FreQuant: A Reinforcement-Learning based Adaptive Portfolio Optimization with Multi-frequency Decomposition

Jihyeong Jeon
Seoul National University
Seoul, South Korea
jeonjihyeong@snu.ac.kr

Jiwon Park
Seoul National University
DeepTrade Technologies Inc.
Seoul, South Korea
jiwon_p@snu.ac.kr

Chanhee Park
Seoul National University
DeepTrade Technologies Inc.
Seoul, South Korea
chan0843@snu.ac.kr

U Kang
Seoul National University
Seoul, South Korea
ukang@snu.ac.kr

ABSTRACT

How can we leverage inherent frequency features of stock signals for effective portfolio optimization? Portfolio optimization in the domain of finance revolves around strategically allocating assets to maximize returns. Recent advancements highlight the efficacy of deep learning and reinforcement learning (RL) in capturing temporal asset patterns for portfolio optimization. However, previous methodologies focusing on time-domain often fail to detect sudden market shifts and abrupt events because their models are overly tailored to prevalent patterns, resulting in significant losses.

In this paper, we propose FreQuant (Adaptive Portfolio Optimization via Multi-**Fre**quency **Quant**itative Analysis), an effective deep RL framework for portfolio optimization that fully operates in the frequency domain, tackling the limitations of time domain-focused models. By bringing the analysis into the frequency domain with the Discrete Fourier Transform, our framework captures both prominent and subtle market frequencies, enhancing its adaptability and stability in response to market shifts. This approach allows FreQuant to adeptly identify primary asset patterns while also effectively responding to less common and abrupt market events, providing a more accurate and comprehensive asset representation. Empirical validation on diverse real-world trading datasets underscores the remarkable performance of FreQuant, showing its superiority in terms of profitability. Notably, FreQuant achieves up to 2.1× higher Annualized Rate of Return and 2.9× higher Portfolio Value than the best-performing competitors.

KDD '24, August 25–29, 2024, Barcelona, Spain.

ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3671668

CCS CONCEPTS

• **Computing methodologies** → **Reinforcement learning**; • **Applied computing** → *Economics*; • **Information systems** → Data mining.

KEYWORDS

quantitative investment; reinforcement learning; portfolio optimization; Fourier transform; deep learning

ACM Reference Format:
Jihyeong Jeon, Jiwon Park, Chanhee Park, and U Kang. 2024. FreQuant: A Reinforcement-Learning based Adaptive Portfolio Optimization with Multi-frequency Decomposition. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3637528.3671668

1 INTRODUCTION

How can we leverage inherent frequency features of stock signals for effective portfolio optimization? Portfolio optimization is a crucial area of research that aims to identify the best portfolio allocation to maximize returns over a given period. The ability to make decisions based on complex market conditions enables investors to manage their portfolio effectively. As a result, there is growing demand for methods that prioritize robustness and superior returns. Recent studies have shown that Deep Learning [7, 15, 21, 22, 52], especially when combined with Reinforcement Learning (RL) [2, 20, 25, 40, 51], achieves effective portfolio optimization.

One of the primary challenges in this problem is how to model an asset's representation at a given trading moment. The dynamics of the market change over time, and comprise complex factors, making it difficult to accurately determine which factor contributes the most to an asset's behavior [5]. To address this issue, researchers have explored various approaches, such as accounting for correlational information between assets [26, 31, 43] or using attentive recurrent networks [10, 38, 49]. These methodologies analyze the complex signal of the asset in the time domain to identify implicit temporal patterns, which are then used in expressing individual assets.

However, these time domain-focused models often implicitly smooth over temporal variations and place excessive emphasis on prevalent patterns, favoring continuous trends over abrupt

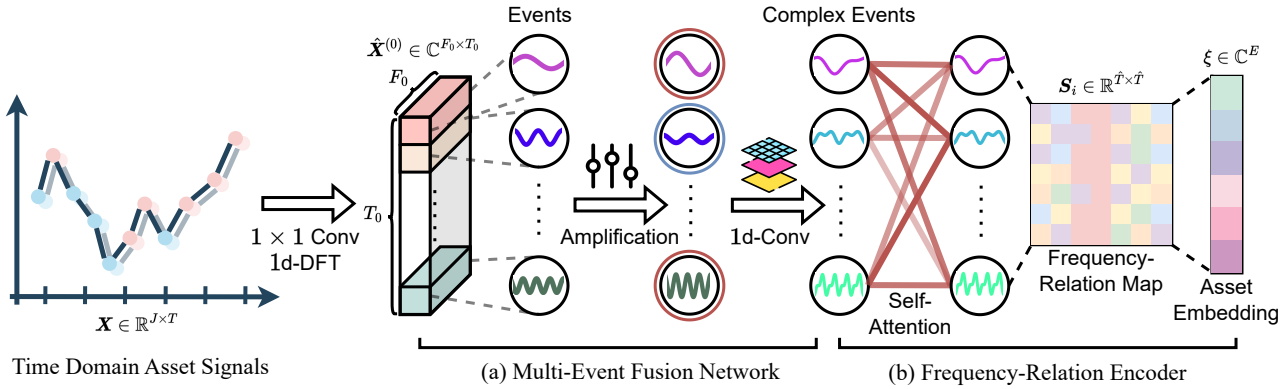


Figure 1: The Frequency State Encoder (FSE) in FreQuant is pivotal for multi-granular asset representation that captures both prevalent and abrupt patterns in asset signals. The FSE comprises two main components: (a) the Multi-Event Fusion Network which amplifies and merges frequency features to capture complex patterns, and (b) the Frequency-Relation Encoder which uses self-attention to interlink complex events and spotlight those essential for asset representation.

changes [16, 51]. This smoothing mechanism—such as RNNs' averaging operations or attention mechanisms' blending of temporal information—helps identify prevalent trends, but inadvertently diminishes the significance of abrupt changes, treating them as anomalies or outliers rather than signals. As a result, crucial market shifts can be overlooked, potentially leading to substantial investment losses. Therefore, the challenge lies in developing models that maintain sensitivity to both ongoing trends and sudden shifts.

To tackle this challenge, we shift our analysis to the frequency domain, enabling discernment of pervasive trends and abrupt deviations in asset signals. For this purpose, we use the Discrete Fourier Transform (DFT) [6, 27], which converts time domain signals into frequency domain features, reflecting the intensity and timing of periodic patterns. Notably, prevalent patterns correlate with lower-frequency features, while rapid shifts are captured by higher-frequency ones. This distinction aids the model's direct identification and handling of features tied to prevalent and sudden shifts.

We propose FreQuant (Adaptive Portfolio Optimization via Multi-**Fre**quency **Quant**itative Analysis), a novel deep RL (DRL) framework for adaptive portfolio optimization managing both prevalent and abrupt patterns in the signals, thereby enhancing stability and responsiveness to market changes. Operating fully in the frequency domain, FreQuant retains critical frequency attributes, directly linking to the intensity and timing of periodic patterns.

For optimal portfolio construction, asset signals are processed through FreQuant's modules: the frequency state encoder and the portfolio generator. The frequency state encoder decomposes temporal patterns, learns to amplify and compose complex patterns, and uses the self-attention mechanism to highlight key frequencies, thereby creating multi-granular asset embeddings. Figure 1 depicts this process for a single asset. The portfolio generator then integrates these embeddings to produce real-valued portfolio weights, effectively capturing the interrelationships among the assets.

To implement our framework, we use the deep deterministic policy gradient [23], with effective guidance for stabilizing optimization process and an explicit formulation of transaction fee in long-short portfolio settings. Experimental results demonstrate the effectiveness of our model in capturing market shifts and optimizing portfolio returns, achieving the state-of-the-art performance.

Our Contributions are summarized as follows:

- We propose FreQuant, a novel DRL framework that leverages multi-frequency features to adapt to sudden market shifts by fully operating in the frequency domain while preserving the essential information of these features.
- We present a precise formulation of transaction fees, a crucial factor in long-short trading markets, to enhance the real-world relevance of our model.
- Through extensive experiments conducted on diverse market datasets, we demonstrate the superior performance of FreQuant compared to other existing methods. Our framework achieves up to 2× higher Annualized Rate of Return, making it a promising approach for portfolio optimization tasks.

2 RELATED WORKS

Asset Movement Prediction Strategies. Recent advancements in deep neural networks have led to significant progress in resolving asset movement prediction problem [18, 39–42, 49, 50]. Utilizing attention mechanisms [38] to capture long-range dependencies, several studies have uncovered temporal patterns from assets signals [33, 41, 49]. Additionally, the integration of external datasets, such as expert trading signals [42] and high-frequency signals [18], has been explored. Zhang et al. [50] employed DFT to extract multi-frequency features within recurrent units. However, these techniques often fall short in real-trading scenarios as they lack access to essential information, such as adjusted rewards with transaction fees or adjusted portfolio weights for the dynamic allocation.

Deep RL for Portfolio Optimization. The fusion of deep neural networks and reinforcement learning (RL), known as deep RL, has demonstrated promise in portfolio optimization [20, 26, 31, 35, 36, 43, 44, 46, 47]. For example, the EIIE framework [20] introduced the first RL-based portfolio optimization with temporal pattern modeling using Temporal Convolutional Networks (TCNs) [4]. Other deep RL-based methods have integrated attention mechanisms to enhance explainability [31] and incorporate asset correlations [43, 45]. Additional efforts have leveraged external resources such as high-frequency data [36, 44, 46], expert trading insights

[26], or news articles [47] to enrich context embeddings. Nevertheless, the complexity of capturing sudden and abrupt events from intricate asset signals remains a challenge for these methodologies.

