Beyond Regression: Binary Encoding Classification with Confidence for Stock Index Prediction

Junzhe Jiang[1,*], Chang Yang[1,*], Xinrun Wang[2,†], Bo Li[1]

[1]The Hong Kong Polytechnic University, Hong Kong
[2]Singapore Management University, Singapore

{junzhe.jiang, chang.yang, comp-bo.li}@connect.polyu.hk, xrwang@smu.edu.sg

***Abstract*—Stock market indices serve as fundamental market measurements that quantify systematic market dynamics. However, accurate index price prediction remains challenging, primarily because existing approaches treat indices as isolated time series and frame the prediction as a simple regression task. These methods fail to capture indices' inherent nature as aggregations of constituent stocks with complex, time-varying interdependencies. To address these limitations, we propose CUBIC, a novel end-to-end framework that explicitly models the adaptive fusion of constituent stocks for index price prediction. Our main contributions are threefold. i) Fusion in the latent space: we introduce the fusion mechanism over the latent embedding of the stocks to extract information from a large number of constituents. ii) Binary encoding classification: since regression tasks are challenging due to continuous value estimation, we reformulate the regression into a classification task, where the target value is converted to binary and we optimize the prediction of the value of each digit with cross-entropy loss. iii) Confidence-guided prediction and trading: we introduce the regularization loss to address market prediction uncertainty for the index prediction and design the rule-based trading policies based on the confidence. Extensive experiments across multiple stock markets demonstrate that CUBIC achieves superior performance on both forecasting accuracy and trading profitability compared to baseline approaches, with performance gains varying across different market conditions and model architectures.**

***Index Terms*—stock index prediction, binary encoding.**

I. INTRODUCTION

The prediction of stock market indices has long been an intriguing topic in financial research, driven by the potential for stable profits through index option trading [1]. Stock market indices, such as the Dow Jones Industrial Average (DJIA), measure the value of the market through selected representative stocks [2]. Accurate index predictions are crucial for modern financial markets, underlying portfolio management, and various financial products that collectively represent trillions in trading volume [3].

In recent years, deep learning models have gained significant attention in stock price prediction. Various architectures such as transformers [4], multilayer perceptrons (MLP) [5], and long-short-term memory networks (LSTMs) [6] have been extensively explored for capturing complex patterns and temporal dependencies in financial time series. Recent advances include attribute-driven graph attention networks [7] that model momentum spillover effects among stocks, the market-guided stock transformer (MASTER) [8] that incorporates market information into attention mechanisms, and StockMixer [9] that captures nonlinear relationships through MLP-based architecture. Deep Transformer architectures [10] and attention-enhanced GRU models [11] have shown effectiveness in handling long-sequence dependencies, while hybrid approaches such as CEEMDAN-LSTM [2] and LSTM studies [12], [13] have validated capabilities in financial forecasting.

Despite these advances, existing methods face two major limitations [14]. First, stock indices present unique modeling challenges compared to individual stocks [15]. Index prediction requires capturing complex cross-sectional dependencies among constituents with time-varying correlations, while traditional approaches treat indices as univariate time series, failing to leverage constituent relationships. Moreover, indices exhibit non-stationary statistical properties due to periodic rebalancing and constituent changes, creating additional modeling complexity. Second, formulating price prediction as regression faces significant challenges due to market volatility [16]. Regression models assume continuous, smooth movements but struggle with sharp fluctuations and sudden market events, leading to suboptimal performance in volatile financial markets [17]–[19], especially during abrupt jumps and regime shifts.

To address the above challenges, we propose a novel framework, i.e., **C**omponent f**U**sion and **B**inary encoding class**I**fication with **C**onfidence (CUBIC), for the prediction of stock index. Our framework serves as a general-purpose enhancement that can work with various deep learning architectures to improve prediction and trading performance. The four contributions of CUBIC are:

- We present a novel formulation that transforms stock index prediction from regression to binary encoding classification by binarizing target values, enabling robust predictions while quantifying uncertainty.
- We employ latent space fusion to alleviate the rigidity of predefined index aggregation and capture nonlinear cross-stock interactions.

*These authors contributed equally.
†Corresponding author

- We introduce a confidence-guided framework that leverages classification probability outputs as confidence measures and incorporates regularization loss to enhance reliability and guide trading decisions.
- CUBIC integrates stock component fusion with binary encoding classification, providing a general enhancement framework for various deep learning architectures. Experiments on three market indices demonstrate improved prediction and trading performance across different market conditions. The code and dataset is released at https://github.com/JiangInsight/CUBIC/tree/master.

Related Work. Stock market index prediction has evolved from statistical approaches like ARMA [20], [21] to sophisticated deep learning architectures. Recent advances include Transformers [4] for capturing long-range dependencies [22], [23], and inverted Transformers [24] that treat multiple stocks as channels in a unified latent embedding space. LSTM models for temporal patterns [25], [26], and hybrid approaches combining traditional and deep learning methods [27]. Regression-to-classification reformulation has gained attention in machine learning [28] and financial applications [29]. While general approaches exist for binarized regression [30] and portfolio optimization [31], [32], our work uniquely integrates constituent-level fusion with binary encoding for index prediction, providing uncertainty quantification and trading strategy derivation within a unified framework.

II. Problem Statement

Let $[N]$ denote the set of constituent stocks in the target market index. For each stock $i \in [N]$, we extract M technical indicators at each time step $t \in [1, \tau]$, generating feature vectors $x_{i,t} \in \mathbb{R}^M$. Based on established research [33], we incorporate 16 technical indicators classified into three groups: Trend indicators for directional movements, oscillator indicators for momentum and reversal signals, and volatility indicators for risk levels, which collectively capture essential market dynamics. The concise overview is in Table I. In line with [9], we define our prediction target as the market return: $y_t = \frac{I_{t+1} - I_t}{I_t}$, where I_t represents the value of the market index at time t. To ensure numerical stability and model convergence, we apply standardization to obtain $\hat{y}_t = \text{standardize}(y_t)$. To apply binary encoding, we map the standardized returns $\hat{y}_t$ to the range $[-1, 1]$. Since standardized returns typically fall within $[-3, 3]$, we apply robust normalization: $v = \text{clip}(\hat{y}_t/3, -1, 1)$ where $\text{clip}(\cdot, -1, 1)$ ensures values remain within the encoding range. The market index prediction task is formulated as: given the technical indicators $\{x_{i,s}\}_{i\in[N], s\in[t-\tau+1,t]}$ of the constituent stocks over τ consecutivetime steps, predict the normalized next-day market return $\hat{y}_t$. We do not directly regress the index level I_{t+1}; instead, we predict the next-day return and convert it to actionable trading decisions.

