Can Large Language Models Mine Interpretable Financial Factors More Effectively? A Neural-Symbolic Factor Mining Agent Model

Zhiwei Li[1], Ran Song[2], Caihong Sun[1] *; Wei Xu[1,3], Zhengtao Yu[2], Ji-Rong Wen[1,4]

[1] School of Information, Renmin University of China, Beijing, China
[2] Faculty of Information Engineering and Automation,
Kunming University of Science and Technology, Kunming, China
[3] School of Smart Governance, Renmin University of China, Beijing, China
[4] Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
{lizhiwei2,chsun,weixu,jrwen}@ruc.edu.cn,
song_ransr@163.com, ztyu@hotmail.com

Abstract

Finding interpretable factors for stock returns is the most vital issue in the empirical asset pricing domain. As data-driven methods, existing factor mining models can be categorized into symbol-based and neural-based models. Symbol-based models are interpretable but inefficient, while neural-based approaches are efficient but lack interpretability. Hence, mining interpretable factors effectively presents a significant challenge. Inspired by the success of Large Language Models (LLMs) in various tasks, we propose a FActor Mining Agent (FAMA) model that enables LLMs to integrate the strengths of both neural and symbolic models for factor mining. In this paper, FAMA consists of two main components: Cross-Sample Selection (CSS) and Chain-of-Experience (CoE). CSS addresses the homogeneity challenges in LLMs during factor mining by assimilating diverse factors as in-context samples, whereas CoE enables LLMs to leverage past successful mining experiences, expediting the mining of effective factors. Experimental evaluations on real-world stock market data demonstrate the effectiveness of our approach by surpassing the SOTA RankIC by 0.006 and RankICIR by 0.105 in predicting S&P 500 returns. Furthermore, the investment simulation shows that our model can achieve superior performance with an annualized return of 38.4% and a Sharpe ratio of 667.2%.

1 Introduction

The task of predicting market trends in finance presents a formidable challenge, given the intricate interplay of various factors (Hou et al., 2011), such as the dynamics of demand and supply (Hendricks and Singhal, 2009), market sentiment (Verma and Soydemir, 2009) and government regulations (Ali Imran et al., 2020). In the field of quantitative trading, the conventional approaches often extract factors as indicative signals for market trends from raw historical stock data first, then serve them as input features for machine learning models (Sharpe, 1964; Ross, 2013; Duan et al., 2022). A pivotal step in this process entails discerning and extracting effective factors that demonstrate robust predictive capabilities for market trends (Ng et al., 1992). As an illustrative example, the Capital Asset Pricing Model (CAPM) (Sharpe, 1964) employed the market's excess return as a predictive factor for the return of a financial asset, thereby contributing a seminal factor to finance.

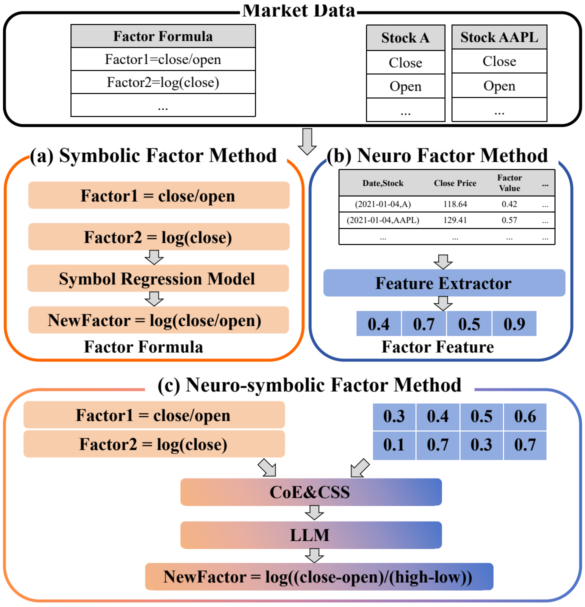

Figure 1: An illustration of three distinct factor mining approaches: (a) symbolic factor model, (b) neural factor model, and (c) our proposed neural-symbolic model.

Hence discovering factors with high returns has been a trendy topic among investors and researchers. The prevailing methods for mining factors can be in general divided into two groups,

[*] Corresponding author

Findings of the Association for Computational Linguistics: ACL 2024, pages 3891–3902
August 11-16, 2024 ©2024 Association for Computational Linguistics

namely symbolic factor and neural factor models. As illustrated in Figure 1(a), in symbolic factor models, factors are represented as symbolic expressions, then symbolic regression (Makke and Chawla, 2024) serves as a common technique for factor mining (Jin et al., 2019; Zhang et al., 2020; Chen et al., 2021; Cui et al., 2021). For instance, considering two factors, $Factor1$ = "$close/open$" and $Factor2$ = "$\log(close)$", the factor values are calculated by the opening and closing price, then the two factors are inputted into a symbolic regression model to generate a novel factor, $NewFactor$ = "$\log(close/open)$". The interpretability of the symbolic factor model arises from the explicit representation of the calculation process for the factors. However, due to the vast search space of symbolic factors, mining with symbolic factor models often proves inefficient. Conversely, neural factor approaches, gaining recent popularity, transform factors into numerical features to optimize factor extraction. As depicted in Figure 1(b), neural factor models predict market trends by extracting numerical factor features from stock data through feature extractors (Kelly et al., 2019; Gu et al., 2021; Duan et al., 2022). Compared with symbolic factor models, neural factor models exhibit proficiency in extracting effective numerical factors. However, the financial interpretability in neural factor models struggles with implicit features. The question we are facing is: Can an **effective** approach be devised for mining **financially interpretable** factors conducive to predicting market trends?

