DHMoE: Diffusion Generated Hierarchical Multi-Granular Expertise for Stock Prediction

Weijun Chen[1], **Yanze Wang**[2*]

[1]School of Computer Science, Peking University, Beijing, China
[2]Wangxuan Institute of Computer Technology, Peking University, Beijing, China
oncecwj@stu.pku.edu.cn; wangyanze@stu.pku.edu.cn

Abstract

Stock prediction stands as a pivotal research objective within the *Fintech*. Existing deep learning research revolves around the development and scaling of one individual neural network predictor. However, in the dynamic and noisy landscape of the stock market, reliance solely on a single predictor poses risks of limited adaptability to diverse market conditions and challenges in effectively integrating multi-source information. Besides, top-down teaching and bottom-up hierarchical decision-making paradigms are critical for robust and accurate stock prediction within successful quantitative firms. Nonetheless, there is scarcely any research that integrates this workflow into stock prediction. To this end, we propose Diffusion Generated Hierarchical Mixture-of-Experts (DHMoE) to emulate such workflow in stock prediction. Specifically, DHMoE is crafted as a three-layer tree structure, where each expert functions as a node within the tree and their parameters are generated in a top-down, recursive manner. Recognizing the leading role of the top-level root expert, we harness the robust capabilities of diffusion models for generating and introduce the Diffusion Inverted Transformer (DIT) as the root expert. The DIT is tailored to receive information from various modalities as conditional inputs and allocate parameters to bottom-level experts. These bottom-level experts are responsible for performing predictions specific to their respective input modalities. The prediction results are then synthesized in a bottom-up manner, culminating in the final prediction outcomes. Experiments on three stock trading datasets reveal that DHMoE outperforms state-of-the-art methods in terms of both cumulative and risk-adjusted returns.

Introduction

The global stock market, with over $90 trillion in market capitalization in 2020[1], stands as a key financial ecosystem where investors actively seek financial assets to meet their investment objectives. Accurate stock prediction is fundamental in realizing trading returns for both individual investors and financial institutions. Nonetheless, the diverse market contexts and heterogeneous data sources present significant challenges for investors in making accurate decisions. The swift advancement of deep learning leads to the proposal of various deep models for stock prediction and quantitative investment. Many models (Wang et al. 2022c; Feng et al. 2022, 2019a; Wang et al. 2022a; Zhang et al. 2022; Wang et al. 2021), evolving from initial frameworks like GRU (Chung et al. 2014) and Transformer (Vaswani et al. 2017), are enhanced specifically for the stock market to yield stronger predictive capabilities. While these advanced models show improved predictive performance compared to traditional machine learning methods (Wang and Leu 1996; Zivot and Wang 2006), they overlook diverse data sources, such as social media information, which hinders their effectiveness in stock prediction and investment profitability.

To address the issue of information loss from relying on a single source in the diverse stock market, a series of multimodal models (Ang and Lim 2022; Sawhney et al. 2021c; Wang et al. 2022b; Yang et al. 2023b; Zhao et al. 2022) are proposed. Despite their meticulous design and commendable performance, they function as individual experts. The prediction results from individual predictors are typically marked by high uncertainty, which contrasts sharply with real-world investment decisions (Sun et al. 2023). Besides, a single neural network predictor often falls short in diversity and flexibility to process varied information sources within the stock market. Furthermore, in quantitative trading firms, it's common to find multiple teams of trading experts, each endowed with extensive knowledge in areas such as

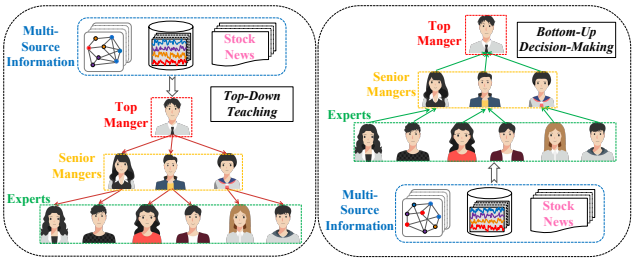

Figure 1: The overall pipeline of our proposed method. The left figure illustrates the top-down teaching paradigm, which helps enhance the collective professional expertise of a team. Conversely, the right figure depicts bottom-up decision-making, highlighting its effectiveness in facilitating robust and precise predictions.

*Corresponding Author.

[1]https://data.worldbank.org/indicator/CM.MKT.LCAP.CD/

finance and psychology. Members within these groups collaborate closely, analyzing the market through their specific lens and making convergent decisions. Following this, a senior manager collates their analyses to render the investment verdict. When making substantial investment decisions, the top decision-maker frequently consults the insights of senior managers to ensure more dependable and informed choices. This bottom-up, hierarchical process is instrumental in crafting trading strategies that are both resilient and profitable. Additionally, teaching within the firm, typically conducted by higher-level and more experienced experts to unseasoned lower-level experts, plays a pivotal role. This educational paradigm promotes the swift transfer of expertise, boosts overall capabilities, and contributes to more robust and precise investment decisions. Therefore, using a single neural network predictor for stock prediction is often disconnected from real trading patterns, which can lead to potential prediction errors and further economic losses.

While the above workflow is effective and well-suited to the task of making more accurate and robust investment decisions in the complex and diverse stock market, integrating these paradigms into the design of model architecture presents two significant challenges, i.e., **(i)** How to design a hierarchical teaching and decision-making process that ensures the effectiveness and efficiency of both the model architecture optimization and inference. **(ii)** How to implement a method that enables the top-level expert to effectively explore and utilize the patterns of heterogeneous data for lower-level teaching or parameter generation purposes.

To handle the challenges mentioned above, we propose the DHMoE, an extension of the Mixture-of-Experts (MoE) architecture designed to facilitate hierarchical teaching and decision-making for multimodal stock prediction. DHMoE employs a three-layer tree structure. Leveraging the efficiency of the tree structure, teaching is conducted through a top-down parameter distribution process, while a sparse synthesized attention mechanism (SSAM) is implemented for bottom-up decision-making. Moreover, drawing inspiration from the powerful representation learning and generative capabilities of diffusion models (Yang et al. 2023a; Lin et al. 2023; Peebles and Xie 2023; Huang, Chen, and Qiao 2024), we develop the Diffusion Inverted Transformer (DIT) to serve as the top-level expert. The DIT receives inputs from diverse modalities as conditional factors, subsequently instructing the middle-level experts, and thereafter, the bottom-level experts. Both the second and third layers consist of experts specific to particular modalities, ensuring tailored processing of distinct data types. Moreover, to maintain a manageable computational demand, all experts at the bottom level are three-layer Multi-Layer Perceptrons (MLPs). The main contributions of our work are as follows:

- We introduce DHMoE, an extension of the Mixture-of-Experts (MoE) architecture designed for heterogeneous data. This initiative represents the first endeavor to adapt the MoE architecture for multimodal stock prediction.
- DHMoE incorporates the paradigm of top-down teaching and bottom-up decision-making. In the design of the top-level expert, we implement the DIT to enhance the robustness and efficacy of heterogeneous pattern mining and parameter generation.
- We conduct extensive experiments on three real-world stock datasets, including NASDAQ, NYSE, and Ashare&HK. Experiments show our method notably outperforms a range of advanced stock prediction models.