III. CUBIC

This section introduces the CUBIC framework, as shown in Figure 1. Specifically, CUBIC first embeds the stock indicators into the latent space and performs fusion using pooling techniques. Then, we introduce binary encoding classification for stock index price prediction, where the continuous target value is discretized and transformed into a binary representation. Finally, CUBIC leverages confidence to construct the regularization loss for prediction and trading.

TABLE I
Summary of the technical indicators used.

Type	Indicators
Trend	Arithmetic Ratio, Open, Close, Close SMA, Volume SMA, Close EMA, Volume EMA, ADX
Oscillator	RSI, MACD, MACD Signal, K, MFI
Volatility	ATR, BB Middle, OBV

A. Fusion in Latent Space

A market index typically comprises multiple constituent stocks; for example, CSI 100 consists of 100 stocks, each with 16 technical indicators across τ time steps, resulting in input dimensionality of $100 \times 16 \times \tau$. This high-dimensional feature space poses significant challenges for deep learning models to extract meaningful patterns, as the curse of dimensionality makes distinguishing signal from noise difficult. Consequently, models are prone to overfitting, capturing spurious correlations rather than market dynamics driving collective stock behaviors and index movements.

a) Feature Embedding of Stocks: Given various technical indicators, we introduce an embedding model to project stock indicators into a low-dimensional latent space of 32 dimensions. This embedding mechanism addresses the curse of dimensionality problem inherent in high-dimensional financial data, where numerous technical indicators may contain redundant or noisy information that can degrade model performance. Specifically, we employ a shared Multi-Layer Perceptron (MLP) across all stocks to obtain embeddings:

$$\boldsymbol{e}^i = \text{EMB}(\boldsymbol{x}^i), i \in [N], \tag{1}$$

where $\boldsymbol{x}^i$ denotes the input features of stock i, and $\boldsymbol{e}^i$ represents its learned embedding. The embedding model $\text{EMB}(\cdot)$ uses shared parameters across all stocks to ensure scalability and parameter efficiency. This shared architecture enables the model to learn common patterns across different stocks while maintaining computational tractability as the number of constituents grows. The embedding model $\text{EMB}(\cdot)$ is implemented as an MLP to capture non-linear relationships and extract the most predictive information.

b) Pooling: Given constituent stock embeddings, a straightforward approach would concatenate all embeddings into a flat vector. However, this approach has limitations: it scales linearly with constituent count, causing parameter explosion, while treating all stocks equally and failing to capture financial markets' asymmetric nature where extreme movements carry disproportionate predictive power. Additionally, concatenation cannot distinguish between market mo-

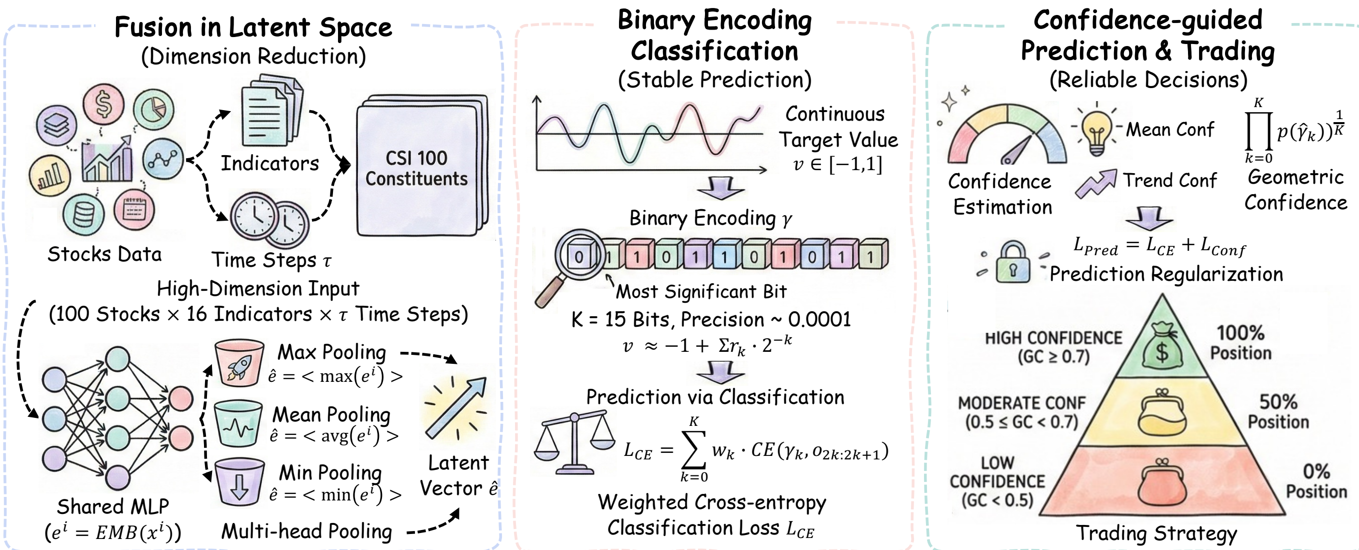

Fig. 1. Overview of the CUBIC framework: latent fusion, binary encoding, and confidence-guided trading.

mentum, reflecting consensus, or signaling contrarian risks, losing distributional information for accurate prediction. To address these limitations, we introduce a multi-head pooling mechanism combining three complementary operations:

- Max pooling selects the maximum value across stocks: $\hat{\boldsymbol{e}} = \langle \hat{e}_j \rangle$, where $\hat{e}_j = \max_{i \in [N]}\{e_j^i\}$, capturing extreme market movements.
- Mean pooling computes the average: $\hat{\boldsymbol{e}} = \langle \hat{e}_j \rangle$, where $\hat{e}_j = \sum_{i \in [N]}\{e_j^i\}/N$, preserving overall market trends.
- Min pooling captures the minimum value: $\hat{\boldsymbol{e}} = \langle \hat{e}_j \rangle$, where $\hat{e}_j = \min_{i \in [N]}\{e_j^i\}$, identifying downturns.