Recent advancements in LLMs have demonstrated success across financial tasks, including sentiment analysis (Guo et al., 2023) and financial text generation (Yang et al., 2023b). Thanks to its powerful In-Context Learning (ICL) ability (Brown et al., 2020), we conceptualize LLMs as a neuro-symbolic model illustrated in Figure 1(c) that bridges two distinct representations, i.e., numerical and symbolic factors, aiming to achieve efficient mining of interpretable ones. It is facile to consider utilizing the factors disclosed (Kakushadze, 2016) as contextual examples to generate new factors through ICL. Since the disclosed factors are often limited in number, high correlation, and low complexity, direct mining factors using ICL encounter challenges. These issues can be summarized in two aspects: (1) The heightened homogeneity observed among factors, characterized by the uniform structure, culminates in the generation of the singular factor form through ICL. (2) The presence of a noteworthy proportion of ineffective factors acts as an impediment, hindering ICL from effectively exploring novel patterns. Therefore, the efficacy of mining effective factors using LLMs is contingent upon selecting diversity-guiding factors as contextual samples to mitigate homogeneity. Additionally, encouraging ICL to explore new patterns is key to increasing the proportion of effective factors.

In this paper, we present the FActor Mining Agent (FAMA), consisting of two main parts: Cross-Sample Selection (CSS) and Chain-of-Experience (CoE) methods. CSS is designed to ensure the diversification of factor mining by amalgamating low correlation classes of factors as contextual samples, which empowers LLMs to incorporate diversity-guiding factors and mitigate the homogeneity of mined factors. CoE efficiently encourages ICL to explore new paradigms by incorporating the paths of mining effective factors as experiential prompts, which contributes to the further optimization of factor mining in LLMs. Our experimental results show better performance of our model in predicting stock market returns compared to previous approaches. Moreover, our model also demonstrates a superior annualized return and Sharpe ratio in the investment simulations.

Our contributions can be summarized as follows:

- To the best of our knowledge, we are the first ones to use LLMs as a bridge between symbolic and neural representations in the task of factor mining.
- We propose a factor mining agent (FAMA) to facilitate LLMs as factor miners, in which its components CSS and CoE are designed to tackle homogeneity issues and encourage ICL in exploring new directions, respectively.
- We expand the capabilities of LLMs to perform factor mining tasks and present a series of experiments to demonstrate the effectiveness of our proposed model.

2 Problem Formulation

2.1 Financial Factor

Consider a stock dataset for N stocks over T trading days. The features of all stocks are denoted as $\mathbf{X} = [x_1, x_2, ..., x_N]$. Consider the M features, such as open and close prices, pertaining to each stock j, denoted as $x_j \in \mathbb{R}^{M \times T}$. We define the factor space as $\mathcal{F}$, where each factor $f \in \mathcal{F}$ is defined as $f : \mathbb{R}^{M \times T} \rightarrow \mathbb{R}^T$. The value of factor f on stock j is defined as $f(x_j) \in \mathbb{R}^T$. To conveniently represent the symbolic form of factors, we employ the symbol function $s(f)$ to denote the symbolic text of factor f_i. For example, $s(f_{101}) =$ "$((close - open)/(high - low))$".

2.2 Factor Distance and Correlation

In practice, factor categorization has traditionally depended on artificial classification rooted in financial principles, such as momentum (Carhart, 1997) and trend (Han et al., 2016) factors. Despite the demonstrated high accuracy associated with this approach, it involves a labor-intensive process. To enhance the efficiency of factor classification, we advocate for a quantitative exploration of correlations among factors. We consider the factor space $\mathcal{F}$ is equipped with a distance mapping $d : \mathcal{F} \times \mathcal{F} \rightarrow \mathbb{R}$, thereby establishing it as a complete distance space $(\mathcal{F}, d)$, then correlations between factors can be defined within this space $(\mathcal{F}, d)$ as $r : \mathcal{F} \times \mathcal{F} \rightarrow [-1, 1]$. This approach enables a more efficient analysis of factor correlations without a labor-intensive process.

2.3 Factor Mining

The goal of factor mining is to produce a new set of factors $F \subset \mathcal{F}$ that will lead to better predictive performance of stocks in their portfolios. To evaluate the predictive performance of factors, we employ the Rank Information Coefficient (RankIC). RankIC measures the correlation between a factor's ranking in equity exposure and its subsequent return ranking. The RankIC on period t and average RankIC γ is defined as follows:

$$RankIC_t(f, r_j) = Corr(order^f_{t-1}, order^{r_j}_t),$$
$$\gamma(f) = \frac{1}{N}\frac{1}{T}\sum_{j=1}^{N}\sum_{t=1}^{T} RankIC_t(f, r_j), \tag{1}$$

where $order^f_{t-1}$ signifies the factor value ranking at time $t-1$, and $order^{r_j}_t$ represents the return ranking of stock j at time t, with $Corr(x, y)$ denoting the correlation coefficient between vectors x and y. Given a factor set $F = \{f_1, ..., f_p\}$, its effectiveness is assessed by computing the average RankIC of the factors within the set, as described below:

$$\overline{\gamma}(F) = \mathbb{E}_i[\gamma(f_i)], f_i \in F. \tag{2}$$

The objective of factor mining is to commence with an initial set of factors, denoted as F_0, and iteratively mine new factors to enhance the overall predictive performance. This process involves a total of m mining iterations. The set of factors obtained after the i-th mining iteration is represented as F_i. Therefore, we aim for the sequence of average predictive performance metrics to satisfy the following relation:

$$\overline{\gamma}(F_0) \leq \overline{\gamma}(F_1) \leq \cdots \leq \overline{\gamma}(F_m) \tag{3}$$

3 Factor Mining Agent

As illustrated in Figure 2, our proposed FActor Mining Agent (FAMA) consists of two main parts: (1) Cross-Sample Selection (CSS) and (2) Chain-of-Experience (CoE). FAMA improves the mining factor effectiveness through iterative mining. In each iteration, FAMA generates diversity guiding factors via CSS and empirical paths through CoE as prompts fed into LLMs for mining factors.