Related Work

Stock Prediction

Current deep learning methods in stock prediction focus on three key areas: analyzing financial time series, examining stock-related news, and exploring stock relational information. Within financial time series analysis, prevailing models employ established frameworks like GRU, Transformer, and TCN, tailoring optimizations to align with the distinctive characteristics of the financial market (Feng et al. 2019a; Ding et al. 2020; Zhang, Aggarwal, and Qi 2017; Rezaei, Faaljou, and Mansourfar 2021). With the advancement of graph neural networks (GNNs), a series of methods aim to incorporate stock relational information into prediction. These approaches share a common concept: the mutual influence and evolution among stocks (Sawhney et al. 2021a; Hsu, Tsai, and Li 2021; Feng et al. 2019b; Zhao et al. 2022; Wang et al. 2021, 2022c). For example, stocks of listed companies from the same sector tend to exhibit synchronous trends. Furthermore, it's common practice to integrate stock news sentiment information with financial time series for stock prediction (Wang, Wang, and Li 2020; Zhao et al. 2022; Sawhney et al. 2021c; Ang and Lim 2022). These models typically integrate heterogeneous data at the feature level for the ultimate prediction. In contrast, the MoE architecture permits each expert model to concentrate on learning distinct patterns within the heterogeneous data, enhancing the model's capacity to discern complex data relationships.

Mixture-of-Experts

Despite the Mixture-of-Experts architecture being extensively researched for many years (Jacobs et al. 1991; Shazeer et al. 2017; Ma et al. 2018), its refinement in the fields of stock prediction and quantitative investment is still in the preliminary phases. AMMOE (Li and Xu 2023) is a multi-tasking model that combines the multi-gate MoE and attention modules for stock selection. AlphaMix (Sun et al. 2023) is the first MoE framework for quantitative investment that incorporates a three-stage process to emulate the bottom-up expert opinion summarization prevalent in real-world trading firms. While their innovative MoE architectures for stock prediction are pioneering, they concentrate on technical analysis, utilizing solely financial time series. This methodology leads to considerable information loss and does not fully showcase the capability of MoE architectures in managing diverse data patterns. Besides, they cannot be applied for multimodal modeling in trivial modifications.

Diffusion Models

Diffusion models (DMs), a family of generative models, have become increasingly popular in areas of cutting-edge research (Yang et al. 2023a; Lin et al. 2023; Croitoru et al.

2023). DMs have shown remarkable performance in generating data samples that closely match the observed data distribution, thereby establishing themselves as a potent tool for pattern mining. In time-series modeling, CSDI (Tashiro et al. 2021) utilizes score-based diffusion models conditioned on observed data for time-series imputation. DiffSTG (Wen et al. 2023) merges the spatio-temporal learning prowess of spatio-temporal graph neural networks with the uncertainty quantification provided by diffusion models. In stock prediction, D-Va (Koa et al. 2023) augments the target series with noise via a coupled diffusion process and predicts through a diffusion generative process. DiffsFormer (Gao et al. 2024) proposes a diffusion transformer for stock factor augmentation. Despite promising prospects, the integration of diffusion models into stock prediction is still in its infancy.

Preliminaries

In this study, we formulate the task of stock prediction as a *learning-to-rank* challenge, aimed at enhancing the efficacy of stock investment recommendations. Let $\boldsymbol{p}_i^t$ denote the closing price of stock i at time t, with $y_i^t = \frac{\boldsymbol{p}_i^t - \boldsymbol{p}_i^{t-1}}{\boldsymbol{p}_i^{t-1}}$ representing the corresponding 1-day return ratio. In our context, provided with a period of historical stock time series $\boldsymbol{x}^h \in \mathbb{R}^{N \times T \times D}$, the stock relational graphs $\mathbf{A} \in \mathbb{R}^{S \times N \times N}$, and the stock news information $\mathcal{N} = \{\{\mathcal{N}_{i,j}\}_{i \in [1,N]}\}_{j \in [1,T]}$, we aim to devise a function $\mathcal{F}(\cdot)$ that optimally leverages information from $\boldsymbol{x}^h$, $\mathbf{A}$, and $\mathcal{N}$ to generate a ranking $\mathcal{R}^{T+1} = \{r_1^{T+1} \geq r_2^{T+1} \ldots \geq r_N^{T+1}\}$ for all stocks at time T+1, wherein stocks positioned higher are anticipated to yield greater returns. The ideal outcome positions $r_i^{T+1} \geq r_j^{T+1}$ for any $s_i, s_j \in \mathcal{S}$ provided that $y_i^{T+1} \geq y_j^{T+1}$, where N denotes the number of stocks, D signifies the temporal feature dimension (e.g., close price and open price), T is the lookback window length, and $S = 2$ indicates the number of types of stock relations (i.e., industry and wiki stock relational graphs), $\mathcal{N}_{i,j}$ is the news text for stock i on day j.

Methodology

In this section, we describe our framework of DHMoE for stock prediction in detail. First, we will detail the top-down teaching or parameter generation process in DHMoE, followed by the bottom-up decision-making process. An overview of these two paradigms is illustrated in Fig. 1.

Diffusion Inverted Transformer

In DHMoE, the top-level expert is critical in receiving heterogeneous data signals and informing lower-level experts. Beyond the powerful representational capabilities of diffusion models, their mechanisms including adding and removing noise provide unique benefits in the noisy stock market, adeptly mimicking stochastic stock variations to detect potential signals amid complex market movements. Next, we offer a detailed exploration of the design behind the DIT.

Denoising Diffusion Probabilistic Models Denoising Diffusion Probabilistic Models (DDPMs) learn to generate samples from the target distribution through a forward-backward Markov process (Tashiro et al. 2021; Yang et al. 2023a). The forward process $q(\boldsymbol{x}_{1:K})$ gradually corrupts initial data $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)$ with Gaussian noise:

$$q(\boldsymbol{x}_{1:K}) := \prod_{k=1}^{K} q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) q(\boldsymbol{x}_0) \tag{1}$$

until reaching an isotropic Gaussian distribution $q(\boldsymbol{x}_K) \approx \boldsymbol{\mathcal{N}}(\boldsymbol{x}_K; \mathbf{0}, \mathbf{I})$. Uniquely, given a noise schedule $\boldsymbol{\beta} = \{\beta_1, \ldots, \beta_K\}$ and injected noises $\epsilon \sim \boldsymbol{\mathcal{N}}(\mathbf{0}, \mathbf{I})$, the corrupted data at step k is:

$$\boldsymbol{x}_k = \sqrt{\tilde{\alpha}_k}\boldsymbol{x}_0 + \sqrt{1 - \tilde{\alpha}_k}\epsilon, \tag{2}$$

where $\tilde{\alpha}_k = \prod_{i=1}^{k}(1 - \beta_i), \beta_i \in [0, 1]$. Following the forward process, the backward process $p_\theta(\boldsymbol{x}_{0:K})$ aims to denoise from $\boldsymbol{x}_K$ back to $\boldsymbol{x}_0$, effectively recovering $p_\theta(\boldsymbol{x}_0)$ through a parametric backward kernel $p_\theta(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k)$, with θ representing all learnable backward parameters. Throughout the denoising process, diffusion models progressively refine their grasp of data distribution, thereby improving their generative capability. Typically, this backward kernel is optimized with a denoising network ϵ_θ by minimizing a specific noise-estimate loss function:

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{k, \boldsymbol{x}^0, \epsilon}\|\epsilon - \epsilon_\theta(\boldsymbol{x}_k, k)\|_2^2. \tag{3}$$

Notably, after obtaining the estimated noise, the reverse version of Eq. (2) can be applied to derive the denoised result.

