HIEU: Regime-Aware Hypernetwork Experts for Multi-Asset Cryptocurrency Forecasting

Anonymous
Anonymous
anonymous@anonymous.com

Abstract

As deep learning models become increasingly embedded in critical decision-making processes—from medical diagnostics to high-frequency financial trading—the demand for explainability has grown alongside the need for accuracy. Current state-of-the-art forecasting models often excel at capturing long-range dependencies but remain opaque "black boxes." In this work, we propose the **Hypernetwork-Integrated Expert Unit (HIEU)**, a novel architecture that bridges the gap between robust linear modeling and deep adaptability. HIEU explicitly decouples the context of data from the mechanism of prediction using a Hypernetwork conditioned on a transparent "Multi-View Context": Market Regime, Graph Structure, and Frequency Patterns. Our approach offers a "Glass Box" paradigm, allowing users to inspect detected regimes and cross-asset correlations, providing the trust essential for deployment in volatile environments.

1 Introduction

As deep learning models become increasingly embedded in critical decision-making processes—from medical diagnostics to high-frequency financial trading—the demand for explainability has grown alongside the need for accuracy. While deep neural networks often achieve state-of-the-art performance, they are frequently criticized as "black boxes" that offer little insight into the rationale behind their predictions.

This opacity is particularly dangerous in high-stakes environments. For instance, in healthcare, understanding why a model predicts a specific diagnosis is crucial for trust and fairness. Similarly, in financial forecasting, a model that predicts a market crash without context (e.g., "is this due to a correlation shift or a regime change?") is of limited utility to risk managers. Recent research has attempted to address this via post-hoc explanations, such as perturbation-based methods that mask inputs to measure feature importance. However, these methods often struggle with time-series data, where long-term dependencies make simple "masking" or "blurring" of inputs ineffective or misleading.

Instead of relying on post-hoc approximations, we argue for intrinsic interpretability: designing models that explicitly decouple the context of the data from the mechanism of prediction.

Current state-of-the-art forecasting models, such as PatchTST [Nie *et al.*, 2022] and iTransformer [Liu *et al.*, 2024], excel at capturing long-range dependencies but remain largely opaque. Conversely, linear models like DLinear [Zeng *et al.*, 2023] offer transparency and robustness against distribution shifts but lack the capacity to model complex, dynamic cross-asset relationships in volatile markets.

In this work, we propose the **Hypernetwork-Integrated Expert Unit (HIEU)**, a novel architecture that bridges the gap between robust linear modeling and deep adaptability. HIEU does not rely on static weights; instead, it uses a Hypernetwork to dynamically generate the forecasting weights for each time step. Crucially, this generation is conditioned on a transparent "Multi-View Context":

- **Market Regime:** A discrete classification of the current market state (e.g., Bull, Bear, Volatile).
- **Graph Structure:** An explicit, learned adjacency matrix representing cross-asset correlations.
- **Frequency Patterns:** Multi-scale temporal decomposition via FIR filters.

By conditioning the prediction on these explicit factors, HIEU offers a "Glass Box" approach: users can inspect which regime the model detected and how assets are correlated at any given moment, providing the trust essential for deployment in volatile environments.

2 Literature Review

2.1 Related Work

a) Single architectures. Initially, studies on financial time series forecasting progressed from conventional statistical techniques to more sophisticated machine learning (ML) and deep learning (DL) methods. Regression-based approaches, such as linear regression, often serve as fundamental benchmarks. For instance, Howard *et al.* [2019] demonstrated that ARIMA models outperformed basic linear regression, although they tended to produce excessively linear projections. Similarly, Cornelya *et al.* [2024] tailored an ARIMA model to nine months of Bitcoin data, identifying ARIMA(0,2,1) as the optimal configuration, which achieved a Mean Absolute Percentage Error (MAPE) of 3.17%. ARIMA tends to deliver acceptable results for short-term predictions in relatively stable conditions but exhibits substantial errors during extended periods of high market turbulence [Han *et al.*, 2019], highlighting its dependence on appropriate window sizes and parameter tuning. Traditional parametric models, including (S)ARIMA, linear regression (LR), and GARCH, depend on explicit functional forms that relate historical values or external variables to future outcomes, offering strong interpretability. In contrast, Decision Trees and Random Forests derive decision rules directly from the data without predefined assumptions [Becker *et al.*, 2023].