The combination of these three pooling operations ($\hat{\boldsymbol{e}} = \langle \hat{e}\mathrm{max}, \hat{e}\mathrm{mean}, \hat{e}_{\min} \rangle$) provides a comprehensive view of the market by capturing diverse aspects of stock behavior: peak signals through maximum pooling, average trends through mean pooling, and bottom signals through min pooling. This multihead design addresses the challenge of dimension while allowing the model to learn from multiple market perspectives in a computationally efficient manner. Our pooling strategy is motivated by financial market theory. Max pooling captures market leaders and momentum stocks that drive index movements, aligning with the concept that a few dominant stocks often determine market direction. Mean pooling reflects the market consensus and average behavior, consistent with efficient market hypothesis assumptions. Min pooling identifies lagging stocks and potential risks, crucial for downside protection. This combination provides a comprehensive market view that no single aggregation method can achieve, while maintaining computational efficiency with $O(N)$ complexity regardless of constituent count.

B. Binary Encoding Classification

Price prediction through regression poses significant challenges in financial markets due to high volatility, non-stationarity, and complex temporal dependencies that regression models struggle to capture. The continuous nature of regression targets complicates learning robust decision boundaries, where small prediction errors can cause substantial trading losses. Regression models are particularly vulnerable to financial markets' low signal-to-noise ratio, leading to unstable predictions. Therefore, we reformulate price prediction by encoding continuous values into binary representations. CUBIC converts each price into fixed-point binary format, treats each bit as a classification target, then reconstructs prices from predicted binary digits. This strategy transforms challenging regression into multiple binary classification tasks, eliminating gradient vanishing issues while ensuring numerical precision through bit-level reconstruction. Each bit prediction provides stable cross-entropy learning signals while preserving price prediction capability.

a) Binary Encoding.: For the prediction target value $v \in [-1, 1]$, we represent it using binary encoding format:

$$v = -1 + \sum\nolimits_{k=0}^{K-1} \gamma_k \cdot 2^{-k}, \gamma_k \in \{0, 1\} \qquad (2)$$

where $K = 15$ is determined by the required precision: to achieve 0.0001 precision over the range $[-1, 1]$, we need at least $\lceil \log_2(2/0.0001) \rceil = 15$ bits. The maximum representable value is $-1+\sum_{k=0}^{14} 2^{-k} = -1+(1-2^{-15}) \approx 0.999969$, which closely approximates the target range $[-1, 1]$. This representation ensures that price movements as small as $2^{-15} \approx 3\times10^{-5}$ can be distinguished, exceeding typical minimum tick sizes in financial markets. This precision requirement is motivated by the minimum tick size in stock price movements, which is typically 0.0001, ensuring our binary representation can capture the finest granularity of actual price changes without losing meaningful information. Here γ_k represents each bit in the binary representation.

b) Prediction via Classification: The model outputs $2K$ dimensions, with each pair of dimensions predicting one binary digit. We propose a weighted cross-entropy loss to emphasize the hierarchical nature of binary encoding. Given the binary encoding $\boldsymbol{\gamma}$ and model output $\boldsymbol{o}$, the loss is:

$$L_{CE}(\boldsymbol{\gamma}, \boldsymbol{o}) = \sum_{k=0}^{K-1} w_k \texttt{CE}(\gamma_k, \boldsymbol{o}_{2k:2k+1}) \tag{3}$$

where $\texttt{CE}(\cdot)$ is the cross-entropy loss and w_k denotes the position-dependent weight. The position-dependent weights w_k are defined to emphasize more significant bits: $w_k = \frac{2^{K-1-k}}{2^K-1}$ This exponentially decreasing weighting scheme ensures that errors in higher-order bits are penalized more heavily than in lower-order bits. The binary decomposition induces a multiresolution representation of target values, where each bit position k corresponds to different magnitudes: higher-order bits capture coarse-grained patterns while lower-order bits refine predictions. Such multiscale learning, combined with stable gradients from cross-entropy loss and position-dependent weighting, enables more effective learning from limited financial data compared to direct regression. Binary encoding transforms unstable continuous regression into stable binary classifications with bounded cross-entropy gradients. The 15-bit precision exceeds typical market tick sizes, ensuring sufficient granularity for financial applications.

c) Binary Encoding Superiority: Binary encoding provides theoretical advantages over regression in financial markets through three mechanisms: (1) **Gradient Stability**: Cross-entropy loss produces bounded gradients $\nabla L \in [-1, 1]$, while MSE gradients can explode during market volatility, leading to training instability. (2) **Decision Boundary Clarity**: Each bit classification creates well-defined decision boundaries, whereas regression struggles with the continuous nature of financial returns that exhibit heavy tails and non-Gaussian distributions. (3) **Hierarchical Learning**: The multi-scale bit representation aligns with financial market dynamics where coarse trends (captured by higher-order bits) are more predictable than fine-grained price movements (lower-order bits).