Conditional Diffusion The DDPMs are unconditional generative models, which are not aligned with our motivation where the informed parameters are generated conditioned on multi-source heterogeneous information. Thus, for our task, we initially adapt DDPM to enable conditional prediction by implementing modifications to the backward process. In DDPM, the backward process $p_\theta(\boldsymbol{x}_{0:K})$ is utilized to compute the final data distribution $q(\boldsymbol{x}_0)$. To develop a conditional diffusion model suited to our aim, we incorporate

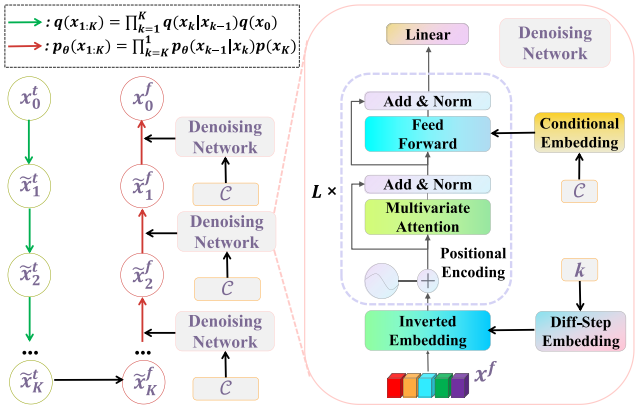

Figure 2: The framework of Diffusion Inverted Transformer (DIT). The DIT acts as the top-level expert, using a conditional diffusion process to generate parameters based on diverse data sources, including stock time series, news, and relational graphs. Through a noise addition and denoising process, it extracts robust features, which are used to generate parameters for lower-level experts.

conditional information $\mathcal{C}$ including historical temporal signals $\boldsymbol{x}^h$ and other embedded heterogeneous information in the backward process. Consequently, the conditioned backward diffusion process can be formulated as:

$$p_\theta(\boldsymbol{x}^f_{0:K}) = p(\boldsymbol{x}^f_K) \prod_{k=K}^{k=1} p_\theta(\boldsymbol{x}^f_{k-1}|\boldsymbol{x}^f_k, \mathcal{C}),$$
$$p_\theta(\boldsymbol{x}^f_{k-1}|\boldsymbol{x}^f_k, \mathcal{C}) = \boldsymbol{\mathcal{N}}(\boldsymbol{x}^f_{k-1}; \mu(\boldsymbol{x}^f_k, k|\mathcal{C}), \sigma_\theta(\boldsymbol{x}^f_k, k|\mathcal{C})\mathbf{I}), \tag{4}$$

where $\boldsymbol{x}^f_0 \in \mathbb{R}^{N\times T\times D}$ denotes the final de-noised signals. Subsequently, let $\boldsymbol{x}^f_0$ serve as an intermediate embedding, which is transformed through a four-layer MLP to generate a set of lower-level parameters $\boldsymbol{G}_i = \mathrm{MLP}_G(\boldsymbol{x}^f_0)$ $(i = 1, ..., a)$ and each $\boldsymbol{G}_i$ encodes distinct conditional information. Additionally, the training objective mentioned in Eq. (3) can be reformulated into a conditional format:

$$\mathcal{L}_{cond} = \mathbb{E}_{k,\boldsymbol{x}^0,\epsilon}\|\epsilon - \epsilon_\theta(\boldsymbol{x}_k, k|\mathcal{C})\|^2_2. \tag{5}$$

Multimodal Embedding In our task, we take three types of modalities into consideration: stock time series, stock news text, and stock relational graphs. Given the timely, granular, and close relevance to stock trading inherent in stock time series, we utilize stock time series as the primary subject for noise addition and removal within the diffusion process. At the same time, we seek to synchronize text information and graph structural information with time series representations via embedding. This approach allows us to incorporate them into the conditional diffusion process. We generate the initial embedding of news texts by averaging token-level outputs from the final layer of BERT (Kenton and Toutanova 2019). Next, we use a one-layer MLP to convert the embedding into specific dimensions that align with the feature dimension of the time series:

$$\mathbf{E}_n = \mathrm{LU}(\mathrm{BERT}(\mathcal{N})W_n + b_n), \tag{6}$$

where W_n, b_n are trainable parameters and LU denotes the LeakyReLU activation function. For embedding stock relational information, we utilize the Simplified Graph Convolution (SGC) (Wu et al. 2019) network to merge graph information with time series information to form the temporal-relational representation, ensuring their alignment:

$$\mathbf{E}_g = \sum_{k=1}^{K}\sum_{i=1}^{S=2}\{(\mathbf{A}_i)^k \boldsymbol{x}^h W_{ik}\}, \tag{7}$$

where W_{ik} are trainable parameters. The conditional information $\mathcal{C} = \{\boldsymbol{x}^h, \mathbf{E}_n, \mathbf{E}_g\}$ includes historical time series and the two types of embedded information.

Inverted Transformer with Conditional Embeddings We depend on timely and granular stock time series and incorporate other alternative data sources as conditional information for noise prediction and further parameter generation, including news text and stock relational graphs. While the Transformer excels in processing heterogeneous data, recent research indicates that vanilla Transformers struggle with temporal data modeling (Zeng et al. 2023). Motivated by the recent advancements in iTransformer's superior temporal modeling capabilities (Liu et al. 2024) and maintaining the core architecture of the Transformer, we enhance the iTransformer better to amalgamate data from diverse modal information for generation purposes. The iTransformer modifies the vanilla input embedding and attention mechanism to focus from the temporal dimension to the variable dimension. Particularly for input $\boldsymbol{x}^f_k \in \mathbb{R}^{N\times T\times D}$, involving multiple stocks, it employs an inverted embedding technique to transform the input into $\boldsymbol{E}_{in}$:

$$\boldsymbol{E}_{in} = (\mathrm{InvEmb}(\boldsymbol{x}^f_k) + \mathrm{StepEmb}(k)) \in \mathbb{R}^{N\times F}, \tag{8}$$

where $\boldsymbol{E}_{in}$ contains embedded tokens, InvEmb and $\mathrm{StepEmb}$ are one-layer MLPs for feature transform. Subsequently, the iTransformer transforms $\boldsymbol{E}_{in}$ into query $\boldsymbol{Q}_h = \boldsymbol{E}_{in}\boldsymbol{W}^Q_h$, key $\boldsymbol{K}_h = \boldsymbol{E}_{in}\boldsymbol{W}^K_h$, and value matrices $\boldsymbol{V}_h = \boldsymbol{E}_{in}\boldsymbol{W}^V_h$ with distinct linear projection parameters $\boldsymbol{W}^Q_h, \boldsymbol{W}^K_h, \boldsymbol{W}^V_h$, where $h = 1, ..., H$ is the head index. Then scaled dot-product attention scores are computed to acquire a weighted sum of the values. The final multivariate attention output is represented by linear transforming the concatenation of all attention heads:

$$\boldsymbol{E}_{att} = \|^H_{h=1}\mathrm{Softmax}(\boldsymbol{Q}_h\boldsymbol{K}^T_h/\sqrt{N})\boldsymbol{V}_h \in \mathbb{R}^{N\times H\cdot F},$$
$$\boldsymbol{E}_m = \mathrm{LayerNorm}((\boldsymbol{E}_{att}\boldsymbol{W}_m + \boldsymbol{b}_m) + \boldsymbol{E}_{in}) \in \mathbb{R}^{N\times F}, \tag{9}$$

where $\boldsymbol{W}_m \in \mathbb{R}^{H\cdot F\times F}, \boldsymbol{b}_m \in \mathbb{R}^F$ are trainable parameters and $\mathrm{LayerNorm}$ is the Layer Normalization (Ba, Kiros, and Hinton 2016). $\boldsymbol{E}_m$ is further processed through a Feed-Forward Network (FFN) and combined with conditional embedding to produce the output of an iTransformer block:

$$\boldsymbol{E}_o = \mathrm{LayerNorm}(\mathrm{FFN}(\boldsymbol{E}_m) + \mathrm{ConEmb}(\mathcal{C}_i)), \tag{10}$$

where ConEmb is one-layer MLP. It's important to note that we utilize different conditional information $\mathcal{C}_i, (i = 1, 2, 3)$, to generate parameters for corresponding middle-level and bottom-level experts, ensuring that the experts are tailored to specific modalities.

