price movement quantile prediction with rising case (top) and falling case (bottom). Our task is to categorize the movement at each step to its correct quantile interval.

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

* Corresponding Author. Email: huangaimin@higgsasset.com.

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 modality 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 explore 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 $\mathcal{T}_h = \{1, 2, ..., T_h\}$ and predict the subsequent T_f future steps $\mathcal{T}_f = \{T_h + 1, ..., T_h + T_f\}$ 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 (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.

Table 1: Numerical inputs and features.

No.	Features	Formulation from raw data
1	v_close	e.g., $v\_close = close_t/close_{t\text{-}1} - 1$
2	$v\_open/high/low$	e.g., $v\_open = open_t/close_t - 1$
3	$v\_avg, k = 5, 10, ...$	e.g., $v\_avg = \frac{\sum_{i=1}^{k} close_{v\text{-}i+1}/k}{close_t} - 1$
4	$v\_trade/vol/amt$	e.g., $v\_vol = vol_t/vol_{t-1} - 1$

First, the *close price movement* v_close is defined as the relative change of close price between step t and $t + 1$ such that $v\_close = (close_t/close_{t\text{-}1} - 1)$; 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_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_{model} as historical inputs $\boldsymbol{X}^h = [\boldsymbol{x}_1; \ldots; \boldsymbol{x}_{T_h}] \in \mathcal{R}^{T_h \times d_{\text{model}}}$. At future steps, we *only* project categorical embeddings to d_{model} as future inputs $\boldsymbol{X}^f = [\boldsymbol{x}_{T_h+1}; \ldots; \boldsymbol{x}_{T_h+T_f}] \in \mathcal{R}^{T_f \times d_{\text{model}}}$.

3.2 Division of movement intervals as classes

Table 2: Movement intervals and quantiles of CSI-21 dataset.

Class $\mathcal{C}^{qt}$	0	1	2	**3**	4	5	6
Move Δp (%)	<-0.3	<-0.2	[-0.2,-0.1)	[-0.1,0.1]	>0.1	>0.2	>0.3
Percentage	10%	10%	10%	40%	10%	10%	10%
M-Quantile	0.05	0.15	0.25	0.5	0.75	0.85	0.95

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 $\mathcal{C} = \{0, 1, ..., 6\}$. 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 0-2 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.

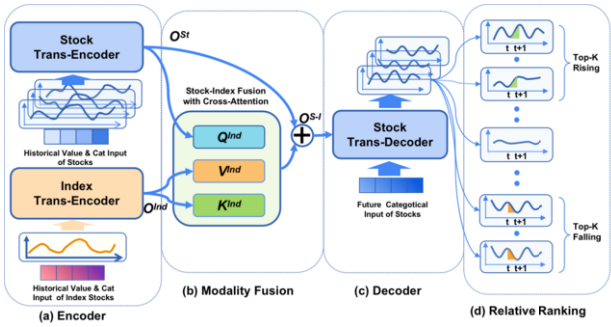

Figure 2: Overview of DQC-Rank learning framework with Encoder, Multi-Modality Fusion, Decoder and Rank Learning modules.

3.3 Model architecture

We show the designed DQC-Rank learning framework in Fig. 2. There are four main components: (a) an input encoder, (b) a stock-index modality fusion module, (c) an output decoder, and (d) a multi-task learning-to-rank module.

Abbreviations. We abbreviate the common Transformer-based attentions [14] to simplify notations.

- *Mult-Head*$(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})$ denotes the multi-head attention which encodes input features with multiple dot-product attentions.
- *Self-Attention*$(\boldsymbol{X})$ is a special multi-head attention in which the query, key and value features are linearly projected from a same feature $\boldsymbol{X}$.
- *Cross-Attention*$(\boldsymbol{X}^1, \boldsymbol{X}^2, \boldsymbol{X}^3)$ generalizes a multi-head attention in which the query, key and value features are linearly projected from different feature inputs. We will specify the inputs at usage.