C. Confidence-guided Prediction and Trading

Classification tasks naturally provide confidence estimates through class probability disparities. Our binary encoding module enables this by transforming regression into multiple binary classifications, where each bit prediction comes with an inherent confidence measure. When the predicted probabilities of two classes exhibit large disparity , the model demonstrates high confidence in its prediction; conversely, when probabilities are similar, the model is less certain. For each binary digit k, given model output probabilities for both classes, we determine the predicted bit value as:

$$\hat{\gamma}_k = \arg\max_{c \in \{0,1\}} \{p(c)\} \tag{4}$$

where $p(c)$ represents the model's predicted probability for class c. We leverage this confidence information to design regularization mechanisms that guide both predictions and trading decisions. Specifically, we propose two variants of geometric confidence (GC) to capture different aspects of prediction reliability across the binary representation.

a) Mean Confidence: The geometric confidence (GC) measure is defined as: $GC_{mean} = \exp\left(\frac{1}{K}\sum_{k=0}^{K-1} \log p(\hat{\gamma}_k)\right)$, where $p(\hat{\gamma}_k)$ represents the model's predicted probability for the chosen bit value $\hat{\gamma}_k$ at position k. This logarithmic formulation prevents numerical underflow that can occur when multiplying many probabilities, while maintaining the geometric mean interpretation. The confidence measure ranges from 0 to 1, where values closer to 1 indicate higher prediction certainty. The geometric mean of probabilities across all binary digits evaluates the model's comprehensive understanding of price patterns across all scales.

b) Trend Confidence: To capture confidence in directional movement prediction, we define: $GC_{trend} = p(\hat{\gamma}_0)$, which examines confidence of the most significant bit. This measure reflects the model's certainty in predicting price trend direction, crucial for directional trading decisions.

These complementary confidence measures enable more nuanced trading strategies by considering both the overall prediction reliability and the directional movement confidence. A high GC_{mean} indicates consistent confidence across all price scales, while a high GC_{trend} suggests strong conviction in the predicted price trend direction.

c) Confidence-guided Prediction: To enhance the model's predictive capability, we leverage geometric confidence to guide prediction behavior through a confidence-aware regularization mechanism. Specifically, when the model correctly identifies the directional indicator corresponding to the sign of the predicted trajectory, our goal is to maximize confidence in the predicted value. Conversely, when the model's predicted trend is incorrect, we tend to minimize the confidence in the predicted value. The confidence-guided regularization loss is designed to encourage high confidence only when predictions are correct. Using the ground-truth most significant bit γ_0, we define:

$$L_{\text{Conf}} = (1 - 2 \cdot \mathbb{I}[\hat{\gamma}_0 = \gamma_0]) \cdot GC_{mean} \tag{5}$$

where $\mathbb{I}[\cdot]$ is the indicator function. This formulation encourages the model to express high confidence (large GC_{mean}) when the predicted trend direction $\hat{\gamma}_0$ matches the ground truth γ_0, and low confidence otherwise. The total prediction loss combines both components with equal weighting:

$$L_{pred} = L_{CE} + L_{Conf} \tag{6}$$

.

d) Confidence-guided Trading.: Signals are computed after day t close and executed on day $t+1$. Position sizing follows a three-tier structure based on geometric confidence: no position (0% allocation) for low confidence ($GC < 0.5$), conservative position (50% allocation) for moderate confidence ($0.5 \leq GC < 0.7$), and full position (100% allocation) for high confidence ($GC \geq 0.7$).

These thresholds are empirically determined based on the distribution of confidence scores in our validation set, representing the 33rd and 67th percentiles. The tiered approach balances risk management with return optimization, allowing the model to scale exposure based on prediction certainty.

IV. Experiments

To evaluate CUBIC's effectiveness, we conduct experiments on three major indices as representative market benchmarks. This validates that CUBIC consistently improves prediction performance across various market conditions, baseline architectures, and financial environments.

TABLE II
STATISTICS OF DATASETS.

	DJIA	HSI	CSI 100
# Stocks	30	80	100
Start Time	12-11-08	12-11-08	12-11-08
End Time	24-11-07	24-11-08	24-11-07
Train Days	2114	2065	2037
Val Days	604	590	582
Test Days	302	295	291

A. Experiment Setup

a) Datasets: We evaluated our model on three major market indices: the Dow Jones Industrial Average (DJIA), the Hang Seng Index (HSI), and the CSI 100, representing US, Hong Kong, and China markets respectively. We use the public end-of-day trading dataset from Yahoo Finance*, with statistics shown in Table II. These indices represent distinct market characteristics: DJIA reflects a mature market with high institutional participation, HSI bridges Eastern and Western trading practices, and CSI 100 tracks China's largest A-share stocks with high retail participation and policy sensitivity. As key benchmarks in their respective markets, they provide ideal testbeds for evaluating predictive capabilities across diverse market environments. All experiments used 10 random seeds.

b) Base Model Architectures: To demonstrate CUBIC's adaptability as a plug-and-play enhancement framework, we deliberately select fundamental architecture backbones representing fundamental building blocks of contemporary financial forecasting models [34]. We deliberately choose basic architectures rather than sophisticated variants to establish clear and reproducible baselines and isolate our framework's contribution. This ensures performance improvements are directly attributed to CUBIC's enhancements rather than confounded by complex architectural modifications.

- Long Short-Term Memory (LSTM) [12] - Recurrent network for temporal sequence modeling.
- Transformer [4] - Attention-based architecture for long-range dependency modeling.
- Multi-Layer Perceptron (MLP) [35] - Feedforward network for non-linear pattern learning.

*https://github.com/yahoo-finance

c) Evaluation Metrics: We adopt predictive performance and portfolio-based metrics to comprehensively evaluate CUBIC's effectiveness across different market conditions and model architectures. For predictive performance, we employ **Information Coefficient (IC)**, measuring daily Pearson correlation between predicted and actual stock returns; **Information Coefficient Information Ratio (ICIR)**, calculated as IC normalized by its standard deviation to assess prediction consistency and stability; and **Direction Accuracy (DA)**, evaluating the model's ability to correctly predict price movement directions. For portfolio-based evaluation, we implement long-short trading strategies with 0.1% transaction costs to simulate realistic trading conditions and market frictions. Baseline models use simple binary signals, taking long positions when predicted returns are positive and short positions otherwise. Models with confidence-guided trading adjust position sizes based on prediction confidence levels, detailed in Section III-C0d. We report **Sharpe Ratio (SR)** for risk-adjusted returns and **Annualized Return (AR)** for comprehensive investment profitability assessment.