Upon gathering the outputs from multiple blocks and concatenating them, passing the combined output through a linear layer yields the final estimated noise for de-noising.

Middle-level Experts

Upon receiving the parameters $\boldsymbol{G}_i, (i = 1, ..., a)$ from the top-level expert DIT, middle-level experts are tasked with integrating their own parameter pool $\boldsymbol{\phi}_i$ to generate diversified experts at the bottom level. $\boldsymbol{\phi}_i = \{\phi_{i,1}, ..., \phi_{i,b}\}$ are bottom-level expert-specific parameters and $\boldsymbol{G}_i$ is the parameter shared among the bottom-level experts, where a is the number of the middle-level experts and b is the number of bottom-level experts for each middle-level expert:

$$\boldsymbol{G}_{i,1:b} = \boldsymbol{G}_i \odot \boldsymbol{\phi}_i, \tag{11}$$

where $\boldsymbol{G}_{i,1:b}$ represents the parameters for the bottom-level experts $1, ..., b$, whose leader (i.e., parent node in the tree structure) is the middle-level expert i.

Bottom-level Experts

Given the potentially large number of low-level experts, along with concerns about computational resource consumption and the challenges of parameter generation, we opt for a lightweight yet effective three-layer FourierMLP (Tancik et al. 2020) as the architecture for bottom-level experts. The ability of FourierMLP to capture high-frequency details makes it preferable to standard MLPs for tasks requiring fine-grained pattern recognition and noise resistance, such as stock prediction. Specifically, the weights and biases for bottom-level experts are derived from $\boldsymbol{G}_{1:b}$. These bottom-level experts are employed to make preliminary decisions.

Bottom-up Decision-Making

Decision by Bottom-level Experts Adhering to the bottom-up decision-making paradigm and reducing the representation learning demands associated with heterogeneous inputs, the bottom-level experts initially make investment decisions on specific modal inputs $\mathbf{X}_{in} \in \{\boldsymbol{x}^h, \mathbf{E}_n, \mathbf{E}_g\}$:

$$\begin{aligned} \mathbf{X}_{\text{fou}} &= \boldsymbol{a}_f[\sin(2\pi\mathbf{X}_{in}\boldsymbol{B}_f)\|\cos(2\pi\mathbf{X}_{in}\boldsymbol{B}_f)], \\ \boldsymbol{H}_{i,k} &= \text{LU}(\text{LU}(\mathbf{X}_{\text{fou}}\boldsymbol{W}_1 + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2), \\ \mathcal{O}_{i,k} &= \text{LU}(\boldsymbol{H}_{i,k}\boldsymbol{W}_3 + \boldsymbol{b}_3), \end{aligned} \tag{12}$$

where $\boldsymbol{a}_f, \boldsymbol{B}_f$ are the trainable parameters of Fourier feature mapping, $\boldsymbol{H}_{i,k} \in \mathbb{R}^{N\times F}$ is the hidden state, $\mathcal{O}_{i,k} \in \mathbb{R}^N$ denotes the ranking result and $\boldsymbol{G}_{i,k} = \{\boldsymbol{W}_{1:3}, \boldsymbol{b}_{1:3}\}$ are the generated parameters. Eq. (12) illustrates the prediction process carried out by the k-th bottom-level expert under the i-th middle-level expert.

Sparse Synthesized Attention Mechanism Upon gathering the prediction results from all bottom-level experts, we employ a sparse synthesized attention mechanism (SSAM) to consolidate the bottom-up investment decisions and arrive at the final prediction outcome. Analogous to the top-down teaching process, bottom-up decision-making unfolds across three layers. Initially, opinions are aggregated from the bottom-level to the middle-level, with insights from bottom-level experts being synthesized into their respective middle-level experts:

$$\begin{aligned} \mathcal{O}_i &= [\mathcal{O}_{i,1}\|\mathcal{O}_{i,2}, ...\|\mathcal{O}_{i,b}] \in \mathbb{R}^{b\times N}, \\ \boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i &= \mathcal{O}_i\boldsymbol{W}_{q,i}, \mathcal{O}_i\boldsymbol{W}_{k,i}, \mathcal{O}_i, \\ \boldsymbol{M}_i &= \text{SA}(\text{RSum}(\text{Softmax}(\boldsymbol{Q}_i\boldsymbol{K}_i^T/\sqrt{N})) + \epsilon)\boldsymbol{V}_i \in \mathbb{R}^N, \end{aligned} \tag{13}$$

where SA denotes the Sparsemax function (Martins and Astudillo 2016) that can guarantee sparse outputs, RSum signifies the operation of summing along the rows, $\boldsymbol{M}_i$ is the prediction result of the i-th middle-level expert, and $\boldsymbol{W}_{q,i}, \boldsymbol{W}_{k,i} \in \mathbb{R}^{N\times F}$ are trainable parameters. Additionally, to avoid the framework's overreliance on specific experts for predictions, we introduce Gaussian noise ϵ. Upon collecting the opinions of all middle-level experts, we similarly aggregate the insights from all middle-level experts to derive the final top-level ranking result $\mathcal{R}$:

$$\begin{aligned} \boldsymbol{M} &= [\boldsymbol{M}_1\|\boldsymbol{M}_2, ...\|\boldsymbol{M}_a] \in \mathbb{R}^{N\times a}, \\ \boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} &= \boldsymbol{M}\boldsymbol{W}_q, \boldsymbol{M}\boldsymbol{W}_k, \boldsymbol{M}, \\ \mathcal{R} &= \text{SA}(\text{RSum}(\text{Softmax}(\boldsymbol{Q}\boldsymbol{K}^T/\sqrt{N})) + \epsilon)\boldsymbol{V} \in \mathbb{R}^N, \end{aligned} \tag{14}$$

where $\boldsymbol{W}_q, \boldsymbol{W}_k \in \mathbb{R}^{a\times F}$ are trainable parameters.