3.1 Definitions

To measure the distance and correlation between factors quantitatively mentioned in Section 2.2, we start with calculating the weighted average price of the stock pool. It is defined as:

$$\mathbf{p} = \mathbf{w}\mathbf{X}, \tag{4}$$

where $\mathbf{w} \in \mathbb{R}^n$ denotes the total market value weight corresponding to the company's stock. Subsequently, we calculate the factor exposure $\mathbf{v}_i$ of factor f_i at the weighted average price p and employ z-score normalization as:

$$\mathbf{v_i} = \frac{f_i(\mathbf{p}) - mean(f_i(\mathbf{p}))}{std(f_i(\mathbf{p}))}. \tag{5}$$

Consequently, we define the distance between two factors as:

$$d(f_i, f_j) = \|\mathbf{v_i} - \mathbf{v_j}\|_2. \tag{6}$$

Then, the correlation coefficient between the factors is defined as:

$$\begin{aligned} r(f_i, f_j) &= \text{Corr}(\mathbf{v_i}, \mathbf{v_j}) \\ &= \frac{\sum_{t=1}^{T}(\mathbf{v}_{it} - \overline{\mathbf{v}}_i)(\mathbf{v}_{jt} - \overline{\mathbf{v}}_j)}{\sum_{t=1}^{T}(\mathbf{v}_{it} - \overline{\mathbf{v}}_i)^2(\mathbf{v}_{jt} - \overline{\mathbf{v}}_j)^2}. \end{aligned} \tag{7}$$

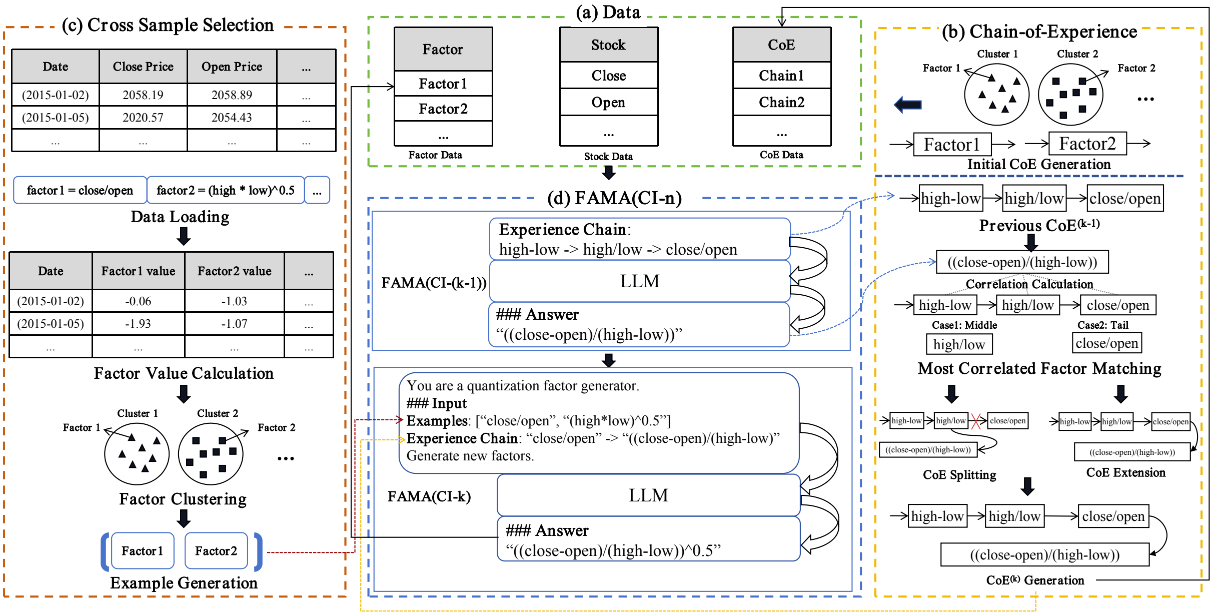

Figure 2: An overview of the FAMA model. FAMA(CI-n) denotes the nth iteration of the FAMA model. Initially, (a) the input factors, stock data, and experience chain data are fed into the FAMA model. Subsequently, (b) the CoE module utilizes the outcomes of FAMA(CI-$(k-1)$) to produce a novel CoE^k, and incorporates the diverse guidance factors generated by the (c) CSS module to formulate a prompt. Lastly, the prompt is fed into the LLMs to mine a new factor of FAMA(CI-k) as illustrated in (d), which is then stored in the factor database.

3.2 Cross-Sample Selection

The CSS selects low-correlation guiding factors as contexts thereby avoiding homogeneity of the generated factors. It categorizes the factors into different classes, sampling from the classes to get a context sample of diversity factors. Here, we propose a clustering algorithm based on KMeans (Krishna and Murty, 1999) for factor clustering. The factor value $\mathbf{v}_i$ of factor f_i obtained from Equation 5 is used for clustering. Initially, we randomly select k factor values as clustering centers $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_k\}$. For each factor value $\mathbf{v}_i$, its class is calculated as $c(f_i) = argmin_j \left\| \mathbf{v}_i - \boldsymbol{\mu}_j \right\|^2$. Subsequently, we update the clustering center using the formula:

$$\boldsymbol{\mu}_j = \frac{1}{\sum\limits_{c(f_i)=j} 1} \sum_{c(f_i)=j} \mathbf{v}_i. \tag{8}$$

We define the loss of the factor cluster model as:

$$J = \sum_{i=1}^{k} \sum_{c(f_j)=i} \left\| \mathbf{v}_j - \boldsymbol{\mu}_i \right\|^2. \tag{9}$$

