STAM-Net: Spectral-Temporal Adaptive Memory Network for Financial Time Series

Kai-Ye Hu[1,*], Wenyun Xiao[1,*], Ziyuan Liu[1], Hongnian Wang[2,†]
[1]College of Information Science and Technology, Jinan University, Guangzhou, China
{hu634565644, xwyxwy, lzy2002}@stu2024.jnu.edu.cn
[2] Key Laboratory of Digital-Intelligent Disease Surveillance and Health Governance,
North Sichuan Medical College, Nanchong, China
hongnianwang@gmail.com

***Abstract*—Stock return prediction from financial time series remains challenging due to the dynamic and non-stationary nature of financial markets. Different market conditions require different temporal horizons and frequency components, yet existing methods typically process temporal and spectral information separately. To address this, we propose STAM-Net, a spectral-temporal adaptive memory network where spectral characteristics guide temporal memory aggregation, and the resulting temporal representations in turn refine frequency filtering. Specifically, global spectral statistics adapt the memory retention of the temporal encoder to balance short-term fluctuations and long-term dependencies, while the learned temporal representations guide spectral gating to suppress noise and preserve informative components. Experiments on the CSI300 and CSI500 benchmarks show that STAM-Net achieves the best overall performance among the compared methods in both accuracy and stability, suggesting the value of adaptive spectral-temporal coupling for modeling non-stationary financial time series.**

***Index Terms*—Stock return prediction, financial time series, spectral-temporal modeling, adaptive memory network, non-stationary sequence modeling**

I. Introduction

Stock return prediction remains a challenging problem in financial time series modeling [1], [2]. Market conditions can rapidly shift between high and low volatility, trending and range-bound patterns, or stable and turbulent periods [3]. These shifts directly affect what information is predictive: during volatile periods, recent patterns may be more informative, while stable markets may benefit from longer historical trends [4]. Similarly, the relevance of different frequency components can vary across market states. This motivates the need for models that can adapt their temporal and spectral processing to current market conditions.

Most existing approaches model temporal and spectral information with limited bidirectional interaction. On the temporal side, Transformer-based models have been proposed for stock return prediction to capture long-term dependencies. Representative works include MASTER [1], which uses alternating intra- and inter-stock attention to model market correlations, graph-based market models such as ALSGCN [5], which exploit long- and short-term relational signals through attention-based graph convolution, and hybrid architectures like BiLSTM-MTRAN-TCN [6] that combine recurrent and attention mechanisms. However, the quadratic complexity of self-attention presents scalability challenges for large-scale markets. As an alternative, RNN-based approaches such as TRNN [7] and linear recurrent models like RWKV-TS [3] offer different trade-offs in terms of computational efficiency and modeling capacity. Despite these advances, temporal models usually operate independently from frequency-domain information, processing temporal patterns without using spectral characteristics that may reveal market regime changes.

On the spectral side, decomposition-based methods using discrete wavelet transform (DWT) [8], complementary ensemble empirical mode decomposition (CEEMD) [9], or variational mode decomposition (VMD) [10] typically apply decomposition as a preprocessing step. More recent work like FusAD [11] introduces adaptive time-frequency fusion with learnable denoising mechanisms. However, these spectral methods process frequency information independently from temporal context, applying consistent filtering criteria across different temporal states, even though the informativeness of frequency components is context-dependent. For instance, high-frequency fluctuations may represent noise during stable periods but important signals during volatile events.

To address this limitation, we propose the Spectral-Temporal Adaptive Memory Network (STAM-Net), in which spectral and temporal processing are mutually conditioned: spectral features inform temporal aggregation, and the resulting temporal representations in turn govern frequency filtering. Unlike prior frequency-aware methods that typically rely on static decomposition or fixed filtering strategies, STAM-Net extracts global spectral statistics from the input and uses them to modulate memory decay in a linear recurrent encoder, enabling adaptive balance between short-term and long-term temporal dependencies. The learned temporal representations then provide instance-specific context for adaptive frequency gating, suppressing noise while preserving informative spectral content. An adaptive fusion module then integrates both representations

[*] Kai-Ye Hu and Wenyun Xiao contributed equally to this work.
[†] Corresponding author: Hongnian Wang.
This work was supported by the Sichuan Science and Technology Program (Grant No. 2026NSFSC1446).

for final prediction.

Our main contributions are as follows:

1) We propose STAM-Net, a spectral-temporal adaptive memory network that uses spectral statistics to guide temporal memory aggregation and uses the learned temporal representations to guide frequency filtering.
2) We design a Spectral-Informed Linear Recurrent (SILR) encoder that adjusts memory retention based on input spectral characteristics, and a Dynamic Spectral Gating Module (DSGM) that uses temporal context for adaptive frequency filtering.
3) Experiments on CSI300 and CSI500 show that STAM-Net achieves the best overall performance among the compared methods in both accuracy and stability.

II. Related Work

A. Deep Learning for Stock Prediction

Early deep learning approaches based on Recurrent Neural Networks (RNNs), such as Long Short-Term Memory (LSTM) networks [12] and Gated Recurrent Units (GRU) [13], improved the modeling of nonlinear dynamics in stock markets. However, recurrent architectures suffer from vanishing gradients and limited parallelization, which reduces their efficiency on long financial sequences [14].

Transformer-based models [15] use self-attention to model long-range dependencies but suffer from quadratic complexity, limiting their scalability to long financial time series. Several efficient architectures, such as Informer [16], LogTrans [17], Reformer [18], and Linformer [19], reduce this complexity through sparsity or low-rank approximations.