3 PROBLEM FORMULATION

We give a formal definition of the portfolio optimization problem in a long-short setting as follows:

Problem 1 (Portfolio Optimization with Long-Short). ***Given:*** *historical price data $\mathcal{X}$ of N assets and the market index,* ***Find:*** *the optimal long-short weights $\boldsymbol{v}_t \in \mathbb{R}^{N+1}$ for cash and each individual asset at every rebalancing time-step t, aiming to maximize the profit over the trading period through these investments.*

This problem's natural alignment with the Markov decision process environment is concisely detailed in the following.

Time-steps. The trading period is defined as $[P_s^{trade}, P_e^{trade})$, where P_s^{trade} and P_e^{trade} are the starting and ending dates of trading period, respectively. Each rebalancing time step t is an index of an opened market day between these dates. Formally, t belongs to the set $\{1, 2, \ldots, \lfloor \frac{|D \setminus C|-1}{r_f} \rfloor\}$, where $r_f \in \mathbb{Z}^+$ represents the rebalancing frequency, $D = \{P_s^{trade}, \ldots, P_e^{trade}\}$ is the set of calendar dates during the trading period, and $C = \{x \in D : x \text{ is a closed market date}\}$ is the set of closed market dates.

State Space. The state s_t at a rebalancing time-step t, is composed of assets' price feature tensor $\boldsymbol{X}_t^{\text{Asset}} \in \mathbb{R}^{J \times N \times T}$, corresponding to market's price feature tensor $\boldsymbol{X}_t^{\text{Market}} \in \mathbb{R}^{J \times 1 \times T}$, and the current portfolio weights $\boldsymbol{w}'_t \in \mathbb{R}^{N+1}$, where J is the feature dimension, and $T \in \mathbb{Z}^+$ is the number of look-up days. A feature tensor $\boldsymbol{X}_t = [\boldsymbol{I}_{t-T}; \ldots; \boldsymbol{I}_{t-1}]$ consists of daily price feature matrix $\boldsymbol{I}_t = [\boldsymbol{f}_{t,1}; \ldots; \boldsymbol{f}_{t,N}] \in \mathbb{R}^{J \times N}$, where $\boldsymbol{f}_{t,i} = [f_{t,i}^{(1)}, \ldots, f_{t,i}^{(J)}]^T \in \mathbb{R}^J$ is a feature vector of i-th asset.

Action Space. An optimal long-short weight vector or the desired portfolio weights $\boldsymbol{v}_t = [v_{t,0}, \ldots, v_{t,N}]^T \in \mathbb{R}^{N+1}$ at rebalancing time-step t, is $N+1$ dimensional vector, where the first component $v_{t,0} \geq 0$ is cash-weight and $-\mathbb{1}_N \leq \boldsymbol{v}_{t,\setminus 0} \leq \mathbb{1}_N$ are individual assets' long-short weights. The magnitude $|v_{t,i}|$ of the i-th asset weight represents the investment ratio relative to the total budget, giving the equality constraint $\sum_{i=0}^{N} |v_{t,i}| = 1$. A non-negative weight $v_{t,i} \geq 0$ for the i-th asset indicates a *long* position, while a negative weight $v_{t,i} < 0$ implies a *short* position.

Reward Function. We assume the first and second row vectors of $\boldsymbol{I}_t$, denoted as $\boldsymbol{p}_1(\boldsymbol{I}_t)$ and $\boldsymbol{p}_2(\boldsymbol{I}_t)$, respectively, to be the daily opening and highest prices, each in $\mathbb{R}^N$. The change rate $\boldsymbol{\delta}_t^o \in \mathbb{R}^N$ of the opening price of the assets during the rebalancing time-step t and $t+1$ is defined with element-wise operations as follows:

$$\boldsymbol{\delta}_t^o = (\boldsymbol{p}_1(\boldsymbol{I}_{t+1}) - \boldsymbol{p}_1(\boldsymbol{I}_t)) \oslash \boldsymbol{p}_1(\boldsymbol{I}_t), \tag{1}$$

where $\oslash$ denotes Hadamard division. Similarly, the change rate $\boldsymbol{\delta}_t^h \in \mathbb{R}^N$ of highest price with respect to the opening price during rebalancing time-step t and $t+1$ is defined as follows:

$$\boldsymbol{\delta}_t^h = (\boldsymbol{p}_2(\boldsymbol{I}_t) - \boldsymbol{p}_1(\boldsymbol{I}_t)) \oslash \boldsymbol{p}_1(\boldsymbol{I}_t). \tag{2}$$

As asset prices fluctuate throughout the rebalancing period, the weights allocated to individual assets at t differ from those at $t+1$. For assets in a long position, the difference in weight from t to $t+1$ is calculated by multiplying the weight at t by the change rate of the opening price, $\boldsymbol{\delta}_t^o$. Conversely, for assets in a short position, considering a "Margin Call" scenario where the position is forcibly closed if the price doubles $\boldsymbol{\delta}_t^h \geq \mathbb{1}_N$, the maximum loss is capped at the initial invested weight. The weight difference for assets, $\boldsymbol{d}_t \in \mathbb{R}^N$, between rebalancing time-step at t and $t+1$ becomes:

$$\boldsymbol{d}_t = (\boldsymbol{v}_{t,\setminus 0})^+ \odot \boldsymbol{\delta}_t^o + (\boldsymbol{v}_{t,\setminus 0})^- \odot \boldsymbol{\delta}_t^m, \tag{3}$$

where $\odot$ is Hadamard product, $x^+ = \max(x, 0)$ and $x^- = -\min(x, 0)$ are the positive and negative part of x, respectively, and $\delta_{t,i}^m = 1$ if $\delta_{t,i}^h \geq 1$, otherwise $\delta_{t,i}^o$ for $i = 1, \cdots, N$. Then the changed portfolio weights $\boldsymbol{w}_{t+1} \in \mathbb{R}^{N+1}$ at the rebalancing time-step $t+1$ turns into:

$$\boldsymbol{w}_{t+1} = [v_{t,0}, \boldsymbol{v}_{t,\setminus 0} + \boldsymbol{d}_t]. \tag{4}$$

Accordingly, the normalized portfolio weights $\boldsymbol{w}'_{t+1} \in \mathbb{R}^{N+1}$ at the rebalancing time-step $t+1$ with L^1-norm becomes:

$$\boldsymbol{w}'_{t+1} = \frac{\boldsymbol{w}_{t+1}}{||\boldsymbol{w}_{t+1}||_1}. \tag{5}$$

Then we define the reward function $r_t \in \mathbb{R}$, which evaluates the portfolio reallocation based on the value v_t at the subsequent rebalancing time step $t+1$. Specifically, the reward function r_t can be formulated as the combination of 1) total profit and loss, and 2) the total paid transaction fee during the reallocation as follows:

$$r_t = \underbrace{\boldsymbol{d}_t^T \operatorname{sign}(\boldsymbol{v}_{t,\setminus 0})}_{\text{Total P\&L Amount}} - \underbrace{g(\boldsymbol{w}'_t, \boldsymbol{v}_t, f_l, f_s)}_{\text{Total Paid Transaction Fee}}, \tag{6}$$

where the function g computes the fee for the reallocation of portfolio weights at time step t and $0 \leq f_l, f_s < 1$ represent the transaction fee rates for closing long and short positions, respectively. We further elaborate the calculation in the upcoming section.

Transaction Fee. In line with previous works that formulated transaction fees for long position settings [20], we extend the formulation to incorporate both long and short positions for assets. In this setting, traders have the flexibility to take either a long or short position for any given asset. In the rest part of this section, we omit the rebalancing time-step t for the simplicity.

Let c_i and o_i denote closing and opening position weights for the i-th asset. Specifically, $c_i \geq 0$ corresponds to closing a long position, while $c_i < 0$ indicates a short position closing. Similarly, $o_i \geq 0$ denotes opening a long position, and $o_i < 0$ indicates an opening short position. Transaction fees are typically incurred only when closing positions, not when opening new ones. The desired weight v_i for an asset needs to be adjusted after closing and opening positions and normalizing for transaction fees, expressed as:

$$v_i = \frac{w'_i - c_i + o_i}{1 - \Theta} \qquad \forall i = 0, \ldots, N, \tag{7}$$

where $\Theta = \sum_{i=1}^{N} c_i^+ f_l + c_i^- f_s \in \mathbb{R}$ is the total paid transaction fee.