The optimal classification is defined as:

$$c^* = \underset{\mathbf{c}}{argmin}\, J. \tag{10}$$

We denote the set of factors C_i belonging to the same class i as:

$$C_i = \{f_j |\ c^*(f_j) = i\}. \tag{11}$$

Subsequently, we randomly draw a sample f^i from each category C_i to get a factor combination:

$$FC = [f^1, f^2, \cdots, f^k],\ f^i \in C_i. \tag{12}$$

Finally, $l (l \leq k)$ factors in the factor combination FC are selected as context samples:

$$S = [s(f^{i_1}), s(f^{i_2}), \cdots, s(f^{i_l})], f^{i_j} \in FC. \tag{13}$$

3.3 Chain-of-Experience

This part aims to involve past successful mining experiences in ICL to facilitate factor mining effectiveness. The generation of experience chains is divided into two phases: the initial generation phase and the enhanced generation phase. In the initial phase, we employ the initial set of factors for generation. Following the acquisition of the previous factor clustering results C_i through Equation 11 and with the size of C_i denoted as p_i, the initial experience chain for category C_i is generated. This generation process relies on the ranking result of γ as defined in Equation 1, which can be described as follows:

$$
\begin{aligned}
&CoE_i^0 = s(f_1^{(i)}) \rightarrow s(f_2^{(i)}) \rightarrow \cdots \rightarrow s(f_{p_i}^{(i)}), \\
&\gamma(f_1^{(i)}) \leq \gamma(f_2^{(i)}) \leq \cdots \leq \gamma(f_{p_i}^{(i)}).
\end{aligned}
\tag{14}
$$

In the enhanced phase, the experience chain utilized in the previous step is denoted as $CoE_i^{(k-1)}$. We choose ICL-generated factor $f^{*(i)}$ with $CoE_i^{(k-1)}$ having a higher γ than all chain factors. Then, we compute the correlation r defined in Equation 7 for the new factor $f^{*(i)}$ and factors on $CoE_i^{(k-1)}$ to get the highest correlation factor $f_h^{(i)}$. If the matched factor $f_h^{(i)}$ is at the end of the chain C_i, the new factor is treated as an extension of the experience. Otherwise, the new factor $f^{*(i)}$ represents a new experience, and it is introduced into the chain subsequent to a split triggered by the matching factor $f_h^{(i)}$. This process can be defined uniformly as:

$$
\begin{aligned}
&\mathrm{CoE}_i^k = s(f_1^{(i)}) \rightarrow \cdots \rightarrow s(f_h^{(i)}) \rightarrow s(f^{*(i)}), \\
&r(f_h^{(i)}, f^{*(i)}) \geq r(f_j^{(i)}, f^{*(i)}), \forall 0 \leq j \leq p_i.
\end{aligned}
\tag{15}
$$

Our proposed Factor Mining Agent (FAMA) synergistically integrates the Cross-Sample Selection (CSS), as outlined in Section 3.2, and the Chain-of-Experience (CoE), explicated in Section 3.3, thereby automating the generation of diversity-guiding factor samples and experience chains for iterative factor mining. Within each iteration, an initial step involves clustering the factors within the collection into distinct categories, serving as preparation for subsequent CSS and CoE processes. We then proceed by leveraging the samples generated through CSS as in-context examples. Subsequently, we select the most pertinent experience chain. Finally, we amalgamate these components into a prompt, which is subsequently inputted into the LLM to generate new factors. If the resultant factor demonstrates improved performance metrics, it will be incorporated into the experience chain. This inclusive step fosters the development of a new experience chain for subsequent iterations, thus nurturing the iterative refinement process. The specific operational framework of FAMA is delineated in Algorithm 1. Detailed elucidations of the functions integral to the algorithmic execution are provided in Appendix A.

Algorithm 1: Factor Mining Agent

```
Data: Initial factor set $F_0 = \{f_1, \cdots, f_n\}$, number
      of mining $m$, minimum factor number $u$.
Result: Final factor set $F_m$, experience chain set
        $CoE^m$.
Generate initial experience chain set
    $CoE^0 = \{e_1, \cdots, e_k\}$; // 14
for $i \leftarrow 1$ to $m$ do
    Initialize new factor set $F_i$ and chain set $CoE^i$;
    $C \leftarrow$ Cluster($F_{i-1}$);// 11
    $S \leftarrow$ SelectSamples($C$);                  // 13
    foreach $s \in S$ do
        $e \leftarrow$ MatchCoE($s$, $CoE^{(i-1)}$);
        $prompt \leftarrow s + e$;
        $f' \leftarrow$ LLM($prompt$);
        if $\gamma(f') > max(\gamma(f)), \forall f \in e$ then
            $e' \leftarrow$GenChain($e$,$f'$);          // 15
            $CoE^i \leftarrow CoE^i \cup \{e'\}$;
        $F_i \leftarrow F_i \cup \{f'\}$;
        if $(\overline{\gamma}(F_i) > \overline{\gamma}(F_{i-1}))$ & $(|F_i| \geq u)$ then
            break;
return $F_m, CoE^m$;
```

4 Experiments

Our experimental investigation revolves around addressing three key questions:

- **Q1:** How does our proposed model compare to prior factor mining models?
- **Q2:** Which factors within the experience chain contribute to the enhancement of the RankIC&RankICIR?
- **Q3:** How does our model perform under a more realistic investment situation?