Hybrid architectures integrate convolutional structures such as CNNs or TCNs with Transformers to capture multi-scale patterns, modeling both local fluctuations and long-term trends [6], [20]. Graph-based methods such as ALSGCN [5] further highlight the value of modeling inter-stock relations with attention mechanisms, but they mainly emphasize relational structure rather than adaptive interaction between temporal and spectral signals. More recently, RWKV-TS [3] demonstrates efficient modeling of long financial sequences. Despite these advances, temporal models do not explicitly use spectral characteristics to modulate temporal aggregation, limiting their adaptability across different market regimes.

B. Frequency-Domain Methods for Time Series

Frequency-domain analysis has been widely explored to enhance stock return prediction by disentangling complex signals and suppressing noise. Many methods use signal decomposition as a preprocessing step to separate noise from trends. These approaches include wavelet-based decomposition, as applied by Ren et al. [8] to LSTM and Informer models, and mode-based decomposition, as used by Li et al. [9], Zhao et al. [10], and Liu et al. [21] in hybrid architectures.

Beyond preprocessing, recent studies embed spectral analysis directly into model backbones to handle non-stationary dynamics. Fourier-based approaches such as FiLM [22], Non-stationary Transformer [23], and FITS [24] leverage frequency representations to mitigate distribution shifts, while WaveletMixer [25] explicitly models multi-scale patterns through wavelet convolutions. Recent frequency-domain graph models such as CFGAN [26] further extend spectral modeling to inter-series dependencies, and FusAD [11] introduces adaptive thresholding to fuse time-frequency representations. However, existing approaches have limitations. Decomposition-based methods [8]–[10], [21] apply spectral analysis as a static preprocessing step, while methods that embed frequency processing into backbones [11], [22]–[24] typically use fixed filtering strategies. Overall, these approaches still lack an adaptive bidirectional mechanism in which temporal context guides spectral processing while spectral characteristics inform temporal aggregation.

III. Methodology

A. Problem Definition

Consider a universe of N stocks. For stock i, the historical observations form a multivariate time series:

$$\mathbf{X}_i = \{\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,T}\},$$

where T is the lookback window length and $\mathbf{x}_{i,t} \in \mathbb{R}^D$ contains D features (e.g., open, high, low, close, volume, and volume-weighted average price (VWAP)).

The task is to learn a function f that predicts future stock returns from historical data:

$$f : \mathbf{X} \to \mathbf{Y},$$

where $\mathbf{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\} \in \mathbb{R}^{N\times T\times D}$ and $\mathbf{Y} = \{y_1, \ldots, y_N\}$ with y_i denoting the one-step-ahead return of stock i. In the following, we use $\mathbf{X} \in \mathbb{R}^{B\times T\times D}$ for the stock sequences processed together, where B is the number of stocks, and omit explicit indexing along this dimension for simplicity.

B. Overall Architecture

Figure 1 illustrates the end-to-end pipeline of STAM-Net. Given $\mathbf{X}$, linear projection and positional encoding produce $\mathbf{H}_0$. SILR uses spectral statistics from $\mathbf{H}_0$ to adapt memory decay and outputs $\mathbf{H}_{\text{time}}$. DSGM then uses $\mathbf{H}_0$ together with temporal context $\mathbf{H}_{\text{time}}$ to generate $\mathbf{H}_{\text{spec}}$ through context-aware frequency gating. ATS-Gate fuses the two representations, and a lightweight GRU decoder predicts the next-period return.

The framework contains three main components:

- **Temporal Path:** The SILR encoder takes the input embedding $\mathbf{H}_0$ and captures long-range sequential dependencies. SIDA first extracts global spectral statistics from $\mathbf{H}_0$ to adapt memory decay, after which the LRM block processes the sequence with the adjusted decay rate.
- **Spectral Path:** The DSGM takes $\mathbf{H}_0$ as input and processes it through two parallel spectral branches: a global Fast Fourier Transform (FFT)-based branch that uses temporal context from SILR for adaptive frequency gating, and a local wavelet-based branch for multi-scale feature extraction.
- **Adaptive Fusion:** ATS-Gate combines the temporal and spectral representations through a learnable gating mechanism.