To solve Eq. (7), we employ a fixed-point iteration method denoted as $g(\boldsymbol{w}'_t, \boldsymbol{v}_t, f_l, f_s)$, iteratively solving c_i and o_i to compute Θ. Specifically, we alternate between fixing o_i and solving for $c_i = v_i(\Theta - 1) + w'_i + o_i$, and fixing c_i and solving for $o_i = v_i(1 - \Theta) - w'_i + c_i$. This iterative approach converges to c_i^*, o_i^*, and Θ^* while adhering to the constraints, including i) $w'_i c_i \geq 0$ to maintain consistent closing position directions with the current position, ii) $|c_i| \leq |w'_i|$

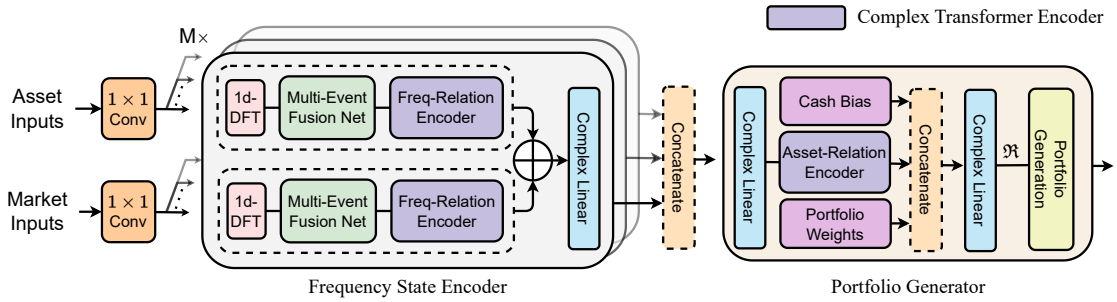


Figure 2: An overview of the proposed architecture FreQuant.

Algorithm 1: Fixed-point Iteration for Θ^*

Input: $\boldsymbol{v}$: The desired portfolio weight vector.
$\boldsymbol{w}$: The current portfolio weight vector.
f_l, f_s: Transaction fee for closing long and short pos.
Output: Total paid transaction fee Θ^* for the reallocation.

$k \leftarrow 0$, $\boldsymbol{c}^{(0)} \leftarrow \mathbf{0}$, $\boldsymbol{o}^{(0)} \leftarrow \mathbf{0}$, $\Theta \leftarrow 0$, $\boldsymbol{c}_{prev} \leftarrow \mathbb{1}_{N+1}$
while $||\boldsymbol{c}^{(k)} - \boldsymbol{c}_{prev}||_2^2 \geq \epsilon$ **do**
  $\boldsymbol{c}^{(k+1)} \leftarrow \boldsymbol{v} \odot \mathbb{1}_{N+1}(\Theta - 1) + \boldsymbol{w} + \boldsymbol{o}^{(k)}$
  if $c_i^{(k+1)} w_i < 0$ **then**
    $c_i^{(k+1)} \leftarrow 0$ /* The first constraint */
  end
  if $|c_i^{(k+1)}| \geq |w_i|$ **then**
    $c_i^{(k+1)} \leftarrow w_i$ /* The second constraint */
  end
  $\Theta \leftarrow \sum_{i=1}^{N} (c_i^{(k+1)})^+ f_l + (c_i^{(k+1)})^- f_s$
  $\boldsymbol{o}^{(k+1)} \leftarrow \boldsymbol{v} \odot \mathbb{1}_{N+1}(1 - \Theta) - \boldsymbol{w} + \boldsymbol{c}^{(k+1)}$
  if $(w_i - c_i^{(k+1)}) o_i^{(k+1)} < 0$ **then**
    $o_i^{(k+1)} \leftarrow 0$ /* The third constraint */
  end
  $\boldsymbol{c}_{prev} \leftarrow \boldsymbol{c}^{(k+1)}$
  $k \leftarrow k + 1$
end
$\Theta^* \leftarrow \Theta$
return Θ^*

to cap closing positions, and iii) $(w'_i - c_i)o_i \geq 0$ to preserve position directionality after opening. The Algorithm 1 illustrates the pseudo-algorithm for computing the transaction fee $g(\boldsymbol{w}'_t, \boldsymbol{v}_t, f_l, f_s)$.

4 PROPOSED METHOD

In this section, we introduce FreQuant, a novel deep reinforcement learning-based architecture for portfolio optimization, operating fully in the frequency domain. We address the following challenges:

C1 **Multi-Granular Asset Representation.** The granular model, overly attuned to prevalent market patterns, struggles to detect the subtleties of fine-grained patterns, such as abrupt, yet significant market shifts. *How can we embed both the granular and the fine-grained patterns within the asset representation?*

C2 **Portfolio Generation with Asset-Correlation.** Modeling the evolving interrelationships among assets for portfolio generation aids in identifying the condition of noisy assets through comparisons and managing risk by diversifying the portfolio. *How can we model asset correlations during portfolio generation?*

C3 **Stabilizing the Optimization Process.** Deep reinforcement learning faces instability in training actor-critic networks with numerous parameters [9, 12, 24], inevitably requiring an additional tactic for stabilization. *How can we enhance stability during the optimization process?*

To tackle these challenges, we introduce an end-to-end framework that operates exclusively in the frequency domain. This approach focuses on extracting meaningful insights from frequency features and adapting to sudden market changes. Figure 2 provides an overview of the FreQuant framework, addressing each challenge with the following main ideas:

I1 **Frequency State Encoder (Section 4.1).** We bring our analysis into the frequency domain with discrete Fourier transform to explicitly disclose prevalent and abrupt patterns. These patterns are effectively fused to capture complex patterns with event-filters and convolution kernels. The Frequency-Relation Encoder, an extension of the Transformer [38], is used to emphasize crucial frequency features and adapt to market shifts.

I2 **Portfolio Generator (Section 4.2).** We capture the underlying asymmetric correlations among the assets using the self-attention mechanism. The context embeddings of each asset, with these correlations, are utilized to compute confidence scores that determine their inclusion in the final portfolio.

I3 **Optimization with Guidance (Section 4.3).** We utilize the DDPG algorithm during the optimization process and guide the event-filter parameters to learn predefined fundamental periodicities, thereby enhancing the process's overall stability.

4.1 Frequency State Encoder

The main objective of the Frequency State Encoder (FSE) is to generate multi-granular asset representations that effectively capture both prevalent and abrupt patterns within respective asset signals. To achieve this, we focus our analysis in the frequency domain, employing DFT to explicitly identify both prevailing and rare patterns within asset signals. We then utilize complex-valued operations to further elaborate asset representation while preserving the characteristics of distinguished patterns in frequency domain. Lastly, we model intricate interrelationships among these patterns to eliminate the redundancies and adaptively identify the pivotal pattern. We omit the rebalancing time-step t from this discussion.

Scrutinizing Latest Patterns. Initially, we emphasize key price signals, while diversifying representation through applying 1×1 convolution [37] along the feature dimension of the asset price

feature tensor $\boldsymbol{X}^{\text{Asset}} \in \mathbb{R}^{J\times N\times T}$ with J price features, N assets, and T window size. This produces the refined input tensor $\tilde{\boldsymbol{X}}^{\text{Asset}} \in \mathbb{R}^{F_0\times N\times T}$, where F_0 denotes the number of feature maps.

We target our analysis on the latest patterns by deploying multiple FSE blocks with temporally segmented input tensors, as these patterns tend to contribute more to price signals. Specifically, for the m-th FSE block, we have the respective m-th segment $\boldsymbol{X}_m^{\text{Asset}} \in \mathbb{R}^{F_0\times N\times \bar{T}_m}$ of the refined input tensor, with the latest $\bar{T}_m = \lfloor \frac{T}{2^{m-1}} \rfloor$ points, for $m = 1, \ldots, M$. For the brevity, we refer to a single input tensor as $\boldsymbol{X}$ and segmented dimension as $\bar{T}$ in the following.

Multi-frequency Decomposition. For the explicit disclosure of periodic patterns within the signals, we bring our analysis into the frequency domain by applying a 1-dimensional real-valued $\bar{T}$-point DFT along the temporal dimension of the input tensor $\boldsymbol{X} \in \mathbb{R}^{F_0\times N\times \bar{T}}$, and obtain complex feature tensor $\hat{\boldsymbol{X}}^{(0)} \in \mathbb{C}^{F_0\times N\times T_0}$, where $T_0 = \lfloor \frac{\bar{T}}{2} \rfloor + 1$ is the truncated temporal dimension. Specifically, for the finite sequence of $\bar{T}$ real numbers of i-th asset's f-th feature map $\boldsymbol{X}^{i,f} \in \mathbb{R}^{\bar{T}}$, we have a Hermitian-symmetric finite complex sequence $\tilde{\boldsymbol{X}}^{i,f} \in \mathbb{C}^{\bar{T}}$ defined as $\tilde{\boldsymbol{X}}_k^{i,f} = \sum_{n=0}^{\bar{T}-1} \boldsymbol{X}_n^{i,f} \cdot e^{-\frac{I2\pi k}{\bar{T}} n}$, where $I = \sqrt{-1}$ and the corresponding frequency is $\frac{k}{\bar{T}} \times$ Sampling Rate. For instance, if we have a daily price sequence and $\bar{T} = 256$, then the 8-th frequency feature $\tilde{\boldsymbol{X}}_8^{i,f}$ corresponds to the amplitude and phase information of the frequency $\frac{1}{32}$cycle/day, which is the pattern that repeats every 32 days. We define each of these frequency feature vectors $\tilde{\boldsymbol{X}}_k^{i} \in \mathbb{C}^{F_0}$ as the k-th periodic "*Event*" of the i-th asset, where each component $\tilde{\boldsymbol{X}}_k^{i,f}$ depicts the characteristics of its event. These events characterize the patterns like cyclical factors, business cycles, seasonal effects, or other periodic influences [17]. We retain only half of $\tilde{\boldsymbol{X}}$ by virtue of the Hermitian-symmetry, resulting in a complex tensor $\hat{\boldsymbol{X}}^{(0)} \in \mathbb{C}^{F_0\times N\times T_0}$, where $T_0 = \lfloor \frac{\bar{T}}{2} \rfloor + 1$.