4.1 Experiment Settings

We begin by employing 38 factors from Alpha101 (Kakushadze, 2016) as our initial factor set F_0. To establish the number of clusters m, we utilize manual application of financial knowledge to classify the initial factors into distinct categories. Through this process, we ascertain that the initial factors achieve reasonable categorization when the number of categories is set to 7; hence, m is selected as 7. The number of randomly sampled factors l is designated as 2, and the minimum factor number u is defined as 15. We selected text-davinci-002[1] as the LLM for factor mining, configuring the following parameters: temperature=0 and max_tokens=1500. We retained the default

[1] https://platform.openai.com/docs/model-index-for-researchers

Category	Model	Interpretability	Training data usage	RankIC	RankICIR
Symbolic	Alpha101	✓	-	0.018(0.000)	0.200(0.000)
	GP	✓	100%	0.017(0.005)	0.141(0.034)
	LLM	✓	-	0.015(0.008)	0.139(0.011)
Neural	DTransformer	✗	100%	0.025(0.005)	0.124(0.015)
	ALSTM	✗	100%	0.028(0.006)	0.167(0.021)
	FactorVAE	✗	100%	0.048(0.008)	0.379(0.042)
Neural Symbolic	FAMA(C)	✓	10%	0.023(0.006)	0.204(0.019)
	FAMA(I-1)	✓	10%	0.016(0.006)	0.149(0.017)
	FAMA(CI-3)	✓	10%	0.030(0.008)	0.372(0.031)
	FAMA(CI-7)	✓	10%	**0.054(0.010)**	**0.485(0.051)**

Table 1: The performance of the compared models in returns prediction on the test dataset. Higher values for RankIC and RanksICIR indicate superior performance. *Interpretability* indicates that the mined factors are financially interpretable. *LLM* is the result of directly mining factors using LLMs. The term *FAMA(C)* corresponds to the CSS model. Additionally, *FAMA(I-n)* signifies the application of the CoE iteration n. The **bold** part highlights the best-performing model in the evaluation. The mean and standard deviation of results from 10 experiments are reported. The results of {*LLM, FAMA(C), FAMA(I-1), FAMA(CI-3), FAMA(CI-7)*} elucidate the ablation result of CSS and CoE methods.

parameters specified in the OpenAI API documentation[2] for any additional settings. The full factors and prompt examples are listed in Appendix B and Appendix D.

4.2 Datasets

Given that these factors are specifically crafted for the U.S. stock market, we opt for the corresponding U.S. stock index, namely the S&P500 as the stock set. Our dataset comprises all stocks from the S&P500 index, with a focus on key fields including closing price, opening price, low price, high price and volume. The temporal scope of the stock data spans from 2015/01/01 to 2022/01/01. The dataset is divided into a training set (2015/01/01-2020/01/01), a validation set (2020/01/01-2021/01/01) and a test set (2021/01/01-2022/01/01). In our model, we only use stock data for the time period 2020/06/01-2021/01/01 as the training and validation set, which is 10% amount of the provided training set.

4.3 Baselines

We explored SOTA models in recent years for comparison, encompassing both symbolic factor models and neural factor models as follows:

- **Alpha101** (Kakushadze, 2016) publicly disclosed by WorldQuant LLC [3], accompanied by precise code-based definitions. It serves as our initial set of factors from which our factors are derived.

- **GP** (Chen et al., 2021) Genetic programming algorithms create new factors through the mutation of factor expression trees, a widely cited model in factor mining.

- **ALSTM** (Qin et al., 2017) proposes a framework based on attentional mechanisms and long and short-term memory to predict stock trends.

- **DTransformer** (Wang et al., 2022) forecasts market indices by leveraging fundamental rules characterizing stock market dynamics through an encoder-decoder architecture and a full attention mechanism.

- **FactorVAE** (Duan et al., 2022) generates a prior risk factor return rate within the Variational Autoencoder (VAE) framework. It refines the prior factor return rate to approximate the posterior factor return rate.

4.4 Cross-Sectional Returns Prediction

In this experiment, we employ both the neural and symbolic factor models to forecast future stock returns for answering **Q1**. The RankIC is calculated between the forecasted and actual stock returns, as

[2] https://platform.openai.com/docs/api-reference/completions/create

[3] https://www.worldquant.com/

defined in Equation 2. For symbolic models, factors with RankIC greater than 0.01 in 2020/06/01-2021/01/01 were selected to form a test factor set. To better illustrate the relationship between prediction effectiveness and risk, we introduce the RankICIR, defined as the ratio of the mean value of the RankIC to the standard deviation:

$$RankICIR = \mathbb{E}_f[\mathbb{E}_j[\frac{\gamma(f)}{\sigma_{RankIC_t(f,r_j)}}]]. \quad (16)$$

As evidenced in Table 1, FAMA demonstrates superior performance compared to the most recent benchmark, FactorVAE. FAMA exhibits improvements of 0.006 on RankIC and 0.106 on RankICIR.

In addition, it can be observed from Table 1, that both CSS and CoE exhibit improvement in factor mining effects. Achieving satisfactory prediction results using CSS or CoE individually faces challenges. When CSS and CoE are employed together, the predictive performance of the model improves with an increasing number of mining iterations.

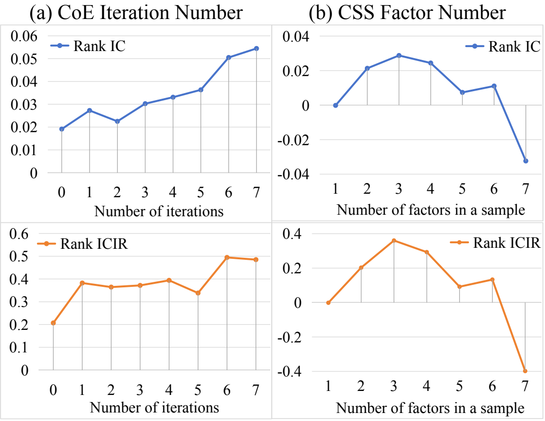

Figure 3: The results of parameter effects. Subfigure (a) illustrates RankIC and RankICIR in relation to the number of CoE iterations. Meanwhile, Subfigure (b) portrays the plot of RankIC and RankICIR with respect to the number of CSS samples.