Framework Optimization

Rank Loss Following the previous work on stock ranking (Feng et al. 2019b; Zheng et al. 2023), we apply rank loss as the optimization function for investment recommendation. This loss function is designed to guide the model in ranking stocks based on future returns. After acquiring stock predicted return ratios $r^{T+1}_{[1:N]}$ on the day T+1, we jointly compute the point-wise regression and pairwise ranking loss with a weighting coefficient λ, to minimize the discrepancy between predicted $r^{T+1}_{[1:N]}$ and ground-truth $y^{T+1}_{[1:N]}$ meanwhile maintaining the relative order of stocks:

$$\mathcal{L}_{\mathcal{R}} = \sum_{i=1}^{N} \|r_i - y_i\|^2 + \lambda \sum_{i=1}^{N}\sum_{j=1}^{N} \max(0, -(r_i - r_j)(y_i - y_j)). \tag{15}$$

By integrating the ranking signals with conditional diffusion loss in Eq. (5), we formulate the final end-to-end training loss function, incorporating a weighting coefficient α:

$$\mathcal{L}_{task} = \mathcal{L}_{\mathcal{R}} + \alpha\mathcal{L}_{cond}. \tag{16}$$

Experiments

Experimental Setup

Datasets We examine DHMoE on three real-world datasets from the US and Chinese Exchange markets. Detailed statistics are presented in Tab. 2. The first dataset, NASDAQ, comprises 80,632 English stock news items related to 112 high-capitalization stocks from the popular NASDAQ market. The second dataset, NYSE, contains 85,162 English news items corresponding to 127 stocks listed on the New York Stock Exchange (NYSE), which is generally more stable compared to NASDAQ. The third dataset (Huang et al. 2018) Ashare&HK collects 162,976 news headlines from major financial websites in Chinese. It targets at 80 top-traded Ashare and HK stocks spanning Shanghai, Shenzhen, and Hong Kong Exchange markets.

Implementation Details Our model is implemented with PyTorch. The stock news data for NASDAQ and NYSE are sourced from the Kaggle[2]. We collect some missing daily quote data of all stocks including normalized *opening-high-low-closing* prices (OHLC) and *trading volumes* from professional Wind-Financial Terminal[3]. We follow (Feng et al. 2019b; Chen et al. 2024) and generate samples by moving a 16-day lookback window along trading days. For the proposed framework, the hidden dimension F is searched within $\{5, 10, 20, 30, 40, 50\}$ and finally set to 20. For DIT, we investigate the number of

[2]https://www.kaggle.com/datasets/miguelaenlle

[3]https://www.wind.com.cn/en/wft.html

	Methods		NASDAQ		NYSE		Ashare&HK	
			SR	IRR	SR	IRR	SR	IRR
CLF	Adv-ALSTM (Feng et al. 2019a)	Strengthen model robustness and prediction stability via temporal adversarial training	0.71	0.26	0.69	0.25	0.60	0.17
	StockNet (Xu and Cohen 2018)	Deep model using recurrent continuous latent variables and neural variational inference	0.46	0.11	0.34	0.05	0.59	0.16
	MAN-SF (Sawhney et al. 2020)	An architecture that jointly model temporal series, social media, and stock relations	0.77	0.29	0.88	0.36	0.42	0.07
	LSTM-RGCN (Li et al. 2021)	A framework that models stock relations from price data and incorporates overnight news	0.64	0.20	0.60	0.18	0.51	0.12
REG	SFM (Zhang, Aggarwal, and Qi 2017)	LSTM + DFT-based state frequency memory to extract multiple frequency patterns	0.55	0.15	0.40	0.06	0.37	0.05
	THGNN (Qin et al. 2017)	Temporal and heterogeneous GNN-based approach to learning the dynamic relations	0.60	0.18	<u>0.90</u>	<u>0.38</u>	0.91	0.35
	AMMoE (Li and Xu 2023)	Attention-based multi-gate MoE for quantitative stock selection	0.76	0.30	0.61	0.19	0.54	0.14
	AlphaMix (Sun et al. 2023)	Three-stage MoE to mimic the bottom-up hierarchical trading strategy	1.08	0.52	0.78	0.31	<u>0.98</u>	<u>0.38</u>
RL	iRDPG (Liu et al. 2020)	Use imitation learning to optimize trading policies for the reward of Sharpe Ratio	0.91	0.39	0.83	0.27	0.49	0.11
	RAT (Xu et al. 2021)	A RL framework that adapts Transformer with asset relations for portfolio trading	1.06	0.45	0.89	0.37	0.86	0.32
RAN	RSR (Feng et al. 2019b)	Temporal GCN using stock sequential embeddings and relations for stock ranking	1.04	0.44	0.51	0.13	0.57	0.15
	STHAN-SR (Sawhney et al. 2021a)	A hypergraph neural model integrating stock relations and temporal patterns for ranking	<u>1.10</u>	<u>0.53</u>	0.81	0.33	0.67	0.20
	HyperStockGAT (Sawhney et al. 2021b)	A hyperbolic graph attention network that learns scale-free market structure for ranking	0.73	0.27	0.64	0.21	0.76	0.25
	FAST (Sawhney et al. 2021c)	A hierarchical learning approach leverages textual data for time-aware stock ranking	0.86	0.32	0.50	0.11	0.79	0.29
	ADB-TRM (Chen et al. 2024)	Dual-stage temporal-relational debiasing framework for stock ranking	1.07	0.51	0.50	0.12	0.94	0.36
	DHMoE (Ours)	Diffusion generated hierarchical multi-granular expertise for stock ranking	**1.24**	**0.64**	**1.08**	**0.46**	**1.17**	**0.45**

Table 1: Profitability comparison with Classification (CLF), Regression (REG), Reinforcement Learning (RL), and Ranking (RAN) baselines. The improvement is statistically significant ($p < 0.01$) under Wilcoxon's signed rank test.

Datasets	Stocks	Train Days	Valid Days	Test Days
NASDAQ	112	2014/6/5-2018/6/4	2018/6/5-2019/6/4	2019/6/5-2020/6/5
NYSE	127	2014/6/5-2018/6/4	2018/6/5-2019/6/4	2019/6/5-2020/6/5
Ashare&HK	80	2014/1/1-2014/9/30	2014/10/1-2014/12/31	2015/1/1-2015/12/31

Table 2: Dataset statistics.

blocks, the embedded dimension, and the attention heads within the sets $\{1, 2, 4, 6, 8, 10\}$, $\{32, 64, 128, 256, 512\}$, and $\{2, 4, 6, 8, 16\}$. These parameters are determined to be 4 for the number of blocks, 256 for the embedded dimension, and 4 for the number of attention heads. The weighting coefficient λ and α are searched within $\{2, 4, 6, 8, 10, 12\}$ and $\{1, 2, 3, 4, 5, 6\}$, are finally set to 8 and 4, respectively. Besides, we adopt the following quadratic schedule for variance schedule: $\beta_k = (\frac{K-k}{K-1}\sqrt{\beta_1} + \frac{k-1}{K-1}\sqrt{\beta_K})^2$. We set the minimum noise level $\beta_1 = 0.0001$ and search the number of the diffusion step K and the maximum noise level β_K from a given parameter space ($K \in \{50, 100, 200\}$), and $\beta_K \in \{0.1, 0.2, 0.3, 0.4\}$). The K and β_K are set to 100 and 0.2. The number of the middle-level and bottom-level experts a, b are searched within $\{3, 6, 9, 12, 15, 18\}$ and $\{3, 6, 9, 12, 15, 18\}$ and are finally set to 9 and 6, respectively. We conduct experiments on four GeForce RTX 3090 GPUs by AdamW optimizer (Loshchilov and Hutter 2019) for 30 epochs, the learning rate is set to 1e-3, and the batch size is set to 4.