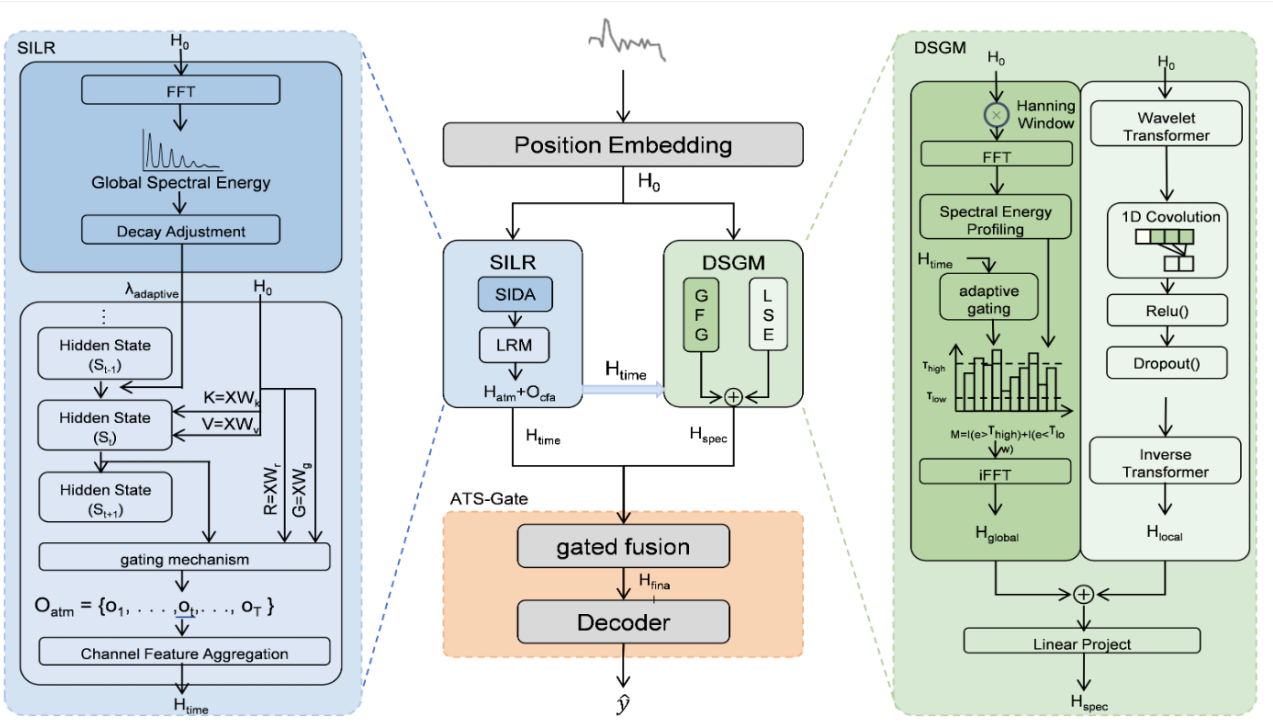

Fig. 1. Overview of the STAM-Net framework. SILR adapts temporal memory using spectral statistics, DSGM performs context-aware spectral gating, and ATS-Gate fuses both branches for prediction.

C. Input Embedding

The raw input $\mathbf{X}$ is projected into a latent space of dimension d via a linear layer, followed by learnable positional encodings $\mathbf{P}$ to preserve temporal order:

$$\mathbf{H}_0 = \text{Linear}(\mathbf{X}) + \mathbf{P} \in \mathbb{R}^{B\times T\times d},$$

where d is the hidden dimension.

D. Spectral-Informed Linear Recurrent Encoder (SILR)

The temporal path consists of the Spectral-Informed Decay Adapter (SIDA) and the Linear Recurrent Memory (LRM) block. The motivation is that a fixed decay rate is unlikely to be optimal across different market regimes: volatile periods often require shorter effective memory, whereas stable periods may benefit from slower decay. SIDA therefore extracts global spectral statistics from the input and uses them to modulate the memory retention of the recurrent encoder.

1) Spectral-Informed Decay Adapter (SIDA): Traditional recurrent models often use fixed or input-agnostic decay rates, limiting their flexibility in handling diverse market regimes. To address this issue, SIDA summarizes the global spectral energy of $\mathbf{H}_0$ and converts it into an instance-specific decay adjustment:

$$\mathbf{E}_{\text{freq}} = \text{Mean}\big(|\mathcal{F}(\mathbf{H}_0)|^2, \text{dim} = \text{freq}\big) \in \mathbb{R}^{B\times d},$$
$$\Delta\boldsymbol{\lambda} = \text{MLP}(\mathbf{E}_{\text{freq}}) \in \mathbb{R}^{B\times d},$$

where $\mathcal{F}(\cdot)$ denotes the FFT. This design lets the model increase or decrease memory retention according to the dominant spectral characteristics of each sample, rather than relying on a single global decay pattern.

2) Linear Recurrent Memory (LRM) Block: The LRM block extracts temporal features through recurrent processing. Given the normalized input $\mathbf{H}_{\text{in}} = \text{LayerNorm}(\mathbf{H}_0)$, the block contains two sub-layers:

a) Adaptive Temporal Mixing (ATM): The ATM layer implements temporal mixing with adaptive memory decay in a multi-head manner. We first project $\mathbf{H}_{\text{in}}$ into four vectors: Receptance ($\mathbf{R}$), Key ($\mathbf{K}$), Value ($\mathbf{V}$), and Gate ($\mathbf{G}$):

$$\mathbf{R} = \mathbf{H}_{\text{in}}\mathbf{W}_r,\ \mathbf{K} = \mathbf{H}_{\text{in}}\mathbf{W}_k,\ \mathbf{V} = \mathbf{H}_{\text{in}}\mathbf{W}_v,\ \mathbf{G} = \mathbf{H}_{\text{in}}\mathbf{W}_g,$$

where $\mathbf{W}_{r,k,v,g} \in \mathbb{R}^{d\times d}$ are weight matrices. To improve representation diversity, ATM evenly partitions the projected features into H heads, each of dimension $d_h = d/H$, and applies the same recurrent temporal mixing within each head in parallel. The resulting head-wise outputs are then concatenated and projected back to the model dimension through $\mathbf{W}_o$.