To explore the impact of the number of CoE iterations on the model, we set the CoE iterations from 1 to 7 and verify the effect of the corresponding iterations. Results in Figure 3(a) show that the model's prediction effectiveness gradually improves with an increase in CoE iterations. The improvement effect of CoE largely depends on the generation effect of the previous round of factors.

To explore the impact of sample number selection on the model, we changed the number of cross-sample selections and conducted experiments. As shown in Figure 3(b), until the number of samples is 3, increasing the number of samples improves the performance of the model. When the quantity of samples surpasses a threshold of three, the efficacy of the model shows a decrement. This observation signifies that an excessive abundance of samples fails to enhance the performance.

4.5 Randomized Modification of Chain-of-Experience

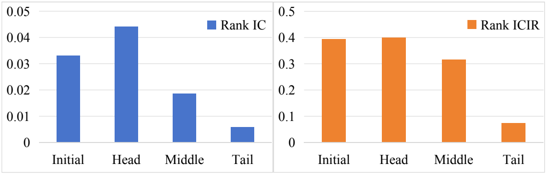

Figure 4: Impact of randomly deleting CoE nodes at different locations on model prediction. *Initial* is the performance of factors generated by retaining the complete experience chain of factors. *Head*, *Middle*, *Tail* are the performance of factors generated after randomly deleting the factors located at the head, middle, and tail of the experience chain.

In the pursuit of unraveling the fundamental components of the CoE function, we conducted an experiment that entailed the random deletion of nodes within the CoE. The objective of this endeavor is to address the inquiry encapsulated in **Q2**. Nodes are categorized into head nodes, middle nodes, and tail nodes. Given that intermediate nodes may consist of multiple nodes, we randomly select one among them as the middle node. In each round of CoEs, we systematically delete the head node, middle node, and tail node, utilizing the modified CoEs for factor mining. The results, averaged over multiple rounds, are depicted in Figure 4. We observed that the removal of initial nodes enhances the performance of factor mining. This observation suggests that the inclusion of an excessive number of low-performing nodes compromises the efficacy of factor mining in the LLM. Thus, it becomes imperative to adjust the length of the chain over time for optimal results.

4.6 Portfolio Investment Simulation

We intend to answer **Q3** by designing an investment simulation of the stock market. For symbolic models, we use the test factors in Section 4.4 and allocate funds for each factor f_i as follows:

$$w_i = \frac{RankIC_i^{past}}{\sum_{i=0}^{n} RankIC_i^{past}}, \quad (17)$$

where $RankIC_i^{past}$ represents the mean RankIC value during 2020/06/01-2021/01/01. We choose stocks with the top 20% factor value to buy and sell them in next day.

We evaluate the portfolio investment performance using standard metrics, including Annualized Return (AR), Volatility (Vol), and Sharpe Ratios (SR):

$$AR = (1 + R)^{252/N} - 1, \tag{18}$$

$$Vol = \sigma_p * \sqrt{252}, \tag{19}$$

$$SR = \frac{(R_p - R_f)}{\sigma_p} * \sqrt{252}, \tag{20}$$

where R represents the cumulative return rate, N is the total number of trading days, σ_p is the daily standard deviation of the portfolio, R_p is the expected daily return rate of the portfolio, R_f is the risk-free rate [4].

Models	AR(↑)	Vol(↓)	SR(↑)
S&P500	26.3%	11.5%	209.3%
GP	11.2%	6.8%	159.2%
Alpha101	19.7%	**4.5%**	406.1%
ALSTM	25.4%	22.5%	89.3%
DTransformer	27.8%	24.7%	93.5%
FactorVAE	31.8%	22.8%	132.2%
FAMA	**38.4%**	4.9%	**667.2%**

Table 2: Portfolio investment performance of the compared models on the test datasets. ↑ indicates a larger value is better, ↓ indicates a smaller value is better. The *S&P500* represents a portfolio comprising all S&P500 stocks.

As depicted in Table 2, the symbol-based approach exhibits lower volatility but yields comparatively lower returns. Conversely, neuro-based approaches show higher returns, albeit accompanied by elevated volatility. It is noteworthy that our approach adeptly strikes a balance between returns and volatility, demonstrating a consistent performance throughout the investment simulation without experiencing significant fluctuations. This delicate equilibrium is achieved while concurrently realizing a commendable return, highlighting the robustness and stability inherent in our model. FAMA surpasses current SOTA models, in the context of portfolio investment simulation. Specifically, there is a notable increase of 6.6% in AR and a substantial improvement of 261.1% in the SR.

5 Related Work

5.1 Financial Factor Mining

The initial phase of factor mining involves the manual mining of factors. The Capital Asset Pricing Model (CAPM) (Sharpe, 1964), posits that the expected return of a financial asset primarily depends on the market's excess return. This contributed a groundbreaking factor to the financial field. To refine this conceptual framework, the Fama-French 3-factor model (Fama and French, 1993) extends the CAPM by introducing size and value risk factors alongside market risk factors. However, manual factor mining is considered labor-intensive. To address this limitation and efficiently mine effective factors in the market, various symbolic factor-based models have been proposed. AutoAlpha (Zhang et al., 2020) expedites the identification of promising factor search spaces through the utilization of genetic algorithms. Furthermore, AlphaEvolve (Cui et al., 2021) has developed a factor mining framework grounded in AutoML, facilitating the evolution of initial factors into new factors characterized by excess returns and correlations. Factors derived through symbolic factor models exhibit clear factor calculation steps, making them easily interpretable. However, constrained by the vast symbolic factor target space, these models are generally challenging to optimize. This has prompted increased interest in the easy-to-optimize neural factor models. In a recent study, AE (Gu et al., 2021) introduces a novel latent factor conditional asset pricing model employing an autoencoder. Additionally, FactorVAE (Duan et al., 2022) integrates a dynamic factor model with a variational autoencoder to approximate the optimal factor model. The neural factor model, a method for extracting numerical characteristic factors from stock data through feature extraction, is known for its heightened optimization efficiency (Md et al., 2023; Lai et al., 2023; Wei et al., 2023). Despite this advantage, factors constrained by implicit features present challenges in terms of artificial identification, resulting in a lack of interpretability in neural factor models. In response to this, our proposed model takes a strategic approach by combining symbolic factors and leveraging neural factors for feature extraction, achieving both financial interpretability and high efficiency in the realm of factor mining. This neural-symbolic model aims to strike a balance, achieving both financial interpretability and high efficiency in the realm of factor

[4] For simplicity, we set the risk-free rate to zero.

mining.