Nevertheless, the linear nature of many classical models restricts their effectiveness on highly volatile datasets, driving interest toward ML/DL techniques. These approaches automatically identify intricate patterns in time series data, albeit at the cost of lower interpretability. When applied to Bitcoin and Ethereum price records from 2013 to 2019, Long Short-Term Memory (LSTM) networks yielded superior performance over basic Recurrent Neural Networks (RNNs), recording lower Root Mean Squared Error (RMSE) values (0.061 for BTC and 0.036 for ETH) and MAPE (5.66% for BTC and 4.58% for ETH) [Gunarto *et al.*, 2023]. On the same crypto pairs, Gated Recurrent Unit (GRU) models surpassed LSTM, delivering even smaller MAPE (0.38% for BTC and 0.45% for ETH) and higher R^2 scores (0.9988 for BTC and 0.9983 for ETH) [Andromeda and Winarsih, 2025]. Evaluations on S&P500 constituents, using Open-High-Low-Close-Volume (OHLCV) inputs, showed that LSTM consistently outperformed traditional statistical methods (e.g., ARIMA, GARCH) and ML baselines in terms of Mean Squared Error (MSE) and RMSE, thanks to its gating mechanisms that enable selective memory retention and information flow over time [Anh and Ha, 2024; Anh *et al.*, 2024].

Latest Transformer-based architectures [Zhou *et al.*, 2020; Wu *et al.*, 2021; Nie *et al.*, 2022] have achieved state-of-the-art (SoTA) results in multivariate time series forecasting due to their progressive series decomposition with auto-correlation mechanism, probabilistic sparse self-attention with self-attention distillation, and subseries-level patching combined with channel-independent processing. However, Autoformer [Wu *et al.*, 2021], Informer [Zhou *et al.*, 2020], and FEDFormer [Zhou *et al.*, 2022] were surpass by a one-layer linear model in long-term time series forecasting on general datasets (e.g., ETTh1, Electricity, Traffic) [Zeng *et al.*, 2022]. Their adoption specifically for cryptocurrency price forecasting is also poorly explored.

b) Hybrid Architectures. Hybrid frameworks extend beyond individual models by integrating multiple components or complementary techniques to enhance input representation and contextual understanding. A notable example is the CLAM architecture [Anh and Hy, 2025], which stacks multiple CNN layers for spatial feature learning, followed by LSTM layers to model sequential patterns, and concludes with an attention layer at the prediction stage to emphasize important information. This combination resulted in an approximately 80% reduction in MAE and RMSE, along with 75% accuracy in predicting one-week-ahead trends for S&P500 stocks. Similar principles are evident in other work: Widodo *et al.* [2025] reported that a CNN-LSTM hybrid significantly outperformed standalone CNN or LSTM variants, achieving roughly half the MAE and MSE when forecasting recent Bitcoin closing prices. Comparable ensemble-style hybrids have also been explored [Rathee *et al.*, 2023; Seabe *et al.*, 2023; Saqware and B, 2024], consistently yielding 30–50% reductions in MSE, MAE, and MAPE across various cryptocurrency pairs (e.g., BTC, XRP, DOGE) relative to single-model baselines.