At each time step t, let $\mathbf{r}_t, \mathbf{k}_t, \mathbf{v}_t, \mathbf{g}_t$ denote the corresponding vectors. The recurrence is governed by the adaptive decay parameter derived from SIDA:

$$\boldsymbol{\lambda}_{\text{adaptive}} = \sigma(\boldsymbol{\lambda}_{\text{base}} + \Delta\boldsymbol{\lambda}).$$

Here, the sample-wise decay adjustment is broadcast along the temporal dimension. For readability, we write the recurrent update in element-wise form:

$$\mathbf{S}_t = \boldsymbol{\lambda}_{\text{adaptive}} \odot \mathbf{S}_{t-1} + \mathbf{k}_t \odot \mathbf{v}_t.$$

The intermediate output is then retrieved from the recurrent state and regulated by an output gate:

$$\mathrm{wkv}_t = \mathrm{GroupNorm}(\mathbf{r}_t \odot \mathbf{S}_t),$$
$$\mathbf{o}_t = \big(\mathrm{wkv}_t \odot \mathrm{SiLU}(\mathbf{g}_t)\big)\mathbf{W}_o,$$

where $\mathbf{S}_t$ acts as the recurrent memory and $\mathrm{SiLU}(\mathbf{g}_t)$ controls how much retrieved content is exposed at time step t. $\mathbf{W}_o$ is the output projection matrix, and the final sequence is $\mathbf{O}_{\mathrm{atm}} = \{\mathbf{o}_1, \ldots, \mathbf{o}_T\}$.

b) Channel Feature Aggregation (CFA): After temporal mixing, a lightweight channel refinement step emphasizes informative dimensions and suppresses noisy ones. The CFA layer operates on $\mathbf{H}_{\mathrm{atm}} = \mathbf{H}_0 + \mathbf{O}_{\mathrm{atm}}$. Let $\mathbf{Z} = \mathrm{LayerNorm}(\mathbf{H}_{\mathrm{atm}})$ and compute

$$\mathbf{R}' = \mathbf{Z}\mathbf{W}'_r, \quad \mathbf{K}' = \mathbf{Z}\mathbf{W}'_k.$$

The channel-refined output is

$$\mathbf{O}_{\mathrm{cfa}} = \sigma(\mathbf{R}') \odot \big(\mathrm{ReLU}(\mathbf{K}')^2\mathbf{W}'_v\big),$$

where $\mathbf{W}'_r, \mathbf{W}'_k \in \mathbb{R}^{d\times d}$ are learnable weight matrices for the receptance and key projections, and $\mathbf{W}'_v \in \mathbb{R}^{d\times d}$ projects the activated features back to the model dimension.

Finally, the encoder output is obtained via

$$\mathbf{H}_{\mathrm{time}} = \mathbf{H}_{\mathrm{atm}} + \mathbf{O}_{\mathrm{cfa}}.$$

$\mathbf{H}_{\mathrm{time}}$ contains the temporal features and provides context for the spectral gating module.

E. Dynamic Spectral Gating Module (DSGM)

While SILR effectively models sequential dependencies, distinguishing structural trends from broadband noise remains difficult in the time domain alone. DSGM addresses this issue by processing $\mathbf{H}_0$ in the frequency domain, while using temporal context from SILR to decide which spectral components are likely to be informative under the current market state. It contains a Global Frequency Gating branch (GFG) and a Local Scale Extractor (LSE).

1) Global Frequency Gating (GFG): To capture dominant periodic patterns without relying on a fixed filter, GFG transforms the input into the frequency domain and applies context-aware gating. We first apply a Hann window to mitigate spectral leakage:

$$\tilde{\mathbf{H}} = \mathbf{H}_0 \odot \mathbf{w}_{\mathrm{hann}}, \quad \mathbf{Z} = \mathcal{F}(\tilde{\mathbf{H}}) \in \mathbb{C}^{B\times F\times d},$$

where F is the number of frequency bins.

We compute the Spectral Energy Profile $\mathbf{e} \in \mathbb{R}^{B\times F}$ by aggregating squared magnitudes across the feature dimension:

$$E_{\mathrm{raw}} = \sum_{i=1}^{d} |\mathbf{Z}_{:,:,i}|^2, \quad \mathbf{e} = \frac{E_{\mathrm{raw}}}{\mathrm{Median}(E_{\mathrm{raw}}) + \epsilon}.$$

To enable adaptive spectral filtering, we use temporal context from SILR to compute instance-specific thresholds. We aggregate the temporal features $\mathbf{H}_{\mathrm{time}} \in \mathbb{R}^{B\times T\times d}$ via global average pooling and use an MLP to adjust the filtering thresholds:

$$\mathbf{c}_{\mathrm{ctx}} = \mathrm{GAP}(\mathbf{H}_{\mathrm{time}}) \in \mathbb{R}^{B\times d},$$
$$[\tau_{\mathrm{high}}, \tau_{\mathrm{low}}] = \mathrm{MLP}(\mathbf{c}_{\mathrm{ctx}}) + [\tau^{\mathrm{hi}}_{\mathrm{base}}, \tau^{\mathrm{lo}}_{\mathrm{base}}],$$

where GAP denotes global average pooling along the temporal dimension, and $\tau^{\mathrm{hi}}_{\mathrm{base}}, \tau^{\mathrm{lo}}_{\mathrm{base}}$ are learnable baseline parameters. The thresholds $\tau_{\mathrm{high}}, \tau_{\mathrm{low}} \in \mathbb{R}^B$ are computed independently for each sample, adapting to the current market state.