Multi-Event Fusion Network. The goal of the multi-event fusion network is to filter important frequency features out of the complex tensor $\hat{\boldsymbol{X}}^{(0)}$ and capture the *complex event*, which defies simple sinusoidal representation and appears through a mixture of multi-events. For this purpose, we employ a series of units $i = 1, \ldots, I$, each consisting of an event-filtering layer and a 1d-convolutional layer, to signify crucial frequency features and discover fusion rules for complex events, respectively.

Specifically, we apply the u-th event-filtering of the i-th unit with parameters $\boldsymbol{M}_u^{(i)} \in \mathbb{C}^{F_{i-1}\times N\times T_{i-1}}$, where parameters across the asset dimension are shared, $u = 1, \ldots, u_i$ represents the index of the event-filter within the i-th unit, and F_{i-1} and T_{i-1} denote the number of feature maps and the temporal dimension at the $(i-1)$-th unit, respectively. Each u-th event-filter aims to selectively preserve the most relevant and informative features, resulting in a more refined and meaningful representation. Collectively, concatenating along the feature dimension of all event-filtered outputs $\overline{\boldsymbol{X}}^{(i)} \in \mathbb{C}^{u_i F_{i-1}\times N\times T_{i-1}}$ can be expressed as follows:

$$\overline{\boldsymbol{X}}^{(i)} = \text{Concat}(\boldsymbol{M}_1^{(i)} \odot \hat{\boldsymbol{X}}^{(i-1)}, \ldots, \boldsymbol{M}_{u_i}^{(i)} \odot \hat{\boldsymbol{X}}^{(i-1)}). \tag{8}$$

Subsequently, the 1d-convolution $\hat{\boldsymbol{X}}^{(i)} \in \mathbb{C}^{F_i\times N\times T_i}$ of i-th unit along the temporal dimension for each asset with the kernel size k_i, stride size s_i, and F_i filters is:

$$\hat{\boldsymbol{X}}^{(i)} = \text{Conv1d}(\overline{\boldsymbol{X}}^{(i)}; k_i, s_i, F_i), \tag{9}$$

where $\text{Conv1d}(\boldsymbol{X})$ is the concatenated result of independently applied 1d-convolution operations for each asset of the 3d-input tensor $\boldsymbol{X}$ with the same parameters, and $T_i = \lfloor \frac{T_{i-1}-k_i}{s_i} + 1 \rfloor$ is the changed temporal dimension. Consequently, 1d-convolution of the last unit outputs a complex feature tensor $\hat{\boldsymbol{X}} \in \mathbb{C}^{\hat{F}\times N\times \hat{T}}$, where $\hat{F}$ and $\hat{T}$ denotes the number of filters and temporal dimension at the last 1d-convolution, respectively.

Frequency-Relation Encoder. We further enhance asset representation by modeling the extant relationships among the captured events [8], which aids in eliminating redundancies and spotting pivotal events. To this end, we adapt the Transformer encoder [38] into a Complex Transformer Encoder (CTE), leveraging the self-attention mechanism's proficiency in delineating these relationships. To function fully in a complex vector space, we employ the Hermitian inner product in place of the dot product to compute the similarity between two complex vectors, and use complex softmax to normalize and retrieve the attention scores of these vectors. In the following, we refer to CTE as either the Frequency-Relation Encoder (FRE) when modeling events' correlations or the Asset-Relation Encoder (ARE) when designing assets' interdependencies.

To exemplify within the FRE context, consider a permuted complex feature tensor $\hat{X} \in \mathbb{C}^{N\times \hat{T}\times \hat{F}}$, with $\boldsymbol{W}^Q, \boldsymbol{W}^K, \boldsymbol{W}^V \in \mathbb{C}^{\hat{F}\times d_k}$ as parameters for the complex linear projection [14] of query, key, and value into d_k-dimensional spaces, respectively. The asset dimension N is treated as the mini-batch dimension. The attention scores $\boldsymbol{S}_i \in \mathbb{R}^{\hat{T}\times \hat{T}}$ for the i-th asset across their $\hat{T}$ complex events are:

$$\boldsymbol{S}_i = \text{cSoftmax}\left(\frac{(\hat{\boldsymbol{X}}_i \boldsymbol{W}^Q)(\hat{\boldsymbol{X}}_i \boldsymbol{W}^K)^*}{\sqrt{d_k}}\right), \tag{10}$$

where $\hat{\boldsymbol{X}}_i \in \mathbb{R}^{\hat{T}\times \hat{F}}$ is the i-th asset's feature matrix, $(\cdot)^*$ denotes the Hermitian transpose, $\sqrt{d_k}$ is a scaling factor, and $\text{cSoftmax}(\boldsymbol{z}) = e^{|\boldsymbol{z}|}/\sum_i e^{|z_i|}$ is a row-wise complex softmax function adapted from [30].

The correlated embedding $\boldsymbol{X}^c = \boldsymbol{S}(\hat{\boldsymbol{X}}\boldsymbol{W}^V) \in \mathbb{C}^{N\times \hat{T}\times d_k}$ is an aggregation of linearly projected values, weighted by asymmetric correlation scores between events. The correlated embedding is then passed through a Feed Forward Network [14] and a layer normalization layer [3] with a residual connection [13] to match the input feature dimension $\hat{F}$. We denote the output embedding of FRE as $\xi = \text{FRE}(\hat{\boldsymbol{X}}) \in \mathbb{C}^{N\times E}$, where $E = \hat{T} \times \hat{F}$.

Final Asset Representation. The output embeddings, $\xi^{\text{Asset}} \in \mathbb{C}^{N\times E}$ for asset inputs and $\xi^{\text{Market}} \in \mathbb{C}^{1\times E}$ for market inputs, are obtained using their respective networks with different parameterization (see Figure 2). To consider both local and global semantics, we sum them in an element-wise manner, following [48, 49]:

$$\xi^O = \xi^{\text{Asset}} + \mathbb{1}_N \xi^{\text{Market}}. \tag{11}$$

Finally, a context embedding $\boldsymbol{C} \in \mathbb{C}^{N\times d_c}$, which is the representation of the assets, is achieved by concatenating linearly projected embeddings of each FSE outputs along the feature dimension as:

$$\boldsymbol{C} = \text{Concat}(\xi_1^O \boldsymbol{W}_1, \ldots, \xi_M^O \boldsymbol{W}_M), \tag{12}$$

where $\boldsymbol{W}_m \in \mathbb{C}^{E_m\times d_l}$ are complex parameters, E_m is the output dimension of the m-th FSE block, and $d_c = M \times d_l$ is the dimension after concatenating M number of d_l features.

4.2 Portfolio Generator

The central focus of the portfolio generator is to generate an optimal portfolio $\boldsymbol{v} \in \mathbb{R}^{N+1}$ with the attained asset context embeddings $\boldsymbol{C} \in \mathbb{C}^{N \times d_c}$. This optimal portfolio must meet two crucial requirements for high-quality investment: stability and profitability. To achieve these objectives, we model the evolving interdependencies among the assets [26, 43, 49] to enhance stability, and select a portion of assets with the highest confidence scores to satisfy both criteria.

Asset-Relation Encoder. We embed the assets' interdependencies through the Asset-Relation Encoder (ARE), which mirrors the structure of the CTE, with a specific focus on creating asset-correlated context embeddings $\hat{\boldsymbol{H}}$. Specifically, it takes asset context embeddings $\boldsymbol{C} \in \mathbb{C}^{N \times d_c}$ and produces asset-correlated context embeddings $\hat{\boldsymbol{H}} = \text{ARE}(\boldsymbol{C}\boldsymbol{W}_c) \in \mathbb{C}^{N \times d_c}$, where $\boldsymbol{W}_c \in \mathbb{C}^{d_c \times d_c}$ is the linear projection weights, which serves as integrating the various levels of representations obtained from the multiple FSE blocks.

Portfolio Generation. In real-world trading scenarios, the number of assets held is often limited due to practical considerations, making asset selection crucial. However, previous strategies [26, 43, 45] fall short in markets with clear trends, such as consistent bull markets, because they maintain a fixed number of short positions regardless of market conditions. To address this, we construct the optimal portfolio $\boldsymbol{v}$, based on the highest magnitude of confidence scores, offering a targeted response to the market.

Specifically, we define the confidence scores $\boldsymbol{h} \in \mathbb{R}^{N+1}$ of the asset with the asset-correlated context embeddings $\hat{\boldsymbol{H}}$, the cash bias $\boldsymbol{b}_c \in \mathbb{C}^{d_c}$, and the current portfolio weights $\boldsymbol{w'} \in \mathbb{R}^{N+1}$ as:

$$\boldsymbol{h} = \Re(\text{Concat}_2(\text{Concat}_1(\boldsymbol{b}_c^T, \hat{\boldsymbol{H}}), \boldsymbol{w'})\boldsymbol{u} + \mathbb{1}_{N+1} b_u), \tag{13}$$

where $\text{Concat}_n(\cdot)$ denotes concatenation along the n-th dimension, $\Re(\cdot)$ is the real part, and $\boldsymbol{u} \in \mathbb{C}^{d_c+1}$ and $b_u \in \mathbb{C}$ are the weight and bias, respectively. Subsequently, let $\mathcal{G} \subset \{0, \ldots, N\}$ be the set of indices corresponding to the assets with the $|\mathcal{G}|$ highest magnitudes of the confidence scores. Then, the optimal portfolio weights $\boldsymbol{v} = [v_0, \ldots, v_N] \in \mathbb{R}^{N+1}$ is defined as

$$v_i = \begin{cases} \frac{h_i}{\sum_{j \in \mathcal{G}} |h_j|} & \text{if } i \in \mathcal{G} \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

4.3 Optimization with Guidance

We use the deep deterministic policy gradient [23] algorithm for optimization, but it presents instability issues, particularly with numerous parameters to optimize. To mitigate this, we propose a novel regularization method that guides the event-filter parameters using the inductive prior of fundamental periodicities.