Encoder. Given the historical feature $\boldsymbol{X}^h \in \mathcal{R}^{T_h \times d_{\text{model}}}$, we design a feature encoder to learn temporal patterns from each stock together with each stock index simultaneously, as shown in Fig. 2(a). We utilize the standard *self-attention* mechanism to attend to all historical steps in bi-directions, encoding their temporal patterns from input $\boldsymbol{X}^h$ to the hidden feature $\boldsymbol{O}^{St}$ of each stock, such that:

$$\boldsymbol{O}^{St} \leftarrow \text{Self-Attention}(\boldsymbol{X}^h) \in \mathcal{R}^{T_h \times d_{\text{model}}} . \quad (1)$$

Similarly, we apply self-attention to each stock index m and obtain the feature $\boldsymbol{O}_m^{Ind} \in \mathcal{R}^{T_h \times d_{\text{model}}}$.

Multi-Modality Fusion (MMF). We aim to capture both the market-level dynamics which affects each individual stock, as well as the internal dynamics of each stock. Hereby, we design an adaptive MMF mechanism to embed market index information as global market context into each individual stock representation, as follows.

We first combine all M indices as the market context as:

$$\boldsymbol{O}^{Ind} \leftarrow \text{Concat}[\boldsymbol{O}_1^{Ind}; ...; \boldsymbol{O}_M^{Ind}] \in \mathcal{R}^{T_h \times (M \times d_{\text{model}})} . \quad (2)$$

Then we use each stock feature $\boldsymbol{O}^{St}$ to query $\boldsymbol{O}^{Ind}$ with cross-attention and apply a residual connection as follows:

$$\boldsymbol{O}^{S\text{-}I} \leftarrow \boldsymbol{O}^{St} \oplus \text{Cross-Attention}(\boldsymbol{O}^{St}, \boldsymbol{O}^{Ind}, \boldsymbol{O}^{Ind}) , \quad (3)$$

where $\oplus$ is an element-wise addition to fulfill residual connection. Thus, the output $\boldsymbol{O}^{S\text{-}I} \in \mathcal{R}^{T_h \times d_{\text{model}}}$ is the fused stock feature which is both stock-specific and global context-aware in terms of the whole market. The per-step outputs are shown in Fig. 2(b).

Decoder. In the next step, we combine the fused historical features $\boldsymbol{O}^{S\text{-}I}$ with observable future features $\boldsymbol{X}^f$ (time-dependent categorical features such as future date and step, details in Sec. 3.1) and decode to the future representation.

Specifically, we first explore the future temporal pattern of $\boldsymbol{X}^f$ by a standard self-attention such that $\boldsymbol{O}^f \leftarrow \text{Self-Attention}(\boldsymbol{X}^f)$. Then, we utilize the $\boldsymbol{O}^f$ as the future query which attends to the fused historical feature $\boldsymbol{O}^{S\text{-}I}$ with cross-attention such that:

$$\boldsymbol{H}^{dec} \leftarrow \text{Cross-Attention}(\boldsymbol{O}^f, \boldsymbol{O}^{S\text{-}I}, \boldsymbol{O}^{S\text{-}I}) . \quad (4)$$

The output feature $\boldsymbol{H}^{dec} \in \mathcal{R}^{T_f \times d_{\text{model}}}$ is the final hidden feature which combines historical temporal patterns with future date information. We summarize the above steps in Fig. 2(c).

3.4 Distributional Quantile Classification (DQC)

In this section, we describe our training objectives to fulfill the target of *estimating the distribution of the future stock movements* to depict the randomness of the market.

Let the classifier f produces a C-way logits over movement intervals such that $\boldsymbol{\psi}_t \leftarrow f(\boldsymbol{H}^{dec}(t)) \in \mathcal{R}^C$ for each future step t. The **Distributional Quantile Classification Loss** is the weighted cross-entropy loss of C-quantile classification averaged over all T future steps such that:

$$L^{dqc}(\boldsymbol{\psi}, \boldsymbol{z}) = \frac{1}{T} \sum_{t=1}^{T} - \underbrace{(1 - p_t)^{\gamma}}_{\text{focal-term}} \cdot \underbrace{\log p_t}_{\text{ce-loss}} , \quad (5)$$
$$s.t. \;\; p_t = \sigma(\boldsymbol{\psi}_t[z_t]) ,$$

in which σ is softmax function, z_t is the true quantile interval as class label at time t. The focal-term inherits from focal loss [11] to balance class, which is critical as the movements are unevenly distributed as Table 2 shows.