B. Experiment Results

TABLE III
IMPORTANCE OF CONSTITUENT STOCKS AND FUSION. RESULTS BASED ON US STOCK MARKET. REG: REGRESSION BASELINE WITH RAW FEATURES. BN: BINARY ENCODING CLASSIFICATION. "+SINGLE": MODELS USING ONLY INDEX DATA. REG+FS: REGRESSION WITH FEATURE FUSION. BN+FS: BINARY ENCODING WITH FEATURE FUSION.

		IC	ICIR	DA	SR	AR
MLP	Reg+Single	-0.044	-0.314	0.468	0.185	0.012
	BN+Single	-0.055	-0.408	0.485	0.612	0.046
	Reg	0.018	0.130	0.492	0.584	0.056
	BN	0.024	0.249	0.525	0.855	0.089
	Reg+FS	0.014	0.133	0.504	0.682	0.075
	BN+FS	0.027	0.284	0.517	0.916	0.092
LSTM	Reg+Single	-0.025	-0.152	0.439	0.103	0.004
	BN+Single	0.007	-0.001	0.488	0.530	0.044
	Reg	0.007	0.090	0.495	0.293	0.035
	BN	0.013	0.131	0.505	0.531	0.064
	Reg+FS	0.010	0.136	0.502	0.637	0.068
	BN+FS	0.020	0.232	0.508	0.560	0.067
TF	Reg+Single	-0.023	-0.214	0.472	0.467	0.039
	BN+Single	-0.005	0.002	0.475	0.428	0.035
	Reg	0.021	0.234	0.515	0.619	0.062
	BN	0.029	0.310	0.523	0.651	0.078
	Reg+FS	0.023	0.257	0.528	0.513	0.077
	BN+FS	0.025	0.450	0.535	0.712	0.086

a) RQ1: Can CUBIC'S *latent fusion and binary encoding boost model performance?:* **Binary Encoding Enhancement:** Binary classification consistently improves performance across all models, with the magnitude of improvement varying by architecture. For MLP, **BN** increases IC from 0.018 to 0.024 and Sharpe ratio from 0.584 to 0.855, representing 33% and 46% improvements. LSTM models show the most dramatic enhancement, with **BN** improving IC from 0.007 to 0.013 and

Sharpe ratio from 0.293 to 0.531, achieving 86% and 81% improvements. Transformer demonstrates **BN** increasing IC from 0.021 to 0.029 and ICIR from 0.234 to 0.310, with 38% and 32% improvements. The nearly doubled IC improvement in LSTM models demonstrates that binary encoding benefits sequential modeling architectures by providing clearer directional signals that enhance temporal pattern recognition.

Feature Space Fusion Benefits: The **BN+FS** configuration achieves the highest performance across all architectures, with substantial additional gains over binary encoding alone. MLP reaches an exceptional 0.916 Sharpe ratio with 6.8% annualized return, improving over **BN**'s 0.855 Sharpe ratio. For LSTM, BN improves IC from 0.007 to 0.013 and SR from 0.293 to 0.531; adding fusion (BN+FS) further improves ICIR from 0.131 to 0.232. The transformer achieves the best overall performance with 0.450 ICIR, 0.712 Sharpe ratio, and annualized return of 8. 6%. The improvement in ICIR of the transformer from 0.310 under **BN** to 0.450 under **BN+FS** represents a 45% improvement, demonstrating that the aggregation of features across assets captures the dynamics of the market that individual stock analysis cannot access, providing predictive power beyond directional encoding.

TABLE IV

THE IMPORTANCE OF CONFIDENT GUIDED PREDICTION AND TRADING MECHANISM. THE RESULTS ARE BASED ON USA STOCK MARKET. BF: BASE MODEL WITH BINARY ENCODING AND FUSION. BF+MEAN: ADDS CONFIDENCE MEAN LOSS. BF+TREND: ADDS CONFIDENCE TREND LOSS. BF+MEAN+DM AND BF+TREND+DM: INCORPORATE CONFIDENCE-GUIDED TRADING SIGNALS.

		IC	ICIR	DA	SR	AR
MLP	BF	0.027	0.284	0.517	0.916	0.092
	BF+Mean	0.026	**0.367**	0.538	1.080	0.108
	BF+Trend	0.027	0.277	**0.547**	0.955	0.095
	BF+Mean+DM	0.026	0.272	0.542	0.927	0.130
	BF+Trend+DM	**0.028**	0.348	0.535	**1.324**	**0.132**
LSTM	BF	0.020	0.232	0.508	0.560	0.067
	BF+Mean	0.026	0.243	0.535	0.733	0.081
	BF+Trend	0.027	0.255	0.532	0.573	0.080
	BF+Mean+DM	0.024	0.283	0.537	0.704	0.106
	BF+Trend+DM	**0.028**	**0.344**	**0.543**	**0.807**	**0.113**
TF	BF	0.025	0.450	0.531	0.712	0.085
	BF+Mean	0.031	0.373	**0.559**	0.785	0.102
	BF+Trend	0.036	0.311	0.552	0.783	0.110
	BF+Mean+DM	0.038	0.401	0.558	**1.232**	**0.149**
	BF+Trend+DM	**0.040**	**0.471**	0.547	0.897	0.143

b) RQ2: Can CUBIC'S *confidence mechanisms improve prediction and trading metrics?:* Following our previous findings, we now examine how CUBIC'S confidence mechanisms affect model performance across five configurations. As shown in Table IV, the baseline **BF** establishes IC benchmarks of 0.020-0.027, with MLP achieving 0.517 DA and TF reaching 0.712 SR. Mean confidence integration **BF+Mean** demonstrates substantial gains: MLP's ICIR jumps from 0.284 to 0.367, representing a 29% improvement, while LSTM's SR improves from 0.560 to 0.733, indicating effective filtering of unstable predictions. Trend confidence **BF+Trend** further enhances market momentum capture, with MLP's DA increasing to 0.547 and maintaining competitive SR at 0.955. The decision-making layer integration yields the most dramatic improvements. **BF+Mean+DM** elevates TF's performance significantly, with SR climbing from 0.712 to 1.232 and AR reaching 14.9%, demonstrating optimized position sizing. Meanwhile, **BF+Trend+DM** achieves peak performance with MLP's SR of 1.324, up from baseline 0.916, and AR of 13.2%, reflecting sophisticated timing decisions. These progressive enhancements validate the synergistic effects of combining confidence mechanisms with adaptive decision-making for superior trading performance.