We generate a frequency gating mask $\mathbf{M} \in \mathbb{R}^{B\times F}$ using a dual-criteria strategy that preserves high-energy frequencies ($\mathbf{e} > \tau_{\mathrm{high}}$), which capture dominant periodic patterns, and selected low-energy frequencies ($\mathbf{e} < \tau_{\mathrm{low}}$), which may contain weak but informative components that would otherwise be discarded:

$$\mathbf{M} = \mathbb{I}\big((\mathbf{e} > \tau_{\mathrm{high}}) \vee (\mathbf{e} < \tau_{\mathrm{low}})\big).$$

The resulting binary mask is then applied to the spectrum and transformed back to the time domain:

$$\mathbf{H}_{\mathrm{global}} = \mathcal{F}^{-1}(\mathbf{Z} \odot \mathbf{M}).$$

In implementation, the gating is applied in a way that preserves the Hermitian symmetry of the spectrum, ensuring that $\mathbf{H}_{\mathrm{global}}$ remains real-valued after the inverse transform.

This mechanism allows temporal context to determine which frequency bands behave like signal and which behave like noise under the current sequence state.

2) Local Scale Extractor (LSE): Adaptive global gating may still miss short-lived local structures, so we add a local wavelet branch to preserve multi-scale fluctuations. Following the LSE branch in Figure 1, we first refine wavelet coefficients locally and then project them back to the model space:

$$\mathbf{U}_{\mathrm{wav}} = \mathrm{Dropout}(\mathrm{ReLU}(\mathrm{Conv1D}(\mathrm{CWT}(\mathbf{H}_0)))),$$
$$\mathbf{H}_{\mathrm{local}} = \mathrm{Linear}\big(\mathrm{CWT}^{-1}(\mathbf{U}_{\mathrm{wav}})\big),$$

where CWT and CWT^{-1} denote the Continuous Wavelet Transform and its inverse. The 1D convolution captures local patterns in the wavelet domain, while the final linear projection matches the *Linear Project* block in Figure 1. We then combine the adaptive global frequency features and the local wavelet features:

$$\mathbf{H}_{\mathrm{spec}} = \mathbf{H}_{\mathrm{global}} + \mathbf{H}_{\mathrm{local}}.$$

F. Adaptive Temporal-Spectral Gate (ATS-Gate)

The relative usefulness of temporal and spectral cues changes across market conditions, so a fixed fusion ratio is often suboptimal. Given the temporal feature $\mathbf{H}_{\mathrm{time}}$ from SILR and the spectral feature $\mathbf{H}_{\mathrm{spec}}$ from DSGM, ATS-Gate dynamically balances their contributions through a learnable gating mechanism:

$$g = \sigma\big(\mathbf{W}_{\mathrm{gate}}[\mathbf{H}_{\mathrm{time}}; \mathbf{H}_{\mathrm{spec}}]\big),$$
$$\mathbf{H}_{\mathrm{final}} = g \odot \mathbf{H}_{\mathrm{time}} + (1 - g) \odot \mathbf{H}_{\mathrm{spec}},$$

where $g \in \mathbb{R}^{B\times T\times d}$ with values in $(0, 1)$ is computed instance-wise. In this way, the model can place more weight on temporal memory or spectral evidence depending on the current market state.

G. Prediction and Training

Prediction Head. Following the *Decoder* block in Figure 1, we use a lightweight GRU-based decoder followed by a linear projection:

$$\{\mathbf{h}_t\}_{t=1}^{T} = \mathrm{GRU}(\mathbf{H}_{\text{final}}), \qquad \hat{y} = \mathrm{Linear}(\mathbf{h}_T).$$

Training Objective. We optimize the model using Mean Squared Error (MSE):

$$\mathcal{L} = \frac{1}{B}\sum_{i=1}^{B}(\hat{y}_i - y_i)^2.$$

IV. Experiments

A. Experimental Setup

Datasets. We evaluate STAM-Net on two Chinese stock market benchmarks: CSI300 and CSI500. The datasets span from 2013 to 2024 with daily trading records. For each stock, we construct Alpha360 factors based on six basic variables (open, close, high, low, volume, and VWAP). We follow a chronological split to avoid look-ahead bias: training (2013–2021), validation (2022–2023), and testing (2024). The look-back window is set to 60 days, and the prediction target is the one-step-ahead daily return. All input features are constructed from information available up to trading day t, and the corresponding prediction is used for decision making on day $t + 1$.

Baselines. We compare STAM-Net with a diverse set of baselines from five categories. *Traditional recurrent models* include MLP, LSTM [12], GRU [13], and ALSTM [27]. *Attention-based models* include Transformer [15], GAT [28], and MASTER [1]. *Frequency-domain methods* include SFM [29]. *Time-frequency fusion methods* include FusAD [11]. *Efficient linear recurrent models* include RWKV-TS [3].

Evaluation Metrics. We use standard metrics in quantitative finance, including IC, Rank IC, ICIR, and Rank ICIR [30]. IC and Rank IC measure the correlation between predicted and realized returns, while ICIR and Rank ICIR evaluate stability by normalizing with temporal standard deviations. Higher values indicate better performance.

Implementation Details. All models are implemented in PyTorch within the Qlib framework [30]. For STAM-Net, we tune the number of ATM heads $H \in \{2, 4, 8, 16\}$ and set $H = 4$ in the final model to balance expressiveness and efficiency. We set the model dimension to $d = 64$ for CSI300 and reduce it to $d = 32$ for CSI500 to accommodate its higher volatility and noise level. The dropout rate is set to 0.2. STAM-Net is specifically designed for daily-frequency prediction, where inference is performed once per trading day. For baseline models, we tune the number of layers $\{1, 2, 3\}$, hidden dimensions $\{64, 128, 256\}$, and learning rates $\{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$, selecting the best configuration based on validation IC. All methods are trained for at most 150 epochs on an NVIDIA RTX 4090 GPU with early stopping, and each experiment is repeated 10 times.