Optimization Process. For every rebalancing time-step t, the state $s_t \in \mathcal{S} \subset \mathcal{X} \times \mathcal{W}$ is fed into the parametrized actor network $\mu(s|\theta^\mu)$ to produce an action $a := \boldsymbol{v} = \mu(s|\theta^\mu) \in \mathbb{R}^{N+1}$, where $\mathcal{S}$ is the state space and $\mathcal{W}$ is the portfolio weight space. After the execution of an action a_t, the reward r_t and the next state s_{t+1} are observed. The critic network Q computes a q-value as follows:

$$Q(s, a|\theta^Q) = \boldsymbol{V}_o^T(\boldsymbol{W}_Q \cdot \text{Concat}(\Phi(s|\theta^s), a) + \boldsymbol{b})^+ \in \mathbb{R}, \tag{15}$$

where $(\cdot)^+$ denotes the ReLU activation function, $\Phi(s|\theta^s) \in \mathbb{R}^{N+1}$ is a state encoding network which resembles the actor network $\mu(\cdot)$ but parametrized with θ^s, and $\boldsymbol{V}_o \in \mathbb{R}^{d_q}$, $\boldsymbol{W}_Q \in \mathbb{R}^{d_q \times 2(N+1)}$, and $\boldsymbol{b} \in \mathbb{R}^{d_q}$ are the learnable parameters.

The weights of the critic network θ^Q is updated with the gradients $\nabla_{\theta^Q}\mathcal{L} = \frac{1}{B}\nabla_{\theta^Q}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$ to minimize the temporal difference, where B is the mini-batch size and y_i is the i-th sample target. The weights of the actor network is updated with the action-value gradients [32] to maximize the performance objective:

$$\nabla_{\theta^\mu}\mathcal{J} \approx \frac{1}{B}\sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}. \tag{16}$$

Guiding with Fundamental Periodicities. To further stabilize the optimization process, we offer intuitive guidance for event-filter parameters by incorporating fundamental periodicities. This approach is inspired by the observation that market assets exhibit basic, fundamental periodicities, such as common seasonality effects (e.g., the Friday effect) [1, 11, 29]. Thus, we prompt the frequency features to learn these inductive priors of fundamental periodicities, serving as a stabilizing mechanism for the optimization process.

Recall that the output of the Multi-frequency Decomposition is the complex tensor $\hat{\boldsymbol{X}}^{(0)} \in \mathbb{C}^{F_0 \times N \times T_0}$, and the first unit's very first event-filter is $\tilde{\boldsymbol{M}} = \boldsymbol{M}_1^{(1)} \in \mathbb{C}^{F_0 \times N \times T_0}$, where all N assets share the same $F_0 \times T_0$ matrix. We define the magnitude of the i-th asset's f-th feature vector of the event-filter $\tilde{\boldsymbol{M}}^{i,f} \in \mathbb{C}^{T_0}$ as $\boldsymbol{\rho}_f = |\tilde{\boldsymbol{M}}^{i,f}|$. To provide guidance, we regularize $\boldsymbol{\rho}_f$ so that the parameters of the event-filter gradually follow and represent the predefined periodicities $\eta_1 = 5, \eta_2 = 10, \eta_3 = 20, \ldots$, where $|\eta| \leq F_0$. Since naturally existing patterns with length $\bar{T}$ that repeat every η_f days in the time domain will have high magnitudes at every $\lfloor \frac{\bar{T}}{\eta_f} \rfloor$-th index in the frequency domain, we aim to make the magnitude of the event-filter $\boldsymbol{\rho}_f$ mimic these η_f-periodic patterns. Formally, we jointly minimize the following regularization term:

$$\mathcal{L}_{reg} = \sum_{f=1}^{|\eta|} \|\boldsymbol{\rho}_f \odot \boldsymbol{\alpha}_f\|_2^2, \tag{17}$$

where $\boldsymbol{\alpha}_f = \left\{\Re\left(\frac{-1+\exp(I2\pi j\eta_f/\bar{T})}{-2+\exp(I2\pi j\eta_f/\bar{T})}\right)\right\}_{j=0}^{T_0-1}$ is the guide of the f-th feature vector with zero values only near the every $\lfloor \frac{\bar{T}}{\eta_f} \rfloor$-th index.

As a result, we update the actor network $\mu(s|\theta^\mu)$ with the compositive gradients $\nabla_{\theta^\mu}\mathcal{J}_{reg}$ using gradient ascent:

$$\nabla_{\theta^\mu}\mathcal{J}_{reg} = \nabla_{\theta^\mu}\mathcal{J} - \lambda\nabla_{\theta^\mu}\mathcal{L}_{reg}, \tag{18}$$

where λ is a hyperparameter balancing regularization intensity.

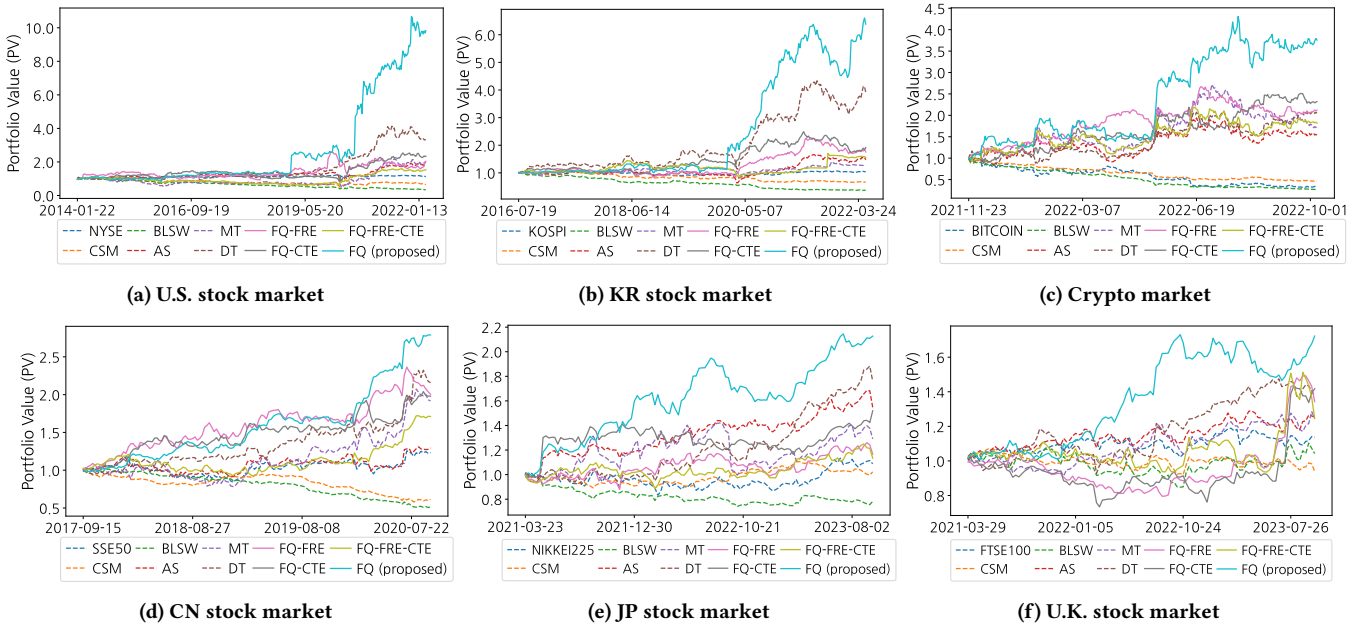
5 EXPERIMENTS

We perform experiments to address the following questions:

- Q1 **Performance Evaluation (Section 5.2).** Does FreQuant outperform other methods, demonstrating its effectiveness in portfolio optimization?
- Q2 **Effects of Frequency-Relation Encoder (Section 5.3).** Does the Frequency-Relation Encoder dynamically emphasize significant frequency features as time progresses?
- Q3 **Interpretation of Event-filters (Section 5.4).** Do the acquired event-filters from the Multi-Events Fusion Network accurately capture event-relevant patterns?
- Q4 **Empirical Validation of the Robustness (Section 5.5).** Does FreQuant have enhanced stability and responsiveness to the rapidly changing market?

Table 1: The results of the proposed method FreQuant (FQ) and its variants, along with those of competitor models, are presented in ARR and ASR metrics. The best result is shown in bold, and the second best is underlined.