Building upon our confidence-guided mechanisms, we conducted an ablation study in two contrasting scenarios, the CSI 100 Index bull market 2024.06–2024.11 and the HSI bear market 2023.08–2024.02. Figure 2 shows the normalized cumulative portfolio value starting from 1 in the top row and the corresponding discrete inventory decisions Full, Half, and Zero in the bottom row. Mean confidence captures prediction reliability and yields smoother, more moderate switching, reducing exposure in the HSI bear market to limit drawdowns. Trend confidence captures directional persistence and regime shifts, de-risking earlier in the HSI bear market and ramping up to Full more decisively in the CSI 100 bull market, leading to stronger upside participation.

c) RQ3: Does CUBIC *demonstrate consistent improvements across markets and models?:* To validate the robustness of the CUBIC strategy, we explore whether it demonstrates consistent improvements across different markets and models, showing progressive performance gains as components are added to the base model.

As shown in Table V, the CUBIC framework demonstrates remarkable performance through four synergistically integrated components that work cohesively to enhance financial prediction accuracy and trading profitability across different market environments. Binary encoding classification establishes the foundation by transforming complex return predictions into directional forecasts, with US market MLP showing IC improvements from 0.018 to 0.024 and SR gains from 0.584 to 0.855. The latent space fusion module builds upon this by capturing temporal patterns, as evidenced by Hong Kong LSTM results where IC improves from 0.013 to 0.018 and ICIR from 0.142 to 0.159. The confident prediction mechanism further enhances performance through dual approaches: CUBICmean w/o trade increases IC from 0.005 to 0.064 in Hong Kong Transformer, while CUBICtrend w/o trade achieves optimal IC of 0.069 in Chinese MLP. Trading optimization completes the framework by dynamically adjusting positions based on prediction confidence, with US MLP demonstrating gains through CUBICtrend: AR increases from 0.056 to 0.165 and SR reaches 1.655. The Chinese MLP with CUBICtrend exemplifies peak performance across all metrics. Notably, all 9 model-market combinations show positive improvements while the highest performing configuration achieves SR of 1.813, proving universal applicability across diverse architectures and market conditions. Quantitative analysis reveals that trend-based and mean-based confidence mech-

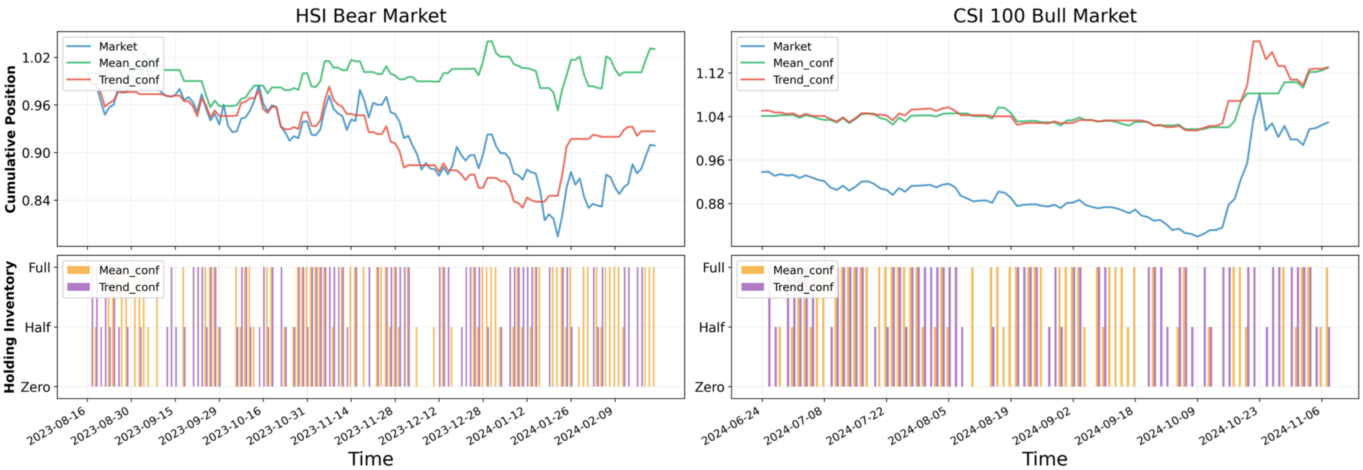

Fig. 2. Ablation for Confidence-guided Trading in Chinese (2024.06-2024.11) and HK (2023.08-2024.02) Stock Market

TABLE V
COMPARATIVE PERFORMANCE OF CUBIC ACROSS MARKETS: MODULE CONFIGURATIONS WITH DIFFERENT BASE MODELS.