4 Inter- / Intra- Stock Rank Learning

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 top rising and declining 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 $0 < K \ll N$. With a binary class space $\mathcal{B} = \{0, 1\}$, we label the top-K *smallest* elements in the list as class 0, while the rest $N - K$ elements are in class 1. Let $\mathbf{1}_N$ be an all-ones vector of dimension N.

4.1 Inter-Stock Rank Learning (Inter-Rank)

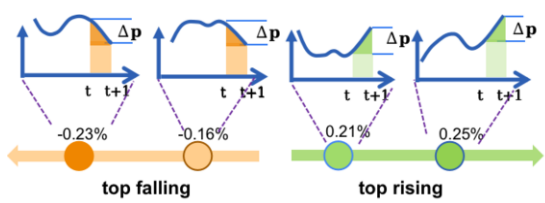

Figure 3: Demo of inter-stock ranking by finding the top-K ($K = 2$) rising and falling stocks over all stocks at step t.

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 $\hat{\boldsymbol{\psi}}$ as in (5), we can estimate the expected movement by $g = \sum_{i=1}^{C} v_i * \sigma(\hat{\boldsymbol{\psi}})[i]$, in which σ is softmax operation, v_i is median value of the discretized movement interval. A large positive/negative g 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 $\boldsymbol{C}_r, \boldsymbol{C}_f \in \mathcal{R}^{N\times 2}$ for rising and falling cases respectively such that:

$$\boldsymbol{C}_r = \begin{bmatrix} (g_1-\eta)^2, & (g_1+\eta)^2 \\ \vdots & \vdots \\ (g_i-\eta)^2, & (g_i+\eta)^2 \\ \vdots & \vdots \\ (g_N-\eta)^2, & (g_N+\eta)^2 \end{bmatrix} , \tag{6}$$

while $\boldsymbol{C}_f$ is built by swapping the two columns of $\boldsymbol{C}_r$.

The first and second column of $\boldsymbol{C}_r$ indicates the difference of g_i to η 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**) [12] task, which seeks to assign each stock to class 0 (if top-K) or 1 (if non-top-K) with costs $\boldsymbol{C}_r$. The minimal cost assignment $\boldsymbol{S} \in \mathcal{R}^{N\times 2}$ satisfies

$$\begin{aligned} \boldsymbol{S}_r^* &= \arg\min_{\boldsymbol{S}\geq 0} \langle \boldsymbol{S}, \boldsymbol{C}_r \rangle - \lambda \underbrace{H(\boldsymbol{S})}_{\text{entropy}} , \\ s.t.\ \ & \underbrace{\boldsymbol{S}\mathbf{1}_2 = \mathbf{1}_N}_{\text{Cond-1}} , \ \underbrace{\boldsymbol{S}^\top \mathbf{1}_N = [K, N-K]^\top}_{\text{Cond-2}} . \end{aligned} \tag{7}$$

in which $H(\boldsymbol{S}) = -\sum_{i,j} S_{i,j} \log S_{i,j}$ is an entropy term. The Cond-1 ensures each row of $\boldsymbol{S}$ 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 $(\boldsymbol{S}_r^*)[i,\cdot] = (1,0)$ indicates the best assignment of each stock i to be top-K (class 0) while $(\boldsymbol{S}_r^*)[i,\cdot] = (0,1)$ indicates non-top-K (class 1).

Intuitively, a positive movement g_i favors assignment of class-0 (*top-K*), as its cost $(g_i-\eta)^2$ (1st column of $\boldsymbol{C}_r$) is less than $(g_i+\eta)^2$ (2nd column) which assigns it as class-1 (*non-top-K*) with positive η. Conversely, a negative g_i favors assignment of class-1.

Lemma 1 *The OT plan $\boldsymbol{S}_r^*$ 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*

$$\begin{aligned} \boldsymbol{A}_r^c &= [A_1^c, \ldots, A_i^c \ldots, A_N^c]^\top = \boldsymbol{S}_r^* \cdot [0,1]^\top \\ s.t.\ \ A_i^c &= \begin{cases} 0, & \text{if } g_i \text{ is a top-}K \text{ element}, \\ 1, & \text{if } g_i \text{ is a non-top-}K \text{ element}, \end{cases} \end{aligned} \tag{8}$$

in which A_i^c indicates the class of each stock i.