Dataset	U.S. stock market		KR stock market		Crypto market		CN stock market		JP stock market		U.K. stock market	
Methodology	ARR(%)↑	ASR↑	ARR(%)↑	ASR↑	ARR(%)↑	ASR↑	ARR(%)↑	ASR↑	ARR(%)↑	ASR↑	ARR(%)↑	ASR↑
CSM [19]	-3.774	-0.238	-6.279	-0.545	-84.36	-3.122	-15.86	-1.358	1.621	0.139	-2.054	-0.181
BLSW [28]	-11.51	-0.706	-16.64	-1.400	-145.0	-4.397	-21.79	-1.722	-9.612	-0.817	3.427	0.187
AlphaStock [43]	9.141	0.495	6.922	0.363	70.56	0.966	11.63	0.486	20.06	1.089	10.43	0.815
DeepTrader [45]	16.80	**0.810**	27.84	0.994	102.6	1.598	29.40	1.154	26.22	1.190	15.48	0.795
MetaTrader [26]	13.69	0.429	5.509	0.368	82.17	1.208	27.31	0.940	15.05	0.526	11.16	0.718
FQ-FRE-CTE	7.213	0.360	10.17	0.427	93.32	1.318	20.25	1.015	6.697	0.399	12.71	0.478
FQ-FRE	12.69	0.434	12.06	0.570	99.28	1.905	25.94	1.219	8.769	0.404	14.62	0.714
FQ-CTE	13.21	0.542	13.42	0.663	116.7	1.834	25.16	1.284	20.29	0.835	18.28	0.688
FQ (proposed)	**36.25**	0.770	**38.57**	**1.008**	**179.4**	**2.370**	**36.45**	**2.109**	**41.93**	**1.263**	**24.80**	**1.413**



(a) U.S. stock market (b) KR stock market (c) Crypto market

(d) CN stock market (e) JP stock market (f) U.K. stock market

Figure 3: The Portfolio Value (PV) across various models in different market datasets during their respective test periods.

Q5 **Case Study in Adaptive Positioning (Section 5.6).** Does FreQuant successfully adapt its weight allocation strategy in response to price fluctuations?

5.1 Experimental Setup

Datasets. We employ six market datasets in our experiments. Table 2 provides an overview of the training and testing periods, as well as the number of assets in each market. Each market dataset encompasses assets that were present throughout both the learning and testing periods. The U.S., KR, and Crypto market datasets have been newly collected from their respective markets. Meanwhile, the CN, JP, and U.K. market datasets, which are public, were utilized in previous studies [34, 49]. To enable a more extensive experiment, we have updated the data periods and assets across all datasets.

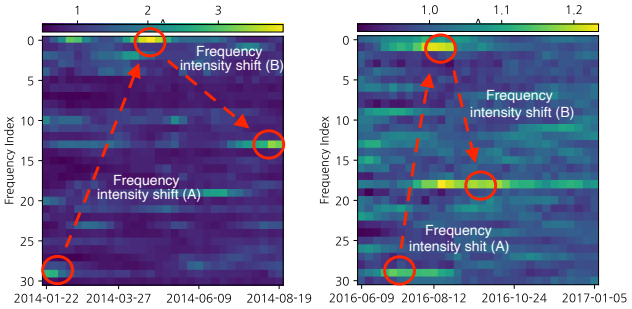
Competitors. We conduct a comparative assessment of FreQuant by evaluating its performance against five established methodologies for portfolio optimization. These methodologies encompass a combination of classical and contemporary techniques. Specifically, our analysis includes two traditional methods, Buying-Loser-Selling-Winner (BLSW) [28] and Cross-Sectional Mean reversion (CSM) [19], and three recently developed portfolio optimization methods, AlphaStock (AS) [43], DeepTrader (DT) [45], and MetaTrader (MT) [26]. All of these approaches are designed for long-short operable scenarios.

Table 2: Time periods and asset counts for different datasets.

Dataset	# Assets	Training Period	Testing Period
U.S. stock market	224	1992.08-2013.08	2013.09-2022.11
KR stock market	528	2001.01-2016.06	2016.07-2022.11
Crypto market	44	2019.01-2021.10	2021.11-2022.11
CN stock market[1]	34	2009.01-2017.09	2017.10-2020.12
JP stock market[2]	118	2013.08-2021.03	2021.04-2023.12
U.K. stock market[2]	21	2011.09-2020.07	2020.08-2023.12

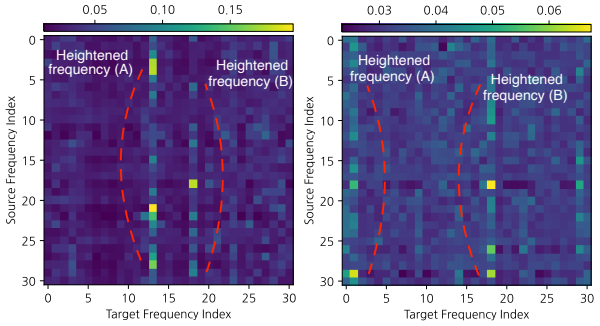
[1] https://github.com/TradeMaster-NTU/TradeMaster
[2] https://datalab.snu.ac.kr/dtml/#datasets

Evaluation Metrics. We use two evaluation metrics to comprehensively evaluate the performance of our approach. First, we examine the Annualized Rate of Return (ARR), providing a direct measure of the method's profitability. Second, we utilize the Annualized Sharpe Ratio (ASR), which offers a balanced perspective by considering both profitability and risk. The higher values for these metrics indicate improved performance.



(a) Sony (SONY) (b) J.P. Morgan (JPM)

Figure 4: The changing frequency intensity ψ within FRE over time for SONY and JPM. The circled regions exhibit dynamic adaptation ability of FRE (see Section 5.3).



(a) SONY on 2014-08-12 (b) JPM on 2016-08-19

Figure 5: Illustrative examples of the heightened frequency features of SONY and JPM on the respective dates using the attention scores S_i from the FRE (details are in Section 5.3).

5.2 Performance Evaluation (Q1)

Table 1 presents the outcomes of the performance evaluation involving FreQuant and competitor models across multiple datasets. In addition to other competitors, we compare FreQuant with its variants. FQ-FRE denotes FreQuant without the FRE module, and FQ-CTE signifies the replacement of the CTE module with two original Transformer encoders [38], separately for the real and imaginary parts of complex features. FQ-FRE-CTE removes the FRE module from FreQuant while incorporating the original Transformer encoder in place of CTE.

FreQuant demonstrates robust performance in the portfolio optimization task, as evident from Table 1. It shows superior performance in terms of ARR, and exhibits strong risk-profit performance by ASR. While FQ-FRE-CTE exhibits performance comparable to previous methods, the persistent performance improvement observed across its variants validates the effectiveness of each module.

Additionally, Figure 3 depicts the dynamic evolution of Portfolio Value (PV) over time, defined as $\prod_{t=2}^{t'}(1 + r_{t-1})$, where t' is the current rebalancing time-step and r_{t-1} is the reward at rebalancing time-step $t-1$, following Eq. (6). While high volatility induced minor fluctuations in earnings, FreQuant effectively identifies and capitalizes pivotal trading assets by incorporating both dominant and abrupt events. This aligns with the results in Table 1 where FreQuant achieves the highest PV across all datasets.

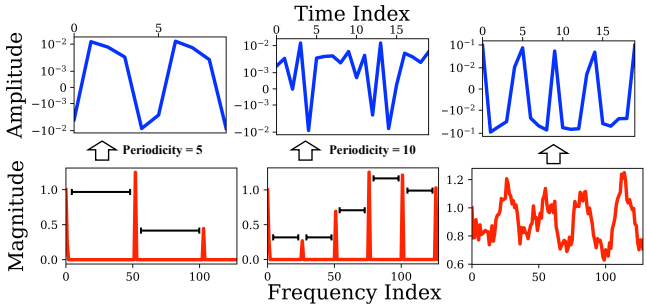


Figure 6: The visualization of event-filters in both the frequency domain (red) and the time domain (blue). The first and second columns correspond to the event-filters with predefined periodicities of 5 and 10, respectively, induced by the guidance in Eq. (17). The third column shows an event-filter trained without any prior, exhibiting the capability of FreQuant to capture and amplify significant multi-periodicity.

5.3 Effects of Frequency-Relation Encoder (Q2)

To demonstrate the dynamic adaptation effect of the Frequency-Relation Encoder (FRE), we investigate the frequency intensity $\psi_i^k = \sum_j S_i^{j,k}$, where $S_i^{j,k}$ denotes the attention score between the j-th and k-th complex features of the i-th asset, as defined in Eq. (10). Figure 4 shows the changing frequency intensities for two assets, Sony (SONY) and J.P. Morgan (JPM), across various time frames within the U.S. market. Notably, the launch of a new game console by Sony in early 2014 prompted a significant event, activating high-frequency domain features in response to rapid price fluctuations. Soon afterward, it shifted its attention towards low-frequency domain features, reflecting settled market conditions. Similarly, the decision by the United Kingdom to exit the European Union (Brexit) in June 2016 led to heightened price volatility, impacting financial institutions like J.P. Morgan. Consequently, both assets exhibited changes in frequency intensities during these periods, effectively capturing evolving market conditions.

Furthermore, Figure 5 provides illustrative examples of heightened frequency features through the FRE. The attention scores depicted exhibit a systematic arrangement, with specific features emphasized by comparatively larger values, highlighting their importance. While importance is generally distributed across both low and high-frequency domain features, there are instances where the most critical component is underlined, especially when distinct patterns emerge and require focused attention.