5.2 Financial Large Language Model

LLMs find extensive application in the financial sector, spanning portfolio management, financial risk modeling, financial text mining, and financial advisory (Li et al., 2023). They hold promise in elevating the intelligence level of accounting practices and driving advancements in financial management intelligence (Minggao et al., 2023). Two prevalent methodologies exist for integrating LLMs into finance. The first method involves enhancing the financial text comprehension capabilities of LLMs through fine-tuning with specialized financial corpora (Yang et al., 2023a; Wu et al., 2023). The second approach employs prompt engineering, enabling generic LLMs to directly undertake various financial tasks such as predicting future stock trends (Lopez-Lira and Tang, 2023) and offering investment advice (Ko and Lee, 2024). The deployment of LLMs in the financial sector encounters numerous complex challenges, including business requirements, industry barriers, data privacy concerns, accountability, ethical considerations, and the knowledge gap between financial professionals and AI specialists (Lee et al., 2024). Firstly, there is a significant difficulty in collecting and processing high-quality financial data in diverse formats. Furthermore, LLMs are prone to generating information that, while seemingly plausible, may lack accuracy due to their inherent propensity for hallucinations. In addition, financial texts typically have stringent timeliness requirements, and the difficulty of updating LLMs complicates the handling of such time-sensitive financial information.

6 Conclusion

In this paper, we consider Large Language Models (LLMs) as a neural symbolic model for financial factor mining. To facilitate LLMs to pursue our task, we proposed a model called Factor Mining Agent (FAMA), which comprises two integral components: Cross-Sample Selection (CSS) and Chain-of-Experience (CoE). CSS mitigates the homogeneity in the mined factors by amalgamating diverse guidance factors. CoE encourages In-Context Learning (ICL) to explore novel factor paradigms by leveraging the paths leading to the mining of effective factors as experiential prompts. Both CSS and CoE components are integrated into our factor mining agent to effectively mine financially interpretable factors. Experimental results demonstrate the effectiveness of our proposed approach. Our future work includes exploring more avenues to enhance the optimization of factor mining and addressing the hallucination phenomenon of LLMs.

Limitations

When employing LLMs for factor mining, we observed the hallucination phenomenon of LLMs within the financial domain that introduces interference in the factor mining process. In future endeavors, our emphasis will be directed towards mitigating the hallucination effects of LLMs in the context of factor mining.

Ethics Statement

We utilize the OpenAI API in strict adherence to the OpenAI User Rules for the generation of financial factors, ensuring the absence of harmful and unethical information. Our approach has undergone validation in historical market scenarios and expressly does not offer any form of investment advice.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 72271233, U23A20388), Suzhou Key Laboratory of Artificial Intelligence and Social Governance Technologies (SZS2023007), Smart Social Governance Technology and Innovative Application Platform (YZCXPT2023101) and Key Research and Development Program of Yunnan Province (202303AP140008).

References

Zulfiqar Ali Imran, Abdullah Ejaz, Cristi Spulbar, Ramona Birau, and Periyapatna Sathyanarayana Rao Nethravathi. 2020. Measuring the impact of governance quality on stock market performance in developed countries. *Economic Research-Ekonomska Istraživanja*, 33(1):3406–3426.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Mark M Carhart. 1997. On persistence in mutual fund performance. *The Journal of finance*, 52(1):57–82.

Tianxiang Chen, Wei Chen, and Luyao Du. 2021. An empirical study of financial factor mining based on gene expression programming. In *2021 4th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)*, pages 1113–1117. IEEE.

Can Cui, Wei Wang, Meihui Zhang, Gang Chen, Zhaojing Luo, and Beng Chin Ooi. 2021. Alphaevolve: A learning framework to discover novel alphas in quantitative investment. In *Proceedings of the 2021 International Conference on Management of Data*, pages 2208–2216.

Yitong Duan, Lei Wang, Qizhong Zhang, and Jian Li. 2022. Factorvae: A probabilistic dynamic factor model based on variational autoencoder for predicting cross-sectional stock returns. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4468–4476.

Eugene F Fama and Kenneth R French. 1993. Common risk factors in the returns on stocks and bonds. *Journal of financial economics*, 33(1):3–56.

Shihao Gu, Bryan Kelly, and Dacheng Xiu. 2021. Autoencoder asset pricing models. *Journal of Econometrics*, 222(1):429–450.

Yue Guo, Zian Xu, and Yi Yang. 2023. Is chatgpt a financial expert? evaluating language models on financial natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 815–821.

Yufeng Han, Guofu Zhou, and Yingzi Zhu. 2016. A trend factor: Any economic gains from using information over investment horizons? *Journal of Financial Economics*, 122(2):352–375.

Kevin B Hendricks and Vinod R Singhal. 2009. Demand-supply mismatches and stock market reaction: Evidence from excess inventory announcements. *Manufacturing & Service Operations Management*, 11(3):509–524.

Kewei Hou, G Andrew Karolyi, and Bong-Chan Kho. 2011. What factors drive global stock returns? *The Review of Financial Studies*, 24(8):2527–2574.