Lemma 1 shows that a special case of task (7) with $\lambda = 0$ can yield optimal top-K selection, i.e., the solution $\boldsymbol{S}_r^* \in \mathcal{R}^{N\times 2}$ 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 [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 [2] to solve $\hat{\boldsymbol{S}}_r^*$ and the related $\hat{\boldsymbol{A}}_r^c = \hat{\boldsymbol{S}}_r^* \cdot [0,1]^\top$ in an iterative and differentiable way. We can extend to finding top-K falling stocks by replacing $\boldsymbol{C}_r$ with $\boldsymbol{C}_f$ in Problem (7) which yields the solution $\hat{\boldsymbol{S}}_f^*$ and $\hat{\boldsymbol{A}}_f^c$.

Once we have the estimated top-K rising and falling stock indicator $\hat{\boldsymbol{A}}_r^c$ and $\hat{\boldsymbol{A}}_f^c$, 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 $\boldsymbol{A}_r^c$ and $\boldsymbol{A}_f^c$ respectively. We thus can minimize the L2-distance between the estimated indicators against the true indicators as:

$$\begin{aligned} \ell^r(\hat{\boldsymbol{S}}_r, \boldsymbol{I}_r) &= \frac{1}{2N} ||\hat{\boldsymbol{A}}_r^c, \boldsymbol{A}_r^c||_2^2 \ , \ \text{top rising} \ ; \\ \ell^f(\hat{\boldsymbol{S}}_f, \boldsymbol{I}_f) &= \frac{1}{2N} ||\hat{\boldsymbol{A}}_f^c, \boldsymbol{A}_f^c||_2^2 \ , \ \text{top falling} \ . \end{aligned} \tag{9}$$

With the chain rule on Eq.(6)-(8), the predicted ranking $\hat{\boldsymbol{A}}_r^c$ and $\hat{\boldsymbol{A}}_f^c$ depend on movement predictions g_i for each stock. Thus, minimizing ℓ^r and ℓ^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:

$$L^{inter} = \frac{1}{2 \cdot T_f} \sum_{t=1}^{T_f} (\ell_t^r + \ell_t^f) \ . \tag{10}$$

4.2 Intra-Stock Rank Learning (Intra-Rank)

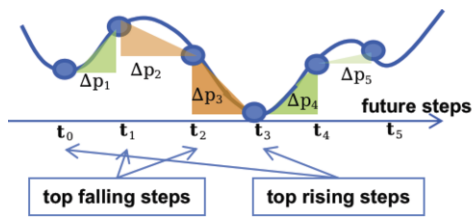

Figure 4: Demo of intra-stock ranking by finding the top-$K'(=2)$ rising and falling future steps of one stock.

In complementary to Inter-Rank, 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 $\mathcal{G}' = \{g_{T_h+t}\}_{t=1}^{T_f}$ be movement predictions of a stock over future steps. With a similar choice of η as (6), we build the cost matrix $\boldsymbol{D}_r \in \mathcal{R}^{T_f\times 2}$ for rising case such as:

$$\boldsymbol{D}_r = \begin{bmatrix} (g_{T_h+1}-\eta)^2, & (g_{T_h+1}+\eta)^2 \\ \vdots & \vdots \\ (g_{T_h+t}-\eta)^2, & (g_{T_h+t}+\eta)^2 \\ \vdots & \vdots \\ (g_{T_h+T_f}-\eta)^2, & (g_{T_h+T_f}+\eta)^2 \end{bmatrix} , \tag{11}$$

while the falling case cost matrix $\boldsymbol{D}_f$ can be built by swapping the two columns of $\boldsymbol{D}_r$.

We formulate the task of finding the top-K' rising (out of T_f) 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 $\boldsymbol{D}_r$. The minimal cost assignment $\boldsymbol{P} \in \mathcal{R}^{T_f \times 2}$ satisfies

$$\begin{aligned} \boldsymbol{P}_r^* &= \arg\min_{\boldsymbol{P} \geq 0} \langle \boldsymbol{P}, \boldsymbol{D}_r \rangle - \lambda \underbrace{H(\boldsymbol{P})}_{\text{entropy}} , \\ s.t.\ \ \boldsymbol{P}\mathbf{1}_2 &= \mathbf{1}_{T_f},\ \boldsymbol{P}^\top \mathbf{1}_{T_f} = \left[K', T_f - K'\right]^\top . \end{aligned} \tag{12}$$

The optimal solution $(\boldsymbol{P}_r^*)[t, \cdot] = (1, 0)$ indicates the best assignment of each step t to be top-K' (class 0) while $(\boldsymbol{P}_r^*)[t, \cdot] = (0, 1)$ indicates non-top-K'.