5.4 Interpretation of Event-filters (Q3)

In Figure 6, we illustrate the appearance of the f-th learned event-filters, $\tilde{\boldsymbol{M}}^{i,f} \in \mathbb{C}^{T_0}$, in each column. These event-filters are designed to amplify significant frequency features derived from Multi-frequency Decomposition, capturing pivotal recurrences of the assets. As a result, the learned event-filters inherently represent the overarching patterns exhibited by the assets. To visualize this in a comprehensible form, we transform the originally existing event-filters in the frequency domain (red) to the time domain (blue) through the Inverse-DFT (IDFT). In the frequency domain, the magnitude of event-filters, $\boldsymbol{\rho}_f = |\tilde{\boldsymbol{M}}^{i,f}|$, readily shows which frequency features are amplified and focused on the most. Subsequently, the

Table 3: Training statistics with and without Predefined Periodicity Guidance (PPG) in Eq. (17). The values indicate mean cumulative reward followed by reward variance during the respective training episodes.

Dataset	U.S. stock market		KR stock market		Crypto market		CN stock market		JP stock market		U.K. stock market	
Episode	FQ	FQ-PPG	FQ	FQ-PPG	FQ	FQ-PPG	FQ	FQ-PPG	FQ	FQ-PPG	FQ	FQ-PPG
1-10	-0.5(1.3)	-1.2(2.7)	0.6(1.5)	-0.1(2.7)	-0.2(1.3)	-0.4(1.6)	-1.1(0.3)	-1.0(0.3)	-0.7(0.2)	-1.2(0.4)	-0.4(1.0)	-0.5(1.2)
11-20	3.1(0.7)	0.1(3.5)	1.9(0.8)	0.3(1.8)	1.1(1.9)	1.0(4.3)	0.4(0.5)	0.5(0.8)	0.6(0.3)	1.6(5.2)	1.2(0.1)	1.7(0.5)
21-30	3.7(0.4)	1.7(4.7)	4.0(0.6)	0.6(1.0)	2.8(0.9)	1.3(2.7)	2.7(0.3)	2.6(0.4)	2.3(2.5)	8.2(0.8)	2.0(0.2)	1.9(0.3)
31-40	5.2(0.6)	3.8(2.9)	6.6(2.0)	3.0(2.1)	3.1(0.5)	1.5(3.1)	4.7(0.3)	3.8(0.3)	9.1(1.0)	8.3(2.1)	2.3(0.1)	2.0(0.7)

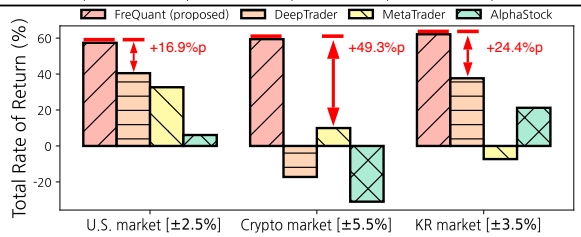


Figure 7: The total rate of returns achieved on days with significant market shifts. The values next to the market type indicate the cut-off points used for identifying market shifts.

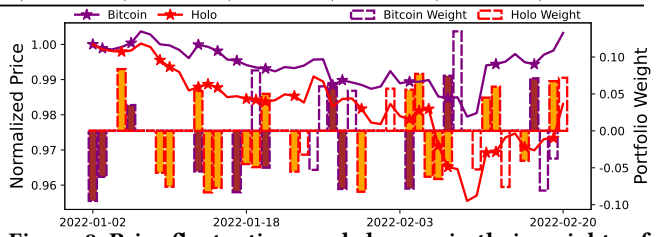


Figure 8: Price fluctuations and changes in their weights of Bitcoin and Holo in the cryptocurrency market dataset. The line chart shows the normalized price, while the bar chart displays portfolio weights. We mark and color instances whenever the sign of the weight matches the price movement on the following day.

partial time domain sequence represented by IDFT($\tilde{\boldsymbol{M}}^{i,f}$) depicts the targeted periodical patterns of price signals that the event-filters aim to amplify. The figure reveals the event-filters' focus on a spectrum of periodic patterns, ranging from fundamental patterns to arbitrarily trained complex patterns shared among the assets. This affirms the ability of FreQuant to incorporate not only fundamental periodic patterns but also multi-periodic patterns.

5.5 Empirical Validation of the Robustness (Q4)

In Figure 7, we validate the robustness of FreQuant using real-market data by identifying periods with significant market shifts via quantile-based outlier detection. Specifically, we formulate a probability density function, denoted as P, which describes the distribution of the rate of change X in the market index, expressed on a percentage scale. The critical cut-off point d is set where $1 - P(-d < X \leq d) = 0.1$, indicating the 10th percentile of market index rate changes. Dates with absolute changes exceeding d were marked as high fluctuation dates, implying sudden market shifts. We compare total rate of returns across different DRL-based models on these dates, with results confirming FreQuant's superior performance and resilience during abrupt market shifts.

In addition, for the regularization term introduced in Eq. (17), we conducted experiments to evaluate the impact of Predefined Periodicity Guidance (PPG) on learning stability by comparing the model's training statistics with and without PPG across several episodes. In Table 3, the values are represented by $\mu(\sigma^2)$; μ indicate the mean cumulative reward, while σ^2 represents its variance. FQ refers to our proposed method FreQuant, incorporating PPG, and FQ-PPG denotes FreQuant without PPG by setting $\lambda = 0$ in Eq. (18). Results indicate that FQ with PPG consistently shows lower variance than FQ-PPG, demonstrating PPG's role in stabilizing learning and confirming its significant contribution.

5.6 Case Study in Adaptive Positioning (Q5)

We conduct case studies on two cryptocurrencies, Bitcoin and Holo, throughout a designated test period to evaluate the adaptive capability of FreQuant in reallocating weights in response to price movements. Figure 8 illustrates the price dynamics of these assets alongside their weight adjustments during this period. Notably, FreQuant demonstrates an ability to make independent investment decisions for each asset, not overly reliant on broader market trends. This indicates that while an influential asset like Bitcoin might determine the overall market trend, individual assets possess distinct behaviors based on their unique circumstances. Consequently, assets do not uniformly follow a single investment direction; instead, they are capable of adopting strategies contrary to prevailing market trends, based on awareness of their distinct conditions. This confirms FreQuant's adaptive positioning in the face of market fluctuations, highlighting its refined approach to portfolio optimization problem in volatile environments.

6 CONCLUSION

In this paper, we introduce FreQuant, a DRL-based method for adaptive portfolio optimization that exploits DFT-derived frequency features to capture both prevalent and abrupt events. FreQuant employs the Frequency State Encoder for asset representation, adaptively highlighting frequency features. The Portfolio Generator generates an optimal portfolio by modeling asset interrelationships. Comparative assessments show the state-of-the-art performance of FreQuant on multiple datasets for profit and risk criteria. Visualization of the Frequency-Relation Encoder's attention map demonstrates dynamic highlighting of key frequency features. These findings underscore the effectiveness of incorporating frequency features in addressing the portfolio optimization problem. Future works include integrating diverse market and asset features.

ACKNOWLEDGMENTS

This work was supported by DeepTrade Technologies Inc. U Kang is the corresponding author.

REFERENCES

[1] Carlo Altavilla, Luca Brugnolini, Refet S Gürkaynak, Roberto Motto, and Giuseppe Ragusa. 2019. Measuring euro area monetary policy. *Journal of Monetary Economics* 108 (2019), 162–179.

[2] Vitor Azevedo and Christopher Hoegner. 2023. Enhancing stock market anomalies with machine learning. *Review of Quantitative Finance and Accounting* 60, 1 (2023), 195–230.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[4] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271* (2018).

[5] L. Blume and D. Easley. 1992. Evolution and market behavior. *Journal of Economic Theory* 58 (1992), 9–40. https://doi.org/10.1016/0022-0531(92)90099-4

[6] James W Cooley and John W Tukey. 1965. An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation* 19, 90 (1965), 297–301.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional Transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[8] Andrea Flori, S. Giansante, C. Girardone, and F. Pammolli. 2019. Banks' business strategies on the edge of distress. *Annals of Operations Research* 299 (2019), 481–530. https://doi.org/10.1007/s10479-019-03383-z

[9] Scott Fujimoto, Herke van Hoof, and David Meger. 2018. Addressing Function Approximation Error in Actor-Critic Methods. In *International Conference on Machine Learning*. https://api.semanticscholar.org/CorpusID:3544558

[10] Siyu Gao, Yunbo Wang, and Xiaokang Yang. 2023. Stockformer: Learning hybrid trading machines with predictive coding. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*. 4766–4774.

[11] Ramazan Gençay, Faruk Selçuk, and Brandon J Whitcher. 2001. *An introduction to wavelets and other filtering methods in finance and economics*. Elsevier.

[12] Tuomas Haarnoja, Aurick Zhou, P. Abbeel, and Sergey Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *ArXiv* abs/1801.01290 (2018). https://api.semanticscholar.org/CorpusID:28202810

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[14] Akira Hirose. 2012. *Complex-valued neural networks*. Vol. 400. Springer Science & Business Media.

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

[16] Thomas Hollis, Antoine Viscardi, and Seung Eun Yi. 2018. A comparison of LSTMs and attention mechanisms for forecasting financial time series. *arXiv preprint arXiv:1812.07699* (2018).