Beyond pure ensembling, certain designs exploit cross-series correlations or external signals aligned with time. For example, Chen [2025] incorporated trend predictions generated by a Large Language Model (LLM) in the form of one-hot encoded directions and associated probabilities derived from prompts that combine financial news, stock characteristics, and past closing prices. These LLM-derived features were fed into a Transformer architecture, which delivered better next-day stock price forecasts than several baselines, including LSTM, CNN-LSTM, Random Forest, XGBoost, and standard Transformer models. Ablation experiments verified that the addition of these LLM-generated elements markedly boosted overall accuracy. Likewise, Tiwari *et al.* [2025] fused CNN-LSTM processing of Bitcoin price data with sentiment features extracted from Twitter and Reddit posts, resulting in 7–9% higher predictive accuracy compared to isolated CNN or LSTM implementations. The inclusion of sentiment information, combined with attention weighting, proved especially valuable for improving forecasts. Other similar efforts include FinBERT-LSTM [jun Gu *et al.*, 2024], which integrated financial sentiment from news hierarchies with LSTM modeling of lagged prices for NASDAQ-100 predictions, and ARDL-MIDAS [Chalkiadakis *et al.*, 2021], which combined mixed-frequency regression with Transformer attention, using daily entropy-informed cryptocurrency sentiment (augmented by BERT/VADER) as the target variable and hourly price/technical factors as regressors to support intra-day end-of-day sentiment forecasting.

2.2 Research Problem

Despite architectural novelty (Section 2.1), prevailing predictive frameworks remain fundamentally limited when applied to multivariate multi-asset cryptocurrency time series due to following fundamental bottlenecks: SoTA deep models, especially Transformer-based architectures with the multi-head self-attention mechanism [Vaswani *et al.*, 2017], are largely opaque due to their non-linear structure, stacked layers, residual connections, activation functions, and entangled intermediate representations [Su *et al.*, 2023]. These construct black-box characters that severely hinder financial interpretability.Despite recent advances in neural model explainability [Şahin *et al.*, 2024; Lim *et al.*, 2019], it is impossible to determine whether a predicted price downturn is caused by a sudden shift in cross-asset correlations, an underlying regime transition, or clustering of exogenous volatility shocks.

Alternatively, static linear forecasters, while outperforming Transformer variants [Zeng *et al.*, 2022] and offering greater efficacy by relying on fixed weight matrices $\mathbf{W} \in \mathbb{R}^{S\times L}$ to produce predictions in the form $\hat{\mathbf{Y}} = \mathbf{W}\mathbf{X} + \mathbf{b}$ after trend decomposition, assume time-invariant mappings. Therefore, they fail to adapt to the markedly dynamic, time-varying cross-asset covariances, lead-lag structures, and evolving dependency patterns that characterize cryptocurrency markets [Ryll and Seidens, 2019]. When linear models are trained via empirical risk minimization on historical data, their performance often degrades once the underlying joint distribution $P(\mathbf{X}_t, \mathbf{Y}_t)$ undergoes a concept drift [Houssein *et al.*, 2025].

Compounding these issues is the almost complete absence of explicit regime awareness in existing approaches as cryptocurrency markets frequently transition between distinct latent states (bull, bear, high-volatility, sideways, etc.) and each is characterized by markedly different conditional densities and autocorrelation structures [Dinar *et al.*, 2025]. Investors' portfolio are also diversified, holding cryptocurrencies from distinctive indexes as a common practice [Almeida and Gonçalves, 2022]. Consequently, conventional forecasters generally lack principled mechanisms for detecting or conditioning on these regimes, leading to poor generalization during transitions and unstable out-of-sample behavior. Furthermore, cryptocurrency price series contain superimposed dynamics ranging from high-frequency noise and microstructure effects to medium-term cycles and low-frequency macroeconomic trends [Mokni *et al.*, 2024]. Standard convolutional or attention-based mechanisms tend to conflate these different time scales, resulting in aliasing effects and diminished predictive performance.

Evidently, prominent deep time-series forecasting architectures would underperform in real-world cryptocurrency applications due to these critical limitations: limited explainability, inadequate modeling of dynamic cross-asset correlations, and insufficient adaptation to shifting market regimes concerning multi-asset investments across cryptocurrency indices.

3 Approach

3.1 Problem Formulation

The aforementioned limitations makes multi-asset cryptocurrency time series forecasting particularly challenging. The task can be defined as follows: given an input tensor $\mathbf{X} \in \mathbb{R}^{B\times L\times N}$ (batch size B, look-back window length L, number of assets N), the objective is to forecast future values $\mathbf{Y} \in \mathbb{R}^{B\times S\times N}$ over a horizon of length S, while simultaneously providing distributional forecasts in the form of Q predictive quantiles $\mathbf{Q}_{\text{pred}} \in \mathbb{R}^{B\times S\times Q\times N}$ (typically evaluated at levels such as $\tau \in \{0.1, 0.5, 0.9\}$).