Metrics Building upon the foundations laid by previous research (Sawhney et al. 2021c,b; Chen et al. 2024), we employ a daily buy-hold-sell trading strategy to evaluate the prediction and investment efficacy of DHMoE, specifically focusing on Sharpe Ratio (SR) and cumulative investment return ratio (IRR) performance metrics. Essentially, this strategy involves purchasing κ stocks that are forecasted to yield the highest returns at the close of the market on day t, followed by selling these shares at the close of the next trading day. Formally expressed, the IRR for day t is calculated as $\mathrm{IRR}^t = \sum_{i \in \hat{\mathcal{S}}^t} \frac{p_i^{t+1} - p_i^t}{p_i^t}$, with $\hat{\mathcal{S}}^t$ representing the selected stocks in the portfolio for day t. Moreover, the Sharpe Ratio is defined as $\mathrm{SR} = \frac{E[R_p] - R_f}{\mathrm{std}[R_p]}$, which quantifies the risk-adjusted return, delineating the excess return per unit of risk assumed. Furthermore, we assess the model's ranking proficiency using the widely recognized metric, nDCG@κ, and present the average outcomes of five distinct trials for $\kappa = 5$ in Tab. 1.

Overall Performance

We consider four categories of baselines for comparison. The results are shown in Tab. 1, from which we have several observations: **(1)** By integrating the MoE architecture with real-world trading paradigms, our model achieves the best results across all datasets. Specifically, it fetches an average relative performance gain of 17.37% and 20.08% in regard to risk-adjusted returns and cumulative profits over the best baselines. **(2)** Our approach diverges from previous multimodal methods concentrating on heterogeneous feature processing and fusion. Instead, we focus on integrating top-down teaching and bottom-up decision-making real-world paradigms with MoE architecture for multimodal prediction. Specifically, we employ heterogeneous data as conditional information to generate parameters for bottom-level experts. The final bottom-level experts are tailored to specific modalities in conditional generation for prediction and decision-making. **(3)** Beyond achieving commendable performance, our method also exhibits superior scalability in comparison to the single neural network predictor. In DHMoE, the number of middle-level and bottom-level experts can be adjusted according to the requirement for processing the source data or features. This flexibility allows for more efficient allocation of computational resources. Furthermore, our method also shows a notable performance improvement compared to the MoE approach that solely depends on temporal information (i.e., AMMoE and AlphaMix). This highlights the importance of equipping the model with the capability to mine heterogeneous data, especially in the intricate stock markets.

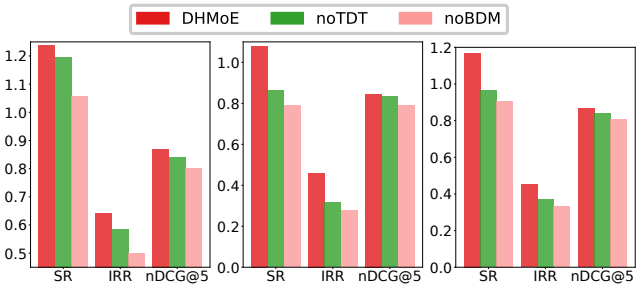

Figure 3: Ablation study over two real-world paradigms (*top-down teaching* and *bottom-up decision making*) on NASDAQ (left), NYSE (middle), and Ashare&HK (right).

Replace Models	NASDAQ		NYSE		Ashare&HK	
	SR	IRR	SR	IRR	SR	IRR
Transformer	1.07	0.49	0.92	0.39	0.91	0.33
iTransformer	1.14	0.55	0.96	0.41	0.97	0.36
DIT	**1.24**	**0.64**	**1.08**	**0.46**	**1.17**	**0.45**

Table 3: Comparison of the results of replacing DIT with different parameter generation models on different datasets.

In-depth Analysis

Ablation Study Through ablation experiments, we investigate the effects of top-down teaching, bottom-up decision-making, and DIT on the overall performances. During ablation studies on the top-down teaching (*TDT*) paradigm, we ceased the transfer of parameters from top to bottom, allowing bottom-level experts to independently initialize their parameters. In the ablation experiments targeting the bottom-up decision-making (*BDM*) paradigm, we refrain from executing a bottom-up opinion-aggregation process. Instead, we directly consolidate the insights of bottom-level experts using a linear layer. Moreover, in evaluating the role of DIT, we substituted it with various other parameter-generation modules. We depict the results in Fig. 3 and Tab. 3. As shown, different components jointly contribute to the performance. The ablation study shows incremental gain from top-down teaching, bottom-up decision-making, and DIT. The primary benefit stems from the *bottom-up decision-making* paradigm, which can effectively aggregate prediction results from various experts hierarchically, thus ensuring the model's robustness. Furthermore, it is noteworthy that employing DIT as the top-level expert yields the best outcomes, emphatically showcasing the viability and efficacy of leveraging the potent pattern mining and generation capabilities of diffusion models.

Parameter Sensitivity The comprehensive hyperparameter experiments are depicted in Fig. 4. Our first concern is the impact of changes in weighting coefficient λ and α on the model's performance. The loss function for training the model consists of three main components: loss $\mathcal{L}_{cond}$ from the denoising process, and loss $\mathcal{L}_{\mathcal{R}}$, which is a combination of prediction and ranking loss. Loss $\mathcal{L}_{cond}$ guides the DIT in generating an effective parameter set for lower-level experts, whereas loss $\mathcal{L}_{\mathcal{R}}$ focuses on refining the model's overall predictive accuracy and profitability. Thus, the values of α and λ play a pivotal role in determining the overall optimization target of DHMoE. It's observed that excessively small or large values for α and λ tend to disrupt the balance between the model's training objectives, leading to a decline in its overall performance. The experiments about a and b demonstrate that a minimal number of experts results in suboptimal outcomes. As the number of experts grows, performance fluctuates but generally improves significantly compared to scenarios with fewer experts. This indicates that a limited number of experts might lead to inadequate predictive capacity and substantial prediction variation. The experiments concerning hyperparameters F and Z offer more straightforward insights. When the values of F and Z are too small, the model may suffer from a scarcity of parameters, leading to inadequate predictive capabilities. Conversely, excessively high values can result in the overfitting issue.

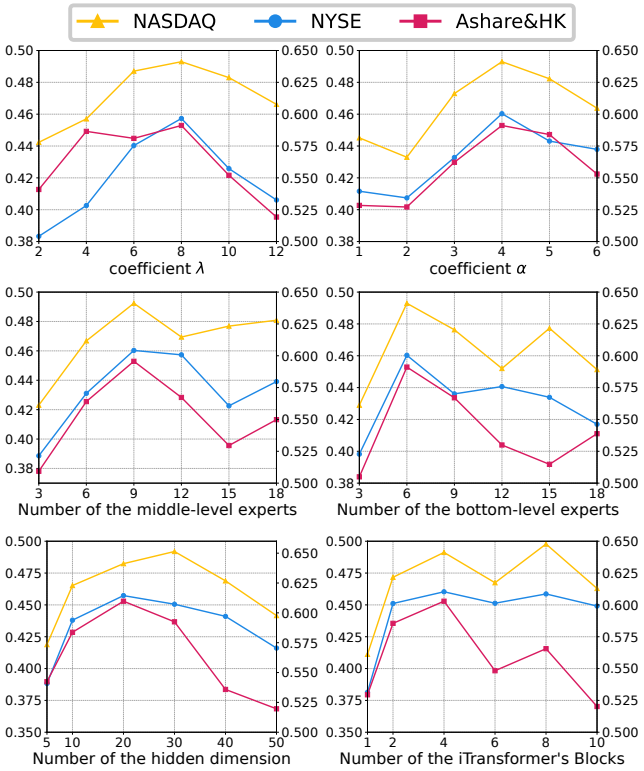

Figure 4: The hyperparameter experiments on the weighting coefficient λ, α, number of the middle-level and bottom-level experts a, b, hidden dimension F, and the number of the iTransformer's blocks Z. The y-axis values in the above figures represent the investment return ratio (IRR) metric. The right y-axis corresponds to the NASDAQ dataset, while the left y-axis pertains to the other two datasets.