Ying Jin, Weilin Fu, Jian Kang, Jiadong Guo, and Jian Guo. 2019. Bayesian symbolic regression. *arXiv preprint arXiv:1910.08892*.

Zura Kakushadze. 2016. 101 formulaic alphas. *Wilmott*, 2016(84):72–81.

Bryan T Kelly, Seth Pruitt, and Yinan Su. 2019. Characteristics are covariances: A unified model of risk and return. *Journal of Financial Economics*, 134(3):501–524.

Hyungjin Ko and Jaewook Lee. 2024. Can chatgpt improve investment decisions? from a portfolio management perspective. *Finance Research Letters*, page 105433.

K Krishna and M Narasimha Murty. 1999. Genetic k-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(3):433–439.

Tzu-Ya Lai, Wen Jung Cheng, and Jun-En Ding. 2023. Sequential graph attention learning for predicting dynamic stock trends (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 16244–16245.

Jean Lee, Nicholas Stevens, Soyeon Caren Han, and Minseok Song. 2024. A survey of large language models in finance (finllms). *arXiv preprint arXiv:2402.02315*.

Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. 2023. Large language models in finance: A survey. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, pages 374–382.

Alejandro Lopez-Lira and Yuehua Tang. 2023. Can chatgpt forecast stock price movements? return predictability and large language models. *Return Predictability and Large Language Models (April 6, 2023)*.

Nour Makke and Sanjay Chawla. 2024. Interpretable scientific discovery with symbolic regression: a review. *Artificial Intelligence Review*, 57(1):2.

Abdul Quadir Md, Sanjit Kapoor, Chris Junni AV, Arun Kumar Sivaraman, Kong Fah Tee, H Sabireen, and N Janakiraman. 2023. Novel optimization approach for stock price forecasting using multi-layered sequential lstm. *Applied Soft Computing*, 134:109830.

Li Minggao, Yi Fengchao, Li Yagang, et al. 2023. Research on the application of llm in power finance middle platform. In *2023 2nd Asian Conference on Frontiers of Power and Energy (ACFPE)*, pages 282–289. IEEE.

Victor Ng, Robert F Engle, and Michael Rothschild. 1992. A multi-dynamic-factor model for stock returns. *Journal of Econometrics*, 52(1-2):245–266.

Yao Qin, Dongjin Song, Haifeng Cheng, Wei Cheng, Guofei Jiang, and Garrison W Cottrell. 2017. A dual-stage attention-based recurrent neural network for time series prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2627–2633.

Stephen A Ross. 2013. The arbitrage theory of capital asset pricing. In *Handbook of the fundamentals of financial decision making: Part I*, pages 11–30. World Scientific.

William F Sharpe. 1964. Capital asset prices: A theory of market equilibrium under conditions of risk. *The journal of finance*, 19(3):425–442.

Rahul Verma and GökÇe Soydemir. 2009. The impact of individual and institutional investor sentiment on the market price of risk. *The Quarterly Review of Economics and Finance*, 49(3):1129–1145.

Chaojie Wang, Yuanyuan Chen, Shuqi Zhang, and Qiuhui Zhang. 2022. Stock market index prediction using deep transformer model. *Expert Systems with Applications*, 208:118128.

Zikai Wei, Anyi Rao, Bo Dai, and Dahua Lin. 2023. 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*, pages 4903–4911.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023a. Fingpt: Open-source financial large language models. *FinLLM Symposium at IJCAI 2023*.

Yi Yang, Yixuan Tang, and Kar Yan Tam. 2023b. Investlm: A large language model for investment using financial domain instruction tuning. *arXiv preprint arXiv:2309.13064*.

Tianping Zhang, Yuanqi Li, Yifei Jin, and Jian Li. 2020. Autoalpha: An efficient hierarchical evolutionary algorithm for mining alpha factors in quantitative investment. *arXiv preprint arXiv:2002.08245*.

A Algorithm

Function	Description
Cluster	generate factor clustering sets
SelectSample	select factors as in-context samples
MatchCoE	match the relevant CoE
GenChain	generate a new CoE

Table 3: Algorithm function description.

B Factor

We select the following factors (alphas) from Alpha101 (Kakushadze, 2016) as the initial factor set: “002, 003, 004, 005, 006, 008, 011, 012, 013, 014, 015, 016, 017, 018, 019, 020, 022, 025, 026, 028, 029, 030, 031, 032, 033, 034, 035, 036, 037, 038, 039, 040, 041, 042, 043, 044, 045, 047, 050, 052, 053, 054, 055, 057, 060, 061, 062, 064, 065, 066, 068, 071, 072, 073, 074, 075, 077, 078, 081, 083, 084, 085, 086, 088, 092, 094, 095, 096, 098, 099, 101”.

C Additional Experiment

Model	RankIC	RankICIR
gpt-3.5-turbo	0.054	0.481
text-davinci-003	0.056	0.492

Table 4: The performance of FAMA using other LLMs in returns prediction.

D Prompt

“function_definition” is from “Functions and Operators” in Alpha101 (Kakushadze, 2016).

Instruction

```
You are an alpha generator. You should follow the following rules:
1. The inputs are the alpha factors that are currently performing well, and you are
     required to output a new alpha factor that is generated from the fusion of
    these factors, and your factor must be different from the input factor.
2. Do not repeat example answer.
3. You should return new different factors in a json array.
4. The specific function is defined as follows:
{function_definition}
5. Follow the path in "improve_path". -> Indicates that the following factors have
    better performance than the previous factors. You should refer it to build new
    alpha.
```

Input Example

```
alphas: ["(-1 * correlation(open, volume, 10))"]
generate_factor_num: 1
improve_path: "close/open" -> "rank(close)/rank(open)"
```

Output Example

```
["rank(correlation(open, volume, 10) / rank(open))"]
```