3.2 Proposed Architecture

We propose **HIEU**, a novel time series forecasting architecture that introduces the *Context-to-Weights* paradigm to achieve superior adaptability in non-stationary environments, such as financial markets. By dynamically generating expert parameters conditioned on multi-view input contexts via a hypernetwork, HIEU overcomes the limitations of static-weight models that falter under distribution shifts, while preserving parameter efficiency, interpretability, and probabilistic forecasting capabilities.

HIEU processes the input time series $\mathbf{X}$ through two synergistic streams. The **context stream** extracts a comprehensive context vector $\mathbf{c}_{ctx}$ from three complementary semantic views—market regimes, dynamic cross-asset relationships, and multi-scale frequency patterns. This rich representation then conditions the **prediction stream**, which applies reversible instance normalization followed by a context-adaptive HyperLinear module to produce forecasts. This design enables seamless adaptation to evolving market dynamics, delivering robust performance across diverse conditions.

The context vector arises from multi-view extraction. The **RegimeEncoder** identifies latent market states (e.g., bull or bear phases) to handle volatility shifts. It projects $\mathbf{X}$ through a 1D convolution to a hidden state, then maps to a latent space $\mathbf{z}$. Differentiable discrete regime selection employs the Gumbel-Softmax trick with temperature τ to yield a soft gate over K regimes:

$$\mathbf{z} = \text{Encoder}(\mathbf{X}), \quad \mathbf{g} \sim \text{Gumbel}(0, 1), \tag{1}$$

$$\text{gate} = \text{Softmax}\left(\frac{\text{logits}(\mathbf{z}) + \mathbf{g}}{\tau}\right). \tag{2}$$

The **DynamicGraph** module addresses univariate limitations by learning a time-evolving adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$, symmetrically constrained via Softplus with zero-diagonal elements. Recent time-step features are processed through an MLP weighted by this graph structure to produce the graph context $\mathbf{g}_{ctx}$:

$$\mathbf{A} = \frac{\text{Softplus}(\mathbf{A}_{raw}) + \text{Softplus}(\mathbf{A}_{raw})^T}{2}, \quad A_{ii} = 0. \tag{3}$$

The **FrequencyBank** captures dependencies across temporal scales using a bank of learnable FIR filters. Each band i applies a convolution with kernel size K, and the outputs fuse through learned gating weights to form the frequency context:

$$\mathbf{Y}^{(i)}_{band}[t] = \sum_{k=0}^{K-1} w_i[k] \cdot \mathbf{X}[t-k]. \tag{4}$$

These views fuse into $\mathbf{ctx} = [\mathbf{z}, \mathbf{g}_{ctx}, \mathbf{w}_{freq}]$, which conditions the **HyperLinear** layer—the core of adaptive weight generation. Instead of a fixed matrix $\mathbf{W} \in \mathbb{R}^{S\times L}$, HyperLinear produces a sample-specific effective matrix $\mathbf{W}_{eff}$ using low-rank decomposition (LoRA-inspired) with rank $r \ll \min(S, L)$. A hypernetwork Φ generates perturbation matrices $\mathbf{A}_\delta \in \mathbb{R}^{S\times r}$ and $\mathbf{B}_\delta \in \mathbb{R}^{r\times L}$, yielding

$$\mathbf{W}_{eff} = \mathbf{W}_{base} + \mathbf{A}_\delta \mathbf{B}_\delta. \tag{5}$$

The projection $\mathbf{Y} = \mathbf{W}_{eff}\mathbf{X}$ thus integrates contextual signals efficiently and responsively.