Attention Visualization In the bottom-up decision-making paradigm, the SSAM is utilized to consolidate the insights from various experts into the final investment decision. Therefore, we delve into the attention weights of SSAM to obtain a more profound comprehension of DHMoE's capability for quantitative investment. Fig. 5

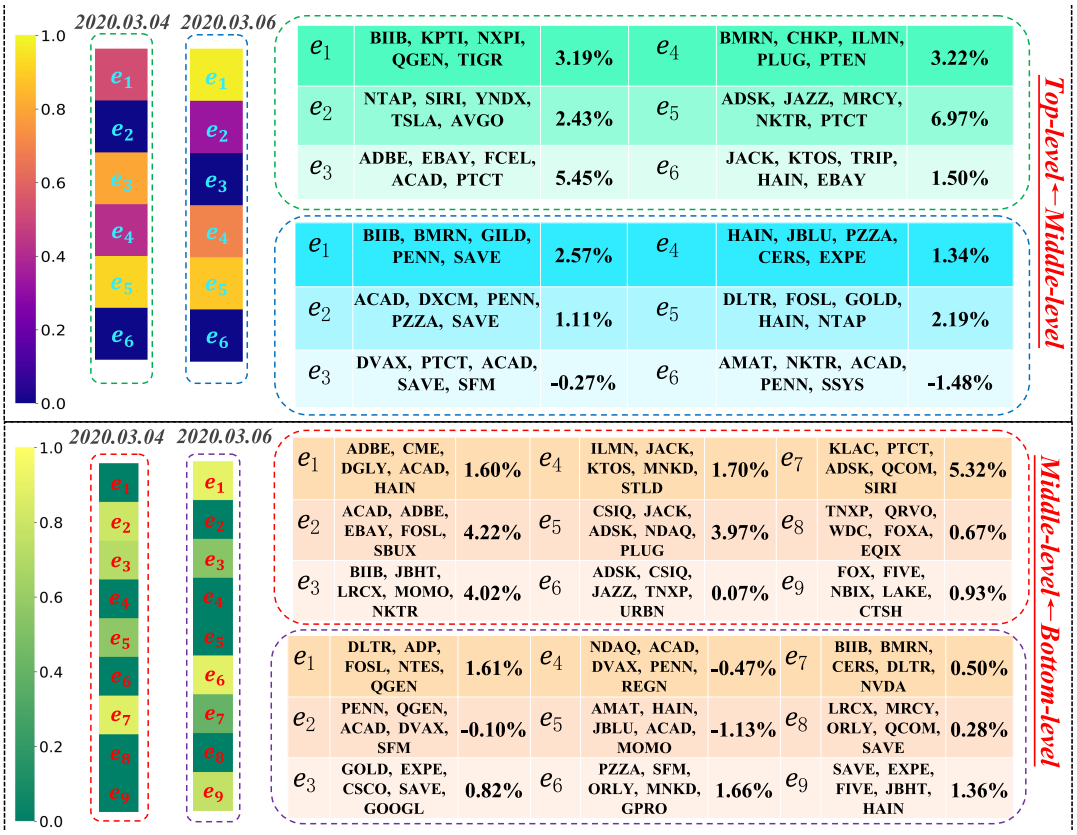

Figure 5: The illustration depicts the SSAM applied to the NASDAQ on 2020.03.04 and 2020.03.06. The table within the figure shows triplets that include the expert number, the stock code invested in, and the total return on investment. The lower figure illustrates the synthesized process from the bottom to the middle level, whereas the upper figure depicts the process from the middle to the top level.

displays the attention weights for each bottom-level and middle-level expert, along with details of the selected Top-5 stocks and the corresponding investment returns. This visual evidence suggests that our attention mechanism efficiently identifies experts who are adept at making profitable investments. Besides, the investment returns of middle-level experts generally surpass those of the bottom-level experts, highlighting the success of our bottom-up decision-making paradigm. Additionally, the profit table indicates that most experts realize commendable investment returns, proving the efficacy of our top-down teaching paradigm.

Conclusion

In our paper, we pioneer in integrating the real-world paradigms of top-down teaching and bottom-up decision-making with the MoE architecture tailored for stock prediction. Our DHMoE model seamlessly combines these paradigms within a tree structure and leverages the robust representational learning ability of the diffusion model at the top level to generate parameters for lower-level experts. For bottom-up decision-making, we introduce a sparse synthesized attention mechanism designed for aggregating experts' opinions hierarchically. Our experiments span three real-world datasets, encompassing four major categories and fifteen methods for comparison. The findings reveal that our model surpasses state-of-the-arts. Furthermore, detailed analyses affirm the effectiveness of our method in enhancing the model's profitability within the complex stock market.

References

Ang, G.; and Lim, E.-P. 2022. Guided attention multimodal multitask financial forecasting with inter-company relationships and global and local news. In *ACL*, 6313–6326.

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Chen, W.; Li, S.; Yu, X.; Wang, H.; Chen, W.; and Wang, T. 2024. Automatic De-Biased Temporal-Relational Modeling for Stock Investment Recommendation. In *IJCAI*, 1999–2008.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Croitoru, F.-A.; Hondru, V.; Ionescu, R. T.; and Shah, M. 2023. Diffusion models in vision: A survey. *TPAMI*.

Ding, Q.; Wu, S.; Sun, H.; Guo, J.; and Guo, J. 2020. Hierarchical Multi-Scale Gaussian Transformer for Stock Movement Prediction. In *IJCAI*, 4640–4646.

Feng, F.; Chen, H.; He, X.; Ding, J.; Sun, M.; and Chua, T.-S. 2019a. Enhancing Stock Movement Prediction with Adversarial Training. *IJCAI*.

Feng, F.; He, X.; Wang, X.; Luo, C.; Liu, Y.; and Chua, T.-S. 2019b. Temporal relational ranking for stock prediction. *TOIS*, 37(2): 1–30.

Feng, S.; Xu, C.; Zuo, Y.; Chen, G.; Lin, F.; and XiaHou, J. 2022. Relation-aware dynamic attributed graph attention network for stocks recommendation. *Pattern Recognition*, 121: 108119.

Gao, Y.; Chen, H.; Wang, X.; Wang, Z.; Wang, X.; Gao, J.; and Ding, B. 2024. DiffsFormer: A Diffusion Transformer on Stock Factor Augmentation. *arXiv preprint arXiv:2402.06656*.

Hsu, Y.-L.; Tsai, Y.-C.; and Li, C.-T. 2021. FinGAT: Financial Graph Attention Networks for Recommending Top-K K Profitable Stocks. *TKDE*, 35(1): 469–481.

Huang, H.; Chen, M.; and Qiao, X. 2024. Generative Learning for Financial Time Series with Irregular and Scale-Invariant Patterns. In *ICLR*.