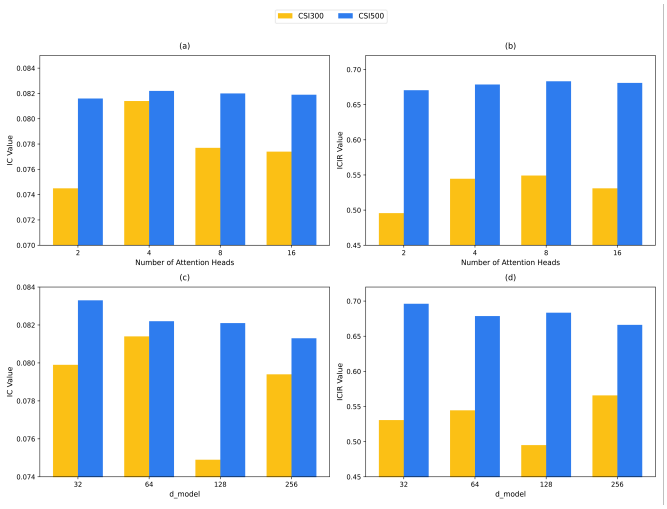

Fig. 2. Hyperparameter sensitivity of STAM-Net to the number of ATM heads and model dimension d.

B. Main Results

Table I reports the performance comparison on CSI300 and CSI500. STAM-Net achieves the best values on all reported metrics among the compared methods.

On CSI500, STAM-Net improves the best baseline IC from 0.0772 to 0.0833 and the best baseline Rank IC from 0.0676 to 0.0720. It also yields higher stability, increasing ICIR from 0.6124 to 0.6962 and Rank ICIR from 0.5079 to 0.5867.

On CSI300, STAM-Net likewise outperforms the best baseline on each of the four metrics, achieving an IC of 0.0814, a Rank IC of 0.0758, an ICIR of 0.5446, and a Rank ICIR of 0.5005. These consistent improvements support the benefit of adapting memory and spectral filtering to changing market conditions. Comparisons with SFM, FusAD, and RWKV-TS further suggest that adaptively coupling temporal and spectral representations is more effective than modeling either view largely in isolation under our evaluation setting.

C. Ablation Study

To validate the contribution of each component, we conduct ablation studies on both datasets, as shown in Table II. Each component contributes to the full model.

Removing DSGM leads to clear degradation on both datasets. On CSI300, ICIR drops from 0.5446 to 0.4780 (-12.2%), while the standard deviation across 10 runs increases from ±0.0185 to ±0.1005. This suggests that without spectral gating the model becomes more sensitive to random initialization and less able to suppress noise. Similarly, removing SILR reduces ICIR to 0.4906 on CSI300 (-9.9%) and 0.6493 on CSI500 (-6.7%), indicating that spectral-informed adaptive memory improves temporal aggregation. Removing ATS-Gate also degrades performance, supporting the use of adaptive fusion rather than simple summation. Overall, the full STAM-Net achieves the best performance with the lowest variance across all metrics in this study.

TABLE I
PERFORMANCE COMPARISON ON CSI300 AND CSI500.

Dataset	Model	IC	Rank IC	ICIR	Rank ICIR
CSI300	MLP	0.0623±0.0033	0.0587±0.0039	0.4172±0.0350	0.3800±0.0360
	LSTM	0.0727±0.0045	0.0685±0.0048	0.4930±0.0362	0.4604±0.0373
	ALSTM	0.0743±0.0066	0.0701±0.0060	0.5012±0.0591	0.4658±0.0557
	GRU	0.0777±0.0021	0.0737±0.0021	0.5137±0.0318	0.4813±0.0278
	RWKV-TS	0.0311±0.0134	0.0271±0.0127	0.1956±0.0789	0.1666±0.0729
	SFM	0.0558±0.0011	0.0490±0.0007	0.3466±0.0113	0.2886±0.0112
	FusAD	0.0323±0.0190	0.0291±0.0171	0.1821±0.1136	0.1607±0.0996
	GAT	0.0711±0.0089	0.0655±0.0102	0.4427±0.0573	0.3957±0.0665
	MASTER	0.0746±0.0047	0.0681±0.0014	0.4326±0.0325	0.4045±0.0328
	Transformer	0.0742±0.0020	0.0714±0.0019	0.5017±0.0130	0.4684±0.0100
	STAM-Net	**0.0814±0.0015**	**0.0758±0.0020**	**0.5446±0.0185**	**0.5005±0.0148**
CSI500	MLP	0.0707±0.0010	0.0616±0.0004	0.5365±0.0075	0.4486±0.0100
	LSTM	0.0702±0.0069	0.0604±0.0061	0.4930±0.0362	0.4504±0.0652
	ALSTM	0.0751±0.0048	0.0650±0.0040	0.5908±0.0634	0.5022±0.0543
	GRU	0.0772±0.0081	0.0668±0.0080	0.5979±0.0838	0.5079±0.0863
	RWKV-TS	0.0500±0.0090	0.0416±0.0084	0.3855±0.0692	0.3091±0.0649
	SFM	0.0755±0.0030	0.0634±0.0027	0.6124±0.0313	0.4974±0.0268
	FusAD	0.0513±0.0152	0.0445±0.0134	0.3735±0.1267	0.3141±0.1034
	GAT	0.0770±0.0036	0.0676±0.0030	0.5764±0.0438	0.4899±0.0403
	MASTER	0.0729±0.0045	0.0639±0.0048	0.4900±0.0426	0.4195±0.0385
	Transformer	0.0752±0.0013	0.0654±0.0018	0.5843±0.0128	0.4884±0.0115
	STAM-Net	**0.0833±0.0027**	**0.0720±0.0025**	**0.6962±0.0173**	**0.5867±0.0194**

TABLE II
ABLATION STUDY ON CSI300 AND CSI500.