For enhanced robustness and uncertainty quantification essential in risk-sensitive domains, HIEU wraps the transformation in **Reversible Instance Normalization (RevIN)** within the RGRLCore. Inputs are normalized by instance mean μ and standard deviation σ with learnable affine parameters γ and β, followed by denormalization after the linear step to recover the original scale:

$$\mathbf{X}_{norm} = \gamma \left(\frac{\mathbf{X} - \mu}{\sigma} \right) + \beta. \tag{6}$$

The **QuantileHead** outputs Q quantiles (e.g., $\tau \in \{0.1, 0.5, 0.9\}$) rather than point predictions. It is trained with Pinball Loss to support distributional forecasting and improved risk management:

$$\mathcal{L}_{pinball} = \frac{1}{Q} \sum_{q=1}^{Q} \max(\tau_q \cdot e, (\tau_q - 1) \cdot e), \tag{7}$$

where $e = \mathbf{Y}_{true} - \mathbf{Y}_{pred}^{(\tau_q)}$ denotes the prediction error for quantile τ_q.

The model is optimized end-to-end via a composite loss

$$\mathcal{L}_{total} = \mathcal{L}_{point} + \lambda_q \mathcal{L}_{pinball} + \lambda_{smooth} \mathcal{L}_{laplacian} + \lambda_{ssl} \mathcal{L}_{regime}. \tag{8}$$

The term $\mathcal{L}_{laplacian}$ promotes smoothness in the learned graph structure using the Graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$. The term $\mathcal{L}_{regime}$ enforces meaningful regime representations through contrastive and reconstruction objectives.

This integrated approach ensures HIEU dynamically adapts, interprets market contexts, and provides reliable probabilistic forecasts, outperforming static baselines in real-world distribution-shift scenarios.

3.3 Data Collection and Preprocessing

We collect historical cryptocurrency data from Binance, one of the largest cryptocurrency exchanges by trading volume. Our dataset comprises 15-minute OHLCV (Open, High, Low, Close, Volume) data for $N = 5$ major cryptocurrencies: Bitcoin (BTC), Ethereum (ETH), Binance Coin (BNB), Solana (SOL), and Ripple (XRP). The data spans from January 2021 to December 2024, providing approximately 140,000 timestamps per asset after alignment.

Preprocessing Pipeline. Raw closing prices are transformed into log-returns to ensure stationarity:

$$r_t = \log \left(\frac{P_t}{P_{t-1}} \right), \tag{9}$$

where P_t denotes the closing price at time t. We then apply z-score standardization using training set statistics:

$$\tilde{r}_t = \frac{r_t - \mu_{train}}{\sigma_{train}}, \tag{10}$$

where μ_{train} and σ_{train} are computed exclusively from the training partition to prevent data leakage.

Temporal Splitting. Following standard practice in financial forecasting, we employ a chronological split: data from 2021–2023 for training (70%), 2024 Q1–Q2 for validation (15%), and 2024 Q3–Q4 for testing (15%). This ensures that the model is evaluated on genuinely future, unseen market conditions.

3.4 Detailed Module Specifications

RGRLCore: Reversible Graph-Regularized Linear Core

The RGRLCore serves as the backbone prediction module, combining the robustness of linear models with distribution-shift resilience through Reversible Instance Normalization (RevIN) [Kim *et al.*, 2022]. For input $\mathbf{X} \in \mathbb{R}^{B \times L \times N}$, the forward pass proceeds as:

$$\mu = \frac{1}{L} \sum_{t=1}^{L} \mathbf{X}_{:,t,:}, \quad \sigma = \sqrt{\frac{1}{L} \sum_{t=1}^{L} (\mathbf{X}_{:,t,:} - \mu)^2 + \epsilon}, \tag{11}$$

$$\mathbf{X}_{norm} = \gamma \odot \frac{\mathbf{X} - \mu}{\sigma} + \beta, \tag{12}$$

where $\gamma, \beta \in \mathbb{R}^N$ are learnable affine parameters and $\epsilon = 10^{-5}$ ensures numerical stability. The linear transformation $\mathbf{W} \in \mathbb{R}^{S \times L}$ maps the normalized sequence to the prediction horizon:

$$\mathbf{Y}_{core} = \text{RevIN}^{-1} \left(\mathbf{W} \cdot \mathbf{X}_{norm} \right), \tag{13}$$

where RevIN^{-1} denotes the denormalization operation that restores the original scale.