Lemma 2 *The optimal plan $\boldsymbol{P}_r^*$ of Problem (12) with entropy weight $\lambda = 0$ provides the exact top-K' highest g_{T_h+t} out of $\mathcal{G}'$, indexed by*

$$\begin{aligned} \boldsymbol{E}_r^c &= [E_1^c, \ldots, E_t^c \ldots, E_T^c]^\top = \boldsymbol{P}_r^* \cdot [0, 1]^\top \\ s.t.\ E_t^c &= \begin{cases} 0, & \text{if } g_t \text{ is a top-}K' \text{ element}, \\ 1, & \text{if } g_t \text{ is a non-top-}K' \text{ element}, \end{cases} \end{aligned} \tag{13}$$

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 $\hat{\boldsymbol{P}}_r^*$ and its depending top-K' rising step indicator $\hat{\boldsymbol{E}}_r^c = T_f \cdot \hat{\boldsymbol{P}}_r^* \cdot [0, 1]^\top$.

The above discussion can extend to finding top-K' falling steps by replacing $\boldsymbol{D}_r$ with $\boldsymbol{D}_f$ in Problem (12) which yields the solution $\hat{\boldsymbol{P}}_f^*$ and the top-K' falling step indicator $\hat{\boldsymbol{E}}_f^c$.

In training, the true top-K' rising and falling steps are taken as the the ground-truth indicator $\boldsymbol{E}_r^c$ and $\boldsymbol{E}_f^c$ respectively. We minimize the L2-distance from the estimated step indicator $\hat{\boldsymbol{E}}_r^c$ and $\hat{\boldsymbol{E}}_f^c$ to the ground-truth indicator as:

$$\begin{aligned} \mathcal{E}^r(\hat{\boldsymbol{P}}_r, \boldsymbol{I}_r') &= \frac{1}{2T_f} ||\hat{\boldsymbol{E}}_r^c, \boldsymbol{E}_r^c||_2^2 \ , \ \text{top rising} \ ; \\ \mathcal{E}^f(\hat{\boldsymbol{P}}_f, \boldsymbol{I}_f') &= \frac{1}{2T_f} ||\hat{\boldsymbol{E}}_f^c, \boldsymbol{E}_f^c||_2^2 \ , \ \text{top falling} \ . \end{aligned} \tag{14}$$

Finally, we average the intra loss for each of N stocks as the complete **Intra-Stock Ranking Loss** as follows:

$$L^{intra} = \frac{1}{2 \cdot N} \sum_{i=1}^{N} (\mathcal{E}_i^r + \mathcal{E}_i^f) \ . \tag{15}$$

4.3 Multi-task training objective

Finally, we can optimize the model by jointly minimizing the distributional quantile loss Eq.(5), inter-rank loss (10), and intra-rank loss (15) in end-to-end fashion as:

$$L^{final} = L^{dqc} + \alpha_1 L^{inter} + \alpha_2 L^{intra} \ , \tag{16}$$

where factors α_1, α_2 are searched by cross-validation. We denote our learning framework as **DQC-Rank** with multi-task learning objective as in Eq. (16).

5 Experiment

We evaluate DQC-Rank on three benchmark datasets and compare with various state-of-the-art methods.

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 88 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. 3.4). **DQC-Rank** further improves DQC-Plain by incorporating our proposed Rank learning framework in Sec. 4 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. **DTML** [20] 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(↑). We evaluate the 7-way quantile classification accuracy which denotes as **QAcc**. To evaluate the performance against class-imbalance issue, we also report the average per-class quantile accuracy denoted as **PAcc**.

MCC(↑) (Matthews Correlation Coefficient) is commonly used [7, 19] to evaluate binary classification (positive or negative), which ranges from $[-1, 1]$.

Gain(↑). We evaluate the trading gains based on the buy-hold-sell strategy [5] 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(↓). 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 $SR = \frac{\text{Gain}}{\sigma(\text{Gain})}$, in which σ 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.