Dataset	Variant	IC	Rank IC	ICIR	Rank ICIR
CSI300	w/o SILR	0.0740±0.0080	0.0680±0.0084	0.4906±0.0871	0.4445±0.0874
	w/o DSGM	0.0722±0.0110	0.0655±0.0117	0.4780±0.1005	0.4296±0.1081
	w/o ATS-Gate	0.0725±0.0097	0.0675±0.0097	0.4816±0.0918	0.4411±0.0865
	Full Model	**0.0814±0.0015**	**0.0758±0.0020**	**0.5446±0.0185**	**0.5005±0.0148**
CSI500	w/o SILR	0.0792±0.0016	0.0680±0.0020	0.6493±0.0212	0.5483±0.0163
	w/o DSGM	0.0802±0.0029	0.0700±0.0024	0.6225±0.0471	0.5312±0.0378
	w/o ATS-Gate	0.0820±0.0027	0.0705±0.0031	0.6791±0.0288	0.5803±0.0321
	Full Model	**0.0833±0.0027**	**0.0720±0.0025**	**0.6962±0.0173**	**0.5867±0.0194**

D. Hyperparameter Analysis

We further analyze the sensitivity of STAM-Net to two key hyperparameters: the number of ATM heads and the model dimension, as shown in Figure 2.

Number of ATM Heads. Figure 2(a) and (b) show that increasing the number of ATM heads from 2 to 4 improves both IC and ICIR on CSI300 and CSI500, indicating that multi-head temporal mixing captures diverse temporal patterns across different representation subspaces. However, further increasing the number of heads to 8 or 16 yields diminishing returns, with performance even dropping in some cases (e.g., IC on CSI300), likely due to overfitting and noise introduced by parameter redundancy. Accordingly, we set the number of heads to 4 to balance expressiveness and efficiency.

Model Dimension. Figure 2(c) and (d) illustrate the impact of dimension d. For CSI300 (dominated by large-cap stocks), $d = 64$ achieves the best performance, indicating sufficient capacity to capture complex market patterns. In contrast, for the noisier and more volatile CSI500, a smaller dimension $d = 32$ yields better results (especially in ICIR). Larger dimensions ($d = 128$ or 256) cause noticeable performance degradation on CSI500, suggesting that excessive complexity tends to fit noise rather than genuine signals. Thus, we adopt $d = 64$ for CSI300 and $d = 32$ for CSI500 to adapt to different market characteristics and ensure generalization and robustness.

E. Investment Simulation

We conduct backtesting experiments to evaluate the ranking performance of the method under a standard frictionless setting.

Setup. A daily rebalancing strategy selects the top-K stocks by predicted scores to form equally weighted long-only portfolios ($K = 10$ for CSI300 and $K = 15$ for CSI500). Scores are generated from information available up to day t, and trades are executed at the next-day opening prices to avoid look-ahead bias. Transaction costs and slippage are not included in this backtest.

Results. As shown in Figure 3, STAM-Net achieves the highest cumulative return in 2024 among the compared methods, especially on CSI500. It also shows smaller drawdowns during early-2024 market declines and maintains performance during the later rebound period, suggesting that the ranking gains translate effectively under this standardized evaluation setting.

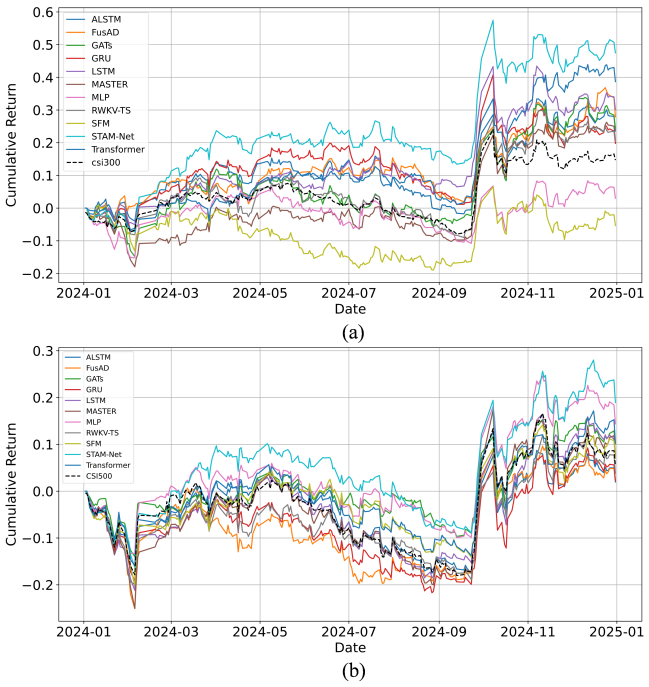

Fig. 3. Cumulative returns of STAM-Net and baselines in 2024 under a frictionless setting on (a) CSI300 and (b) CSI500.