		Hong Kong					USA					China				
		IC	ICIR	DA	SR	AR	IC	ICIR	DA	SR	AR	IC	ICIR	DA	SR	AR
MLP	Reg	0.009	0.069	0.469	0.376	0.038	0.018	0.130	0.492	0.584	0.056	0.012	0.193	0.464	0.385	0.037
	Reg+FS	0.008	0.119	0.489	0.330	0.036	0.014	0.133	0.504	0.682	0.075	0.018	0.187	0.469	0.475	0.052
	BN	0.013	0.197	0.490	0.413	0.041	0.024	0.249	0.525	0.855	0.089	0.024	0.286	0.486	0.444	0.053
	BF	0.016	0.274	0.503	0.431	0.054	0.027	0.284	0.517	0.916	0.092	0.024	0.257	0.483	0.457	0.059
	CUBIC$_{\text{mean w/o trade}}$	0.039	0.465	0.549	0.914	0.130	0.044	0.512	0.538	0.893	0.125	0.060	0.620	0.548	0.929	0.139
	CUBIC$_{\text{trend w/o trade}}$	0.046	0.548	**0.569**	1.026	0.102	0.042	**0.658**	0.568	0.899	0.135	0.069	0.650	0.541	0.918	0.142
	CUBIC$_{\text{mean}}$	**0.058**	0.553	0.565	**1.330**	0.140	**0.044**	0.577	0.562	1.204	0.157	0.072	0.693	**0.550**	**0.955**	0.167
	CUBIC$_{\text{trend}}$	0.050	**0.635**	0.551	1.141	**0.157**	0.041	0.599	**0.575**	**1.655**	**0.165**	**0.076**	**0.759**	0.543	0.933	**0.177**
LSTM	Reg	-0.014	-0.130	0.442	0.328	0.020	0.007	0.090	0.495	0.293	0.035	0.003	0.010	0.458	0.532	0.063
	Reg+FS	0.013	0.142	0.473	0.413	0.028	0.010	0.136	0.502	0.637	0.068	0.006	0.138	0.462	0.499	0.074
	BN	0.012	0.165	0.480	0.452	0.032	0.013	0.131	0.505	0.531	0.064	0.023	0.235	0.475	0.626	0.085
	BF	0.018	0.159	0.473	0.433	0.045	0.020	0.232	0.508	0.560	0.067	0.022	0.238	0.465	0.737	0.096
	CUBIC$_{\text{mean w/o trade}}$	0.057	0.577	0.559	0.791	0.093	0.044	0.524	0.552	0.807	0.121	0.038	0.435	0.535	1.025	0.125
	CUBIC$_{\text{trend w/o trade}}$	0.063	0.740	0.551	0.811	0.101	0.047	0.519	0.558	0.829	0.124	0.036	**0.499**	0.545	1.182	0.135
	CUBIC$_{\text{mean}}$	**0.070**	0.722	0.564	**0.883**	0.125	0.050	**0.581**	**0.566**	**1.531**	0.153	**0.040**	0.402	0.545	1.344	**0.175**
	CUBIC$_{\text{trend}}$	0.069	**0.763**	**0.568**	0.867	**0.126**	**0.050**	0.536	0.559	1.169	**0.164**	0.039	0.483	**0.558**	**1.594**	0.171
TF	Reg	0.005	0.046	0.497	0.417	0.047	0.021	0.234	0.515	0.619	0.062	-0.005	-0.099	0.500	0.446	0.053
	Reg+FS	0.006	0.076	0.503	0.413	0.050	0.023	0.257	0.528	0.513	0.077	0.002	0.070	0.497	0.410	0.061
	BN	0.014	0.142	0.520	0.776	0.076	0.029	0.310	0.523	0.651	0.078	0.028	0.164	0.531	0.728	0.095
	BF	0.022	0.294	0.524	0.644	0.074	0.025	0.450	0.531	0.712	0.085	0.025	0.181	0.539	0.673	0.092
	CUBIC$_{\text{mean w/o trade}}$	0.064	0.329	0.553	0.957	0.153	0.042	0.512	0.572	1.111	0.144	0.043	0.497	0.581	0.877	0.096
	CUBIC$_{\text{trend w/o trade}}$	0.066	**0.791**	0.553	0.953	0.157	0.041	0.446	**0.579**	1.295	0.155	0.053	0.067	0.574	0.848	0.107
	CUBIC$_{\text{mean}}$	0.074	0.709	**0.557**	1.074	0.180	0.044	**0.580**	0.572	**1.373**	**0.179**	**0.057**	**0.685**	0.574	1.356	0.177
	CUBIC$_{\text{trend}}$	**0.075**	0.708	0.551	**1.162**	**0.192**	**0.046**	0.539	0.565	1.348	0.162	0.054	0.648	**0.593**	**1.813**	**0.194**

anisms provide complementary benefits across different architectures and markets, with Chinese LSTM showing SR improvements from 1.344 to 1.594 and Hong Kong MLP achieving SR changes from 1.330 to 1.141 across volatile market conditions. The systematic performance enhancements across all evaluation metrics demonstrate the framework's effectiveness in improving both prediction accuracy and risk-adjusted returns. These consistent improvements validate the framework's robustness and practical effectiveness, where each component progressively enhances capabilities from baseline prediction to sophisticated trading execution.

We also observed that mean-based and trend-based confidence are complementary but market-dependent. Trend confidence can be advantageous when directional persistence is strong; for example, Chinese LSTM SR increases from 1.344 to 1.594. Mean confidence may be more robust when uncertainty is dominated by noisy fine-grained movements, suggesting that the preferable confidence design can vary with market regime and model inductive biases.

V. CONCLUSION

In this paper, we propose CUBIC, a plug-and-play framework for stock index prediction that targets two practical challenges: high-dimensional constituent features and instability of return regression. CUBIC integrates (i latent-space fusion to aggregate constituent information efficiently, (ii binary encoding classification to replace unstable continuous regression with bit-wise classification optimized by cross-entropy, and (iii confidence-guided regularization and trading to quantify uncertainty and scale positions accordingly. Extensive experiments across three market indices and multiple backbone architectures show that CUBIC improves both forecasting accuracy and trading performance under varying market conditions, with the magnitude of gains depending on the market and model. These results validate the effectiveness and generality of the proposed framework.

VI. Acknowledgments

This work is supported in part by the Hong Kong SAR Research Grants Council (Grant No. PolyU 15224823) and Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024A1515011524). This research is also supported by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant (Proposal ID: 23-SIS-SMU-037).