Huang, J.; Zhang, Y.; Zhang, J.; and Zhang, X. 2018. A tensor-based sub-mode coordinate algorithm for stock prediction. In *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, 716–721. IEEE.

Jacobs, R. A.; Jordan, M. I.; Nowlan, S. J.; and Hinton, G. E. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1): 79–87.

Kenton, J. D. M.-W. C.; and Toutanova, L. K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, 4171–4186.

Koa, K. J.; Ma, Y.; Ng, R.; and Chua, T.-S. 2023. Diffusion Variational Autoencoder for Tackling Stochasticity in Multi-Step Regression Stock Price Prediction. In *CIKM*, 1087–1096.

Li, K.; and Xu, J. 2023. An Attention Based Multi-gate Mixture-of-Experts Model for Quantitative Stock Selection. *International Journal of Trade, Economics and Finance*, 14(3).

Li, W.; Bao, R.; Harimoto, K.; Chen, D.; Xu, J.; and Su, Q. 2021. Modeling the stock relation with graph network for overnight stock movement prediction. In *IJCAI*, 4541–4547.

Lin, L.; Li, Z.; Li, R.; Li, X.; and Gao, J. 2023. Diffusion models for time-series applications: a survey. *Frontiers of Information Technology & Electronic Engineering*, 1–23.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2024. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *ICLR*.

Liu, Y.; Liu, Q.; Zhao, H.; Pan, Z.; and Liu, C. 2020. Adaptive quantitative trading: An imitative deep reinforcement learning approach. In *AAAI*, volume 34, 2128–2135.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *ICLR*.

Ma, J.; Zhao, Z.; Yi, X.; Chen, J.; Hong, L.; and Chi, E. H. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *KDD*, 1930–1939.

Martins, A.; and Astudillo, R. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *ICML*, 1614–1623. PMLR.

Peebles, W.; and Xie, S. 2023. Scalable diffusion models with transformers. In *ICCV*, 4195–4205.

Qin, Y.; Song, D.; Cheng, H.; Cheng, W.; Jiang, G.; and Cottrell, G. W. 2017. A dual-stage attention-based recurrent neural network for time series prediction. In *IJCAI*, 2627–2633.

Rezaei, H.; Faaljou, H.; and Mansourfar, G. 2021. Stock price prediction using deep learning and frequency decomposition. *Expert Systems with Applications*, 169: 114332.

Sawhney, R.; Agarwal, S.; Wadhwa, A.; Derr, T.; and Shah, R. R. 2021a. Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach. In *AAAI*, volume 35, 497–504.

Sawhney, R.; Agarwal, S.; Wadhwa, A.; and Shah, R. 2020. Deep attentive learning for stock movement prediction from social media text and company correlations. In *EMNLP*, 8415–8426.

Sawhney, R.; Agarwal, S.; Wadhwa, A.; and Shah, R. 2021b. Exploring the scale-free nature of stock markets: Hyperbolic graph learning for algorithmic trading. In *WWW*, 11–22.

Sawhney, R.; Wadhwa, A.; Agarwal, S.; and Shah, R. 2021c. FAST: Financial news and tweet based time aware network for stock trading. In *EACL*, 2164–2175.

Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; and Dean, J. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Sun, S.; Wang, X.; Xue, W.; Lou, X.; and An, B. 2023. Mastering Stock Markets with Efficient Mixture of Diversified Trading Experts. In *KDD*, 2109–2119. New York, NY, USA: Association for Computing Machinery. ISBN 9798400701030.

Tancik, M.; Srinivasan, P.; Mildenhall, B.; Fridovich-Keil, S.; Raghavan, N.; Singhal, U.; Ramamoorthi, R.; Barron, J.; and Ng, R. 2020. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 33: 7537–7547.

Tashiro, Y.; Song, J.; Song, Y.; and Ermon, S. 2021. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *NeurIPS*, 34: 24804–24816.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *NeurIPS*, 30.

Wang, C.; Chen, Y.; Zhang, S.; and Zhang, Q. 2022a. Stock market index prediction using deep Transformer model. *Expert Systems with Applications*, 208: 118128.

Wang, H.; Li, S.; Wang, T.; and Zheng, J. 2021. Hierarchical Adaptive Temporal-Relational Modeling for Stock Trend Prediction. In *IJCAI*, 3691–3698.

Wang, H.; Wang, T.; Li, S.; Guan, S.; Zheng, J.; and Chen, W. 2022b. Heterogeneous Interactive Snapshot Network for Review-Enhanced Stock Profiling and Recommendation. In *IJCAI*, 3962–3969.

Wang, H.; Wang, T.; Li, S.; Zheng, J.; Guan, S.; and Chen, W. 2022c. Adaptive Long-Short Pattern Transformer for Stock Investment Selection. In *IJCAI*, 3970–3977.

Wang, H.; Wang, T.; and Li, Y. 2020. Incorporating expert-based investment opinion signals in stock prediction: A deep learning framework. In *AAAI*, volume 34, 971–978.

Wang, J.-H.; and Leu, J.-Y. 1996. Stock market trend prediction using ARIMA-based neural networks. In *ICNN*, volume 4, 2160–2165. IEEE.

Wen, H.; Lin, Y.; Xia, Y.; Wan, H.; Wen, Q.; Zimmermann, R.; and Liang, Y. 2023. Diffstg: Probabilistic spatio-temporal graph forecasting with denoising diffusion models. In *SIGSPATIAL*, 1–12.

Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019. Simplifying graph convolutional networks. In *ICML*, 6861–6871. PMLR.

Xu, K.; Zhang, Y.; Ye, D.; Zhao, P.; and Tan, M. 2021. Relation-aware transformer for portfolio policy learning. In *IJCAI*, 4647–4653.

Xu, Y.; and Cohen, S. B. 2018. Stock movement prediction from tweets and historical prices. In *ACL*, 1970–1979.

Yang, L.; Zhang, Z.; Song, Y.; Hong, S.; Xu, R.; Zhao, Y.; Zhang, W.; Cui, B.; and Yang, M.-H. 2023a. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4): 1–39.

Yang, M.; Zhu, M.; Liang, Q.; Zheng, X.; and Wang, M. 2023b. Spotlight news driven quantitative trading based on trajectory optimization. In *IJCAI*, 4930–4939.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are transformers effective for time series forecasting? In *AAAI*, volume 37, 11121–11128.

Zhang, L.; Aggarwal, C.; and Qi, G.-J. 2017. Stock price prediction via discovering multi-frequency trading patterns. In *KDD*, 2141–2149.

Zhang, Q.; Qin, C.; Zhang, Y.; Bao, F.; Zhang, C.; and Liu, P. 2022. Transformer-based attention network for stock movement prediction. *Expert Systems with Applications*, 202: 117239.

Zhao, Y.; Du, H.; Liu, Y.; Wei, S.; Chen, X.; Zhuang, F.; Li, Q.; and Kou, G. 2022. Stock Movement Prediction Based on Bi-Typed Hybrid-Relational Market Knowledge Graph Via Dual Attention Networks. *TKDE*.

Zheng, Z.; Shao, J.; Zhu, J.; and Shen, H. T. 2023. Relational Temporal Graph Convolutional Networks for Ranking-Based Stock Prediction. In *ICDE*, 123–136. IEEE.

Zivot, E.; and Wang, J. 2006. Vector autoregressive models for multivariate time series. *Modeling financial time series with S-PLUS®*, 385–429.