Method \ Setting	QAcc (%) ↑	PAcc (%) ↑	MCC (%) ↑	MDD (%) ↓	Gain (%) ↑	SR ↑
LightGBM [9]	39.2	26.1	-9.05	3.39	2.63	0.16
ALSTM [7]	44.2	21.2	-8.34	2.20	8.07	0.88
TFT [10]	31.4	32.0	-2.38	3.82	14.8	1.09
DTML [20]	33.6	32.7	**-1.76**	3.79	12.7	1.10
RSR [8]	34.5	32.6	-6.71	**2.98**	12.9	1.07
DQC-Plain (ours)	35.2	34.1	-6.22	3.32	18.1	1.19
DQC-Rank (ours)	**36.1**	**34.2**	-4.94	3.05	**21.4**	**1.26**

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 overal 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 4: ACL-18 and KDD-17 results with daily trading data.

ACL-18	QAcc (%) ↑	PAcc (%) ↑	MCC (%) ↑	MDD (%) ↓	Gain (%) ↑	SR ↑
LightGBM [9]	24.9	15.0	-41.4	1.96	-4.47	-0.10
ALSTM [7]	42.9	30.6	36.3	**1.07**	25.1	0.61
TFT [10]	41.0	39.2	37.8	1.17	37.2	1.74
DTML [20]	41.6	38.2	30.9	1.12	36.8	2.02
RSR [8]	41.1	39.7	37.8	1.11	35.9	2.21
DQC-Plain (ours)	41.3	39.7	36.8	1.18	38.9	1.82
DQC-Rank (ours)	**46.4**	**42.4**	**39.0**	1.09	**39.8**	**2.33**
KDD-17	QAcc (%) ↑	PAcc (%) ↑	MCC (%) ↑	MDD (%) ↓	Gain (%) ↑	SR ↑
LightGBM [9]	34.5	17.3	-38.6	2.62	3.57	0.09
ALSTM [7]	42.1	14.3	3.15	1.80	5.94	0.18
TFT [10]	34.9	31.1	10.9	2.61	18.4	1.34
DTML [20]	34.8	29.8	12.6	2.48	17.6	1.31
RSR [8]	34.8	33.7	**13.3**	2.50	19.9	1.47
DQC-Plain (ours)	32.6	32.4	12.5	2.55	18.7	1.50
DQC-Rank (ours)	30.7	**35.0**	10.6	**1.56**	**24.5**	**1.59**

- *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 v.s. 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.

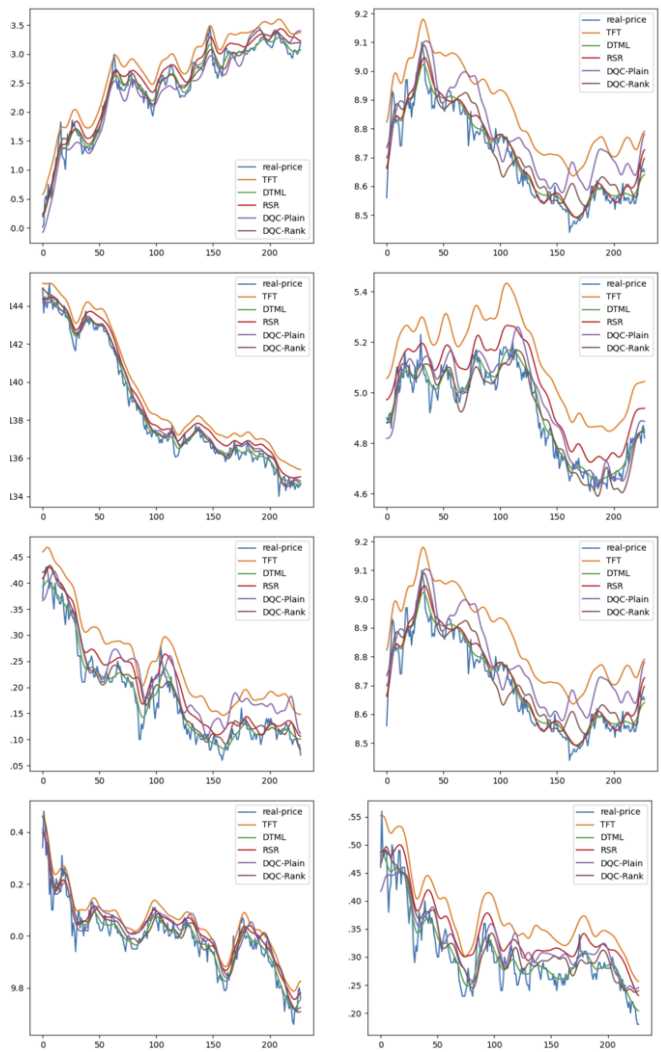

Figure 5: Visualization of distributional quantile prediction.