V. Conclusion

We propose STAM-Net, a spectral-temporal adaptive memory network that couples temporal modeling with frequency-domain information. Experiments on CSI300 and CSI500 show that it achieves strong accuracy and stability, outperforming the compared methods under our evaluation setting. These results suggest that adaptive spectral-temporal coupling is a useful design for non-stationary financial time series.

References

[1] T. Li, Z. Liu, Y. Shen, X. Wang, H. Chen, and S. Huang, "MASTER: Market-guided stock transformer for stock price forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 1, 2024, pp. 162–170.

[2] L. Gao, Z. Li, J. Yu, J. Zhang, W. Yao, and H. Wang, "CFC-GMixer: Closed-form continuous-time graph networks with multi-scale feature mixer for stock predictions," in *International Joint Conference on Neural Networks*, 2025.

[3] H. Hou and F. R. Yu, "RWKV-TS: Beyond traditional recurrent neural network for time series tasks," *arXiv preprint arXiv:2401.09093*, 2024.

[4] Z. Wei, A. Rao, B. Dai, and D. Lin, "HireVAE: An online and adaptive factor model based on hierarchical and regime-switch VAE," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 2023, pp. 4903–4911.

[5] J. Yu, W. Yao, Z. Li, L. Gao, W. Xiao, and H. Wang, "ALSGCN: An attention-based long- and short-term graph convolutional network for stock recommendation," in *IEEE International Conference on Systems, Man, and Cybernetics*, Vienna, Austria, 2025, pp. 3186–3191.

[6] S. Wang, "A stock price prediction method based on BiLSTM and improved transformer," *IEEE Access*, vol. 11, pp. 104 211–104 223, 2023.

[7] M. Lu and X. Xu, "TRNN: An efficient time-series recurrent neural network for stock price prediction," *Information Sciences*, vol. 657, p. 119951, 2024.

[8] S. Ren, X. Wang, X. Zhou, and Y. Zhou, "A novel hybrid model for stock price forecasting integrating encoder forest and informer," *Expert Systems with Applications*, vol. 234, p. 121080, 2023.

[9] C. Li and G. Qian, "Stock price prediction using a frequency decomposition based gru transformer neural network," *Applied Sciences*, vol. 13, no. 1, p. 222, 2022.

[10] L.-T. Zhao, Y. Li, X.-H. Chen, L.-Y. Sun, and Z.-Y. Xue, "Mftm-informer: A multi-step prediction model based on multivariate fuzzy trend matching and informer," *Information Sciences*, vol. 681, p. 121268, 2024.

[11] D. Zhang, B. Li, Z. Zhao, F. Nie, J. Gao, and X. Li, "FusAD: Time-frequency fusion with adaptive denoising for general time series analysis," *arXiv preprint arXiv:2512.14078*, 2025.

[12] A. Graves, "Long short-term memory," *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.

[13] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1724–1734.

[14] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*. Pmlr, 2013, pp. 1310–1318.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[16] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[17] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[18] N. Kitaev, L. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," in *International Conference on Learning Representations*, 2020.

[19] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-attention with linear complexity," *arXiv preprint arXiv:2006.04768*, 2020.

[20] Z. Zeng, R. Kaur, S. Siddagangappa, S. Rahimi, T. Balch, and M. Veloso, "Financial time series forecasting using CNN and transformer," *arXiv preprint arXiv:2304.04912*, 2023.

[21] Y. Liu, S. Huang, X. Tian, F. Zhang, F. Zhao, and C. Zhang, "A stock series prediction model based on variational mode decomposition and dual-channel attention network," *Expert Systems with Applications*, vol. 238, p. 121708, 2024.

[22] T. Zhou, Z. Ma, Q. Wen, L. Sun, T. Yao, W. Yin, R. Jin *et al.*, "FiLM: Frequency improved legendre memory model for long-term time series forecasting," *Advances in Neural Information Processing Systems*, vol. 35, pp. 12 677–12 690, 2022.

[23] Y. Liu, H. Wu, J. Wang, and M. Long, "Non-stationary transformers: Exploring the stationarity in time series forecasting," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9881–9893, 2022.

[24] Z. Xu, A. Zeng, and Q. Xu, "FITS: Modeling time series with $10k$ parameters," in *International Conference on Learning Representations*, 2024.

[25] Z. Zhang, T. D. Pham, Y. An, N. P. Doan, M. Alsharari, V.-H. Tran, A.-T. Hoang, H. Vandierendonck, and S. T. Mai, "WaveletMixer: a multi-resolution wavelets based MLP-mixer for multivariate long-term time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 21, 2025, pp. 22 741–22 749.

[26] Z. Li, L. Gao, W. Qiu, W. Xiao, and H. Wang, "CFGAN: Complex frequency-domain graph attention network for time series forecasting," in *2026 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2026, pp. 1231–1235.

[27] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017, pp. 2627–2633.

[28] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, Feb. 2018.

[29] L. Zhang, C. Aggarwal, and G.-J. Qi, "Stock price prediction via discovering multi-frequency trading patterns," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 2141–2149.

[30] X. Yang, W. Liu, D. Zhou, J. Bian, and T.-Y. Liu, "Qlib: An AI-oriented quantitative investment platform," *arXiv preprint arXiv:2009.11189*, 2020.