References

[1] C. Chalvatzis and D. Hristu-Varsakelis, "High-performance stock index trading via neural networks and trees," *Applied Soft Computing*, vol. 96, p. 106567, 2020.
[2] Y. Lin, Y. Yan, J. Xu, Y. Liao, and F. Ma, "Forecasting stock index price using the ceemdan-lstm model," *The North American Journal of Economics and Finance*, vol. 57, p. 101421, 2021.
[3] J. L. Coles, D. Heath, and M. C. Ringgenberg, "On index investing," *Journal of Financial Economics*, vol. 145, no. 3, pp. 665–683, 2022.
[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, p. 6000–6010.
[5] Y. Kara, M. A. Boyacioglu, and Ö. K. Baykan, "Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange," *Expert systems with Applications*, vol. 38, no. 5, pp. 5311–5319, 2011.
[6] A. Graves and A. Graves, "Long short-term memory," *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.
[7] R. Cheng and Q. Li, "Modeling the momentum spillover effect for stock prediction via attribute-driven graph attention networks," in *Proceedings of the AAAI Conference on artificial intelligence*, vol. 35, 2021, pp. 55–62.
[8] T. Li, Z. Liu, Y. Shen, X. Wang, H. Chen, and S. Huang, "Master: Market-guided stock transformer for stock price forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 162–170.
[9] J. Fan and Y. Shen, "Stockmixer: A simple yet strong mlp-based architecture for stock price forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 8389–8397.
[10] C. Wang, Y. Chen, S. Zhang, and Q. Zhang, "Stock market index prediction using deep transformer model," *Expert Systems with Applications*, vol. 208, p. 118128, 2022.
[11] A. Sethia and P. Raut, "Application of lstm, gru and ica for stock price prediction," in *Information and Communication Technology for Intelligent Systems: Proceedings of ICTIS 2018, Volume 2*. Springer, 2019, pp. 479–487.
[12] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European journal of operational research*, vol. 270, no. 2, pp. 654–669, 2018.
[13] B. Gülmez, "Stock price prediction with optimized deep lstm network with artificial rabbits optimization algorithm," *Expert Systems with Applications*, vol. 227, p. 120346, 2023.
[14] J. F. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez, and A. Troncoso, "Deep learning for time series forecasting: a survey," *Big Data*, vol. 9, no. 1, pp. 3–21, 2021.
[15] C. Cui, P. Wang, Y. Li, and Y. Zhang, "Mcvcsb: A new hybrid deep learning network for stock index prediction," *Expert Systems with Applications*, vol. 232, p. 120902, 2023.
[16] C. Liang, Y. Li, F. Ma, and Y. Zhang, "Forecasting international equity market volatility: A new approach," *Journal of Forecasting*, vol. 41, no. 7, pp. 1433–1457, 2022.
[17] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.
[18] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in neural information processing systems*, vol. 34, pp. 22 419–22 430, 2021.
[19] R. K. Dash, T. N. Nguyen, K. Cengiz, and A. Sharma, "Fine-tuned support vector regression model for stock predictions," *Neural Computing and Applications*, vol. 35, no. 32, pp. 23 295–23 309, 2023.
[20] J. Zhang, H. Liu, W. Bai, and X. Li, "A hybrid approach of wavelet transform, arima and lstm model for the share price index futures forecasting," *The North American Journal of Economics and Finance*, vol. 69, p. 102022, 2024.
[21] F. Pokou, J. Sadefo Kamdem, and F. Benhmad, "Hybridization of arima with learning models for forecasting of stock market time series," *Computational Economics*, vol. 63, no. 4, pp. 1349–1399, 2024.
[22] H. Wang, T. Wang, S. Li, J. Zheng, S. Guan, and W. Chen, "Adaptive long-short pattern transformer for stock investment selection." in *IJCAI*, 2022, pp. 3970–3977.
[23] X. Zhang, S. Qu, J. Huang, B. Fang, and P. Yu, "Stock market prediction via multi-source multiple instance learning," *IEEE Access*, vol. 6, pp. 50 720–50 728, 2018.
[24] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," *arXiv preprint arXiv:2310.06625*, 2023.
[25] S. Hochreiter, "Long short-term memory," *Neural Computation MIT-Press*, 1997.
[26] X. Ye, B. Ning, P. Bian, and X. Feng, "A self-attention-based stock prediction method using long short-term memory network architecture," in *International Conference of Pioneering Computer Scientists, Engineers and Educators*. Springer, 2023, pp. 12–24.
[27] A. F. Kamara, E. Chen, and Z. Pan, "An ensemble of a boosted hybrid of deep learning models and technical analysis for forecasting stock prices," *Information Sciences*, vol. 594, pp. 1–19, 2022.
[28] L. Stewart, F. Bach, Q. Berthet, and J.-P. Vert, "Regression as classification: Influence of task formulation on neural network features," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2023, pp. 11 563–11 582.
[29] J. Hernández-Orallo, C. Ferri, N. Lachiche, A. Martínez-Usó, and M. J. Ramírez-Quintana, "Binarised regression tasks: methods and evaluation metrics," *Data mining and knowledge discovery*, vol. 30, no. 4, pp. 848–890, 2016.
[30] M. Kyoung-Sook and K. Hongjoong, "Performance of deep learning in prediction of stock market volatility." *Economic Computation & Economic Cybernetics Studies & Research*, vol. 53, no. 2, 2019.
[31] J. Farebrother, J. Orbay, Q. Vuong, A. A. Taïga, Y. Chebotar, T. Xiao, A. Irpan, S. Levine, P. S. Castro, A. Faust *et al.*, "Stop regressing: Training value functions via classification for scalable deep rl," *arXiv preprint arXiv:2403.03950*, 2024.
[32] Y. Wang, M. Liu, and L. Zhang, "Categorical deep learning for multi-asset portfolio optimization," *Knowledge-Based Systems*, vol. 270, p. 110211, 2023.
[33] T. T. Huynh, M. H. Nguyen, T. T. Nguyen, P. L. Nguyen, M. Weidlich, Q. V. H. Nguyen, and K. Aberer, "Efficient integration of multi-order dynamics and internal dynamics in stock movement prediction," in *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, 2023, pp. 850–858.
[34] N. Rouf, M. B. Malik, T. Arif, S. Sharma, S. Singh, S. Aich, and H.-C. Kim, "Stock market prediction using machine learning techniques: a decade survey on methodologies, recent developments, and future directions," *Electronics*, vol. 10, no. 21, p. 2717, 2021.
[35] A. V. Devadoss and T. A. A. Ligori, "Forecasting of stock prices using multi layer perceptron," *International journal of computing algorithm*, vol. 2, no. 1, pp. 440–449, 2013.