5.5 Ablation studies

Effects of Rank Learning (RKL). We study the individual contribution of L^{inter} in Eq. (10) and L^{intra} in Eq. (15). On CSI-21 dataset, DQC-Plain with L^{inter} and L^{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^{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

[1] Minna Castoe, 'Predicting stock market price direction with uncertainty using quantile regression forest', *Uppsala University*, (2020).

[2] Marco Cuturi, 'Sinkhorn distances: Lightspeed computation of optimal transport', in *NeurIPS*, pp. 2292–2300, (2013).

[3] Marco Cuturi, Olivier Teboul, and Jean-Philippe Vert, 'Differentiable ranking and sorting using optimal transport', in *NeurIPS*, (2019).

[4] Qianggang Ding, Sifan Wu, Hao Sun, Jiadong Guo, and Jian Guo, 'Hierarchical multi-scale gaussian transformer for stock movement prediction.', in *IJCAI*, (2020).

[5] Matthew Dixon, Diego Klabjan, and Jin Hoon Bang, 'Classification-based financial markets prediction using deep neural networks', *Algorithmic Finance*, (2019).

[6] Chenyou Fan, Heng Huang, et al., 'Multi-horizon time series forecasting with temporal attention learning', in *SIGKDD*, (2019).

[7] Fuli Feng, Huimin Chen, Xiangnan He, Ji Ding, Maosong Sun, and Tat-Seng Chua, 'Enhancing stock movement prediction with adversarial training', in *IJCAI*, (2019).

[8] Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua, 'Temporal relational ranking for stock prediction', *ACM Transactions on Information Systems (TOIS)*, (2019).

[9] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu, 'Lightgbm: A highly efficient gradient boosting decision tree', in *NeurIPS*, (2017).

[10] Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister, 'Temporal fusion transformers for interpretable multi-horizon time series forecasting', *International Journal of Forecasting*, (2021).

[11] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár, 'Focal loss for dense object detection', in *ICCV*, (2017).

[12] Ofir Pele and Michael Werman, 'Fast and robust earth mover's distances', in *Intl. Conf. on Computer Vision*, pp. 460–467, (2009).

[13] Xiaohang Ren, Kun Duan, Lizhu Tao, Yukun Shi, and Cheng Yan, 'Carbon prices forecasting in quantiles', *Energy Economics*, **108**, 105862, (2022).

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, 'Attention is all you need', *NIPS*, (2017).

[15] Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka, 'A multi-horizon quantile recurrent forecaster', in *arXiv preprint arXiv:1711.11053*, (2017).

[16] Haibin Xie, Jiangze Bian, Mingxi Wang, and Han Qiao, 'Is technical analysis informative in uk stock market? evidence from decomposition-based vector autoregressive (dvar) model', *Journal of systems science and complexity*, **27**(1), 144–156, (2014).

[17] Yujia Xie, Hanjun Dai, Minshuo Chen, Bo Dai, Tuo Zhao, Hongyuan Zha, Wei Wei, and Tomas Pfister, 'Differentiable top-k with optimal transport', in *NeurIPS*, (2020).

[18] Qifa Xu, Xi Liu, Cuixia Jiang, and Keming Yu, 'Quantile autoregression neural network model with applications to evaluating value at risk', *Applied Soft Computing*, (2016).

[19] Yumo Xu and Shay B Cohen, 'Stock movement prediction from tweets and historical prices', in *ACL*, (2018).

[20] Jaemin Yoo, Yejun Soun, Yong-chan Park, and U Kang, 'Accurate multivariate stock movement prediction via data-axis transformer with multi-level contexts', in *SIGKDD*, (2021).

[21] Liheng Zhang, Charu Aggarwal, and Guo-Jun Qi, 'Stock price prediction via discovering multi-frequency trading patterns', in *SIGKDD*, (2017).

[22] Zihao Zhang, Stefan Zohren, and Stephen Roberts, 'Deeplob: Deep cnns for limit order books', in *IEEE Transactions on Signal Processing*, (2018).