[17] Harrison Hong and Jiang Wang. 2000. Trading and returns under periodic market closures. *The Journal of Finance* 55, 1 (2000), 297–354.

[18] Min Hou, Chang Xu, Yang Liu, Weiqing Liu, Jiang Bian, Le Wu, Zhi Li, Enhong Chen, and Tie-Yan Liu. 2021. Stock trend prediction with multi-granularity data: A contrastive learning approach with adaptive fusion. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 700–709.

[19] Narasimhan Jegadeesh and Sheridan Titman. 1993. Returns to buying winners and selling losers: Implications for stock market efficiency. *The Journal of finance* 48, 1 (1993), 65–91.

[20] Zhengyao Jiang, Dixing Xu, and Jinjun Liang. 2017. A deep reinforcement learning framework for the financial portfolio management problem. *arXiv preprint arXiv:1706.10059* (2017).

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012).

[22] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.

[23] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[24] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.

[25] John Moody and Matthew Saffell. 1998. Reinforcement learning for trading. *Advances in Neural Information Processing Systems* 11 (1998).

[26] Hui Niu, Siyuan Li, and Jian Li. 2022. MetaTrader: An Reinforcement Learning Approach Integrating Diverse Policies for Portfolio Optimization. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 1573–1583.

[27] Yong-chan Park, Jun-Gi Jang, and U Kang. 2021. Fast and accurate partial fourier transform for time series data. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1309–1318.

[28] James M Poterba and Lawrence H Summers. 1988. Mean reversion in stock prices: Evidence and implications. *Journal of financial economics* 22, 1 (1988), 27–59.

[29] Juan Carlos Rodriguez. 2007. Measuring financial contagion: A copula approach. *Journal of empirical finance* 14, 3 (2007), 401–423.

[30] Simone Scardapane, Steven Van Vaerenbergh, Amir Hussain, and Aurelio Uncini. 2018. Complex-valued neural networks with nonparametric activation functions. *IEEE Transactions on Emerging Topics in Computational Intelligence* 4, 2 (2018), 140–150.

[31] Si Shi, Jianjun Li, Guohui Li, Peng Pan, and Ke Liu. 2021. Xpm: An explainable deep reinforcement learning framework for portfolio management. In *Proceedings of the 30th ACM international conference on information & knowledge management*. 1661–1670.

[32] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. 2014. Deterministic policy gradient algorithms. In *International conference on machine learning*. Pmlr, 387–395.

[33] Yejun Soun, Jaemin Yoo, Minyong Cho, Jihyeong Jeon, and U Kang. 2022. Accurate Stock Movement Prediction with Self-supervised Learning from Sparse Noisy Tweets. In *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 1691–1700.

[34] Shuo Sun, Molei Qin, Haochong Xia, Chuqiao Zong, Jie Ying, Yonggang Xie, Lingxuan Zhao, Xinrun Wang, Bo An, et al. 2023. TradeMaster: A Holistic Quantitative Trading Platform Empowered by Reinforcement Learning. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[35] Shuo Sun, Xinrun Wang, Wanqi Xue, Xiaoxuan Lou, and Bo An. 2023. Mastering Stock Markets with Efficient Mixture of Diversified Trading Experts. In *KDD*. ACM, 2109–2119.

[36] Shuo Sun, Wanqi Xue, Rundong Wang, Xu He, Junlei Zhu, Jian Li, and Bo An. 2022. DeepScalper: A Risk-Aware Reinforcement Learning Framework to Capture Fleeting Intraday Trading Opportunities. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 1858–1867.

[37] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[39] Daixin Wang, Zhiqiang Zhang, Yeyu Zhao, Kai Huang, Yulin Kang, and Jun Zhou. 2023. Financial Default Prediction via Motif-preserving Graph Neural Network with Curriculum Learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2233–2242.

[40] Haozhe Wang, Chao Du, Panyan Fang, Li He, Liang Wang, and Bo Zheng. 2023. Adversarial Constrained Bidding via Minimax Regret Optimization with Causality-Aware Reinforcement Learning. In *KDD*. ACM, 2314–2325.

[41] Heyuan Wang, Tengjiao Wang, Shun Li, Jiayi Zheng, Shijie Guan, and Wei Chen. 2022. Adaptive Long-Short Pattern Transformer for Stock Investment Selection. In *Proc. 31st Int. Joint Conf. Artif. Intell.* 3970–3977.

[42] Heyuan Wang, Tengjiao Wang, and Yi Li. 2020. Incorporating expert-based investment opinion signals in stock prediction: A deep learning framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 971–978.

[43] Jingyuan Wang, Yang Zhang, Ke Tang, Junjie Wu, and Zhang Xiong. 2019. Alphastock: A buying-winners-and-selling-losers investment strategy using interpretable deep reinforcement attention networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 1900–1908.

[44] Rundong Wang, Hongxin Wei, Bo An, Zhouyan Feng, and Jun Yao. 2021. Commission fee is not enough: A hierarchical reinforced framework for portfolio management. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 626–633.

[45] Zhicheng Wang, Biwei Huang, Shikui Tu, Kun Zhang, and Lei Xu. 2021. DeepTrader: a deep reinforcement learning approach for risk-return balanced portfolio management with market conditions Embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 643–650.

[46] Mengyuan Yang, Xiaolin Zheng, Qianqiao Liang, Bing Han, and Mengying Zhu. 2022. A Smart Trader for Portfolio Management based on Normalizing Flows. IJCAI.

[47] Yunan Ye, Hengzhi Pei, Boxin Wang, Pin-Yu Chen, Yada Zhu, Ju Xiao, and Bo Li. 2020. Reinforcement-learning based portfolio management with augmented asset movement prediction states. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 1112–1119.

[48] Jaemin Yoo and U Kang. 2021. Attention-Based Autoregression for Accurate and Efficient Multivariate Time Series Forecasting. In *Proceedings of the 2021 SIAM International Conference on Data Mining, SDM 2021, Virtual Event, April 29 - May*

1, 2021, Carlotta Demeniconi and Ian Davidson (Eds.). SIAM, 531–539.
[49] Jaemin Yoo, Yejun Soun, Yong-chan Park, and U Kang. 2021. Accurate multivariate stock movement prediction via data-axis transformer with multi-level contexts. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2037–2045.
[50] Liheng Zhang, Charu Aggarwal, and Guo-Jun Qi. 2017. Stock price prediction via discovering multi-frequency trading patterns. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2141–2149.
[51] Zihao Zhang, Stefan Zohren, and Stephen Roberts. 2020. Deep learning for portfolio optimization. *The Journal of Financial Data Science* (2020).
[52] Lifan Zhao, Shuming Kong, and Yanyan Shen. 2023. DoubleAdapt: A Meta-learning Approach to Incremental Learning for Stock Trend Forecasting. In *KDD*. ACM, 3492–3503.

APPENDIX

A EXPERIMENT DETAILS

A.1 Detailed Experimental Setup

The search space and values of the hyperparameters used in FreQuant are summarized in Table 4. Besides the hyperparameters shared across the datasets in Table 4, we set the number of invested assets in the portfolio ($|\mathcal{G}|$) as 40, 80, 16, 14, 25, and 10 for the U.S., KR, Crypto, CN, JP, and U.K. market datasets, respectively. These numbers are determined depending on the number of available assets to invest in the respective markets. The transaction fee rates for closing long (f_l) and short (f_s) positions are set to $0.0033 = 0.33\%$ and $0.004 = 0.4\%$, respectively. Oftentimes, the transaction fee rate for the closing short positions is higher than that for long positions, as the trading market treats the short position as a borrowed asset (before realization), hence the interest rate is applied and induced higher fee rate. While we constrain the sum of the absolute weights to be 1 (e.g., invest only what we have), in some other works, the settings may be extended to borrowing assets for short investments, allowing the sum of the absolute weights to exceed 1.

All experiments were conducted on a workstation equipped with a GTX 1080 Ti.

Table 4: The search space and the values of the hyperparameters used in FreQuant.

Hyperparameter	Search Space	Value
Actor learning rate (α_{θ^μ})	$[5 \times 10^{-5}, 1 \times 10^{-3}]$	1×10^{-4}
Critic learning rate (α_{θ^Q})	$[5 \times 10^{-5}, 1 \times 10^{-3}]$	5×10^{-5}
Regularization coefficient (λ)	$[1 \times 10^{-3}, 1]$	5×10^{-1}
Fee rate for long positions (f_l)	–	0.33%
Fee rate for short positions (f_s)	–	0.4%
Window size (T)	$[64 - 378]$	256
Buffer limit	$[5,000 - 100,000]$	$10,000$
Batch size (B)	$[8 - 32]$	8
Soft-update rate (τ)	$[1 \times 10^{-3}, 1 \times 10^{-1}]$	1×10^{-2}
Discount factor (γ)	$(0, 1]$	0.99
# of episode (E)	$[50, 1000]$	150
# of units in MEFN (I)	$[1 - 4]$	2
# of FSE block (M)	$[1 - 4]$	4
1×1-conv feature maps (F_0)	$[5 - 32]$	10
Conv1d feature maps (F_i)	$[5 - 64]$	16
CTE linear projection dim (d_k)	$[32 - 256]$	64
# of event-filters (u_i)	$[1 - 10]$	5
# of predefined periodicities (η)	$[5 - 10]$	10