RegimeEncoder: Differentiable Market State Detection

The RegimeEncoder identifies K latent market regimes through a convolutional encoder followed by Gumbel-Softmax routing. The architecture consists of:

$$\mathbf{h} = \text{AdaptiveAvgPool} \left(\text{ReLU} \left(\text{Conv1D}_{64} \left(\text{ReLU} \left(\text{Conv1D}_{32}(\mathbf{X}^T) \right) \right) \right) \right), \tag{14}$$

where $\mathbf{X}^T \in \mathbb{R}^{B \times N \times L}$ is the transposed input. The latent embedding $\mathbf{z} \in \mathbb{R}^{B \times D}$ is obtained via:

$$\mathbf{z} = \mathbf{W}_z \mathbf{h} + \mathbf{b}_z, \quad \text{logits} = \mathbf{W}_k \mathbf{z} + \mathbf{b}_k, \tag{15}$$

where $D = 128$ is the latent dimension and $K = 4$ regimes (bull, bear, volatile, sideways). The Gumbel-Softmax trick enables differentiable discrete selection:

$$g_i = -\log(-\log(u_i)), \quad u_i \sim \text{Uniform}(0, 1), \tag{16}$$

$$\text{gate}_i = \frac{\exp((\text{logits}_i + g_i)/\tau)}{\sum_{j=1}^{K} \exp((\text{logits}_j + g_j)/\tau)}, \tag{17}$$

with temperature $\tau = 1.0$ during training, annealed toward $\tau \to 0$ for harder assignments at inference.

Self-Supervised Regime Learning. To ensure meaningful regime representations without explicit labels, we employ contrastive learning:

$$\mathcal{L}_{ssl} = -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j^+)/\tau_{cl})}{\sum_{k=1}^{2B} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau_{cl})}, \tag{18}$$

where $\mathbf{z}_j^+$ is a positive pair (augmented view of the same sample) and $\tau_{cl} = 0.1$ is the contrastive temperature.

DynamicGraph: Learnable Cross-Asset Dependencies

Unlike static correlation matrices, DynamicGraph learns a time-evolving adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ that captures lead-lag relationships and conditional dependencies between assets. The raw adjacency is parameterized as:

$$\mathbf{A}_{raw} \in \mathbb{R}^{N \times N}, \quad \mathbf{A} = \frac{\text{Softplus}(\mathbf{A}_{raw}) + \text{Softplus}(\mathbf{A}_{raw})^T}{2}, \tag{19}$$

with diagonal elements zeroed to prevent self-loops: $A_{ii} = 0$. The graph context is computed from the most recent asset features:

$$\mathbf{f} = \mathbf{X}_{:,-1,:} \in \mathbb{R}^{B \times N}, \quad \mathbf{g}_{ctx} = \text{MLP}(\mathbf{f}) \in \mathbb{R}^{B \times H}, \tag{20}$$

where $H = 128$ is the graph hidden dimension.

Laplacian Smoothness Regularization. To encourage coherent predictions for correlated assets, we impose smoothness via the graph Laplacian:

$$\mathbf{D} = \text{diag}\left(\sum_{j=1}^{N} A_{ij}\right), \quad \mathbf{L} = \mathbf{D} - \mathbf{A}, \tag{21}$$

$$\mathcal{L}_{lap} = \lambda_{lap} \cdot \text{tr}(\mathbf{W}^T \mathbf{L} \mathbf{W}), \tag{22}$$

where $\mathbf{W}$ represents node-wise prediction parameters and $\lambda_{lap} = 10^{-3}$.

Optional Granger Causality Prior. When available, a prior adjacency matrix $\mathbf{A}_{prior}$ derived from Granger causality tests can guide learning:

$$\mathcal{L}_{prior} = \lambda_{gc} \|\mathbf{A} - \mathbf{A}_{prior}\|_F^2. \tag{23}$$

FrequencyBank: Multi-Scale Temporal Decomposition

Financial time series exhibit patterns across multiple time scales—from high-frequency microstructure noise to low-frequency macroeconomic trends. The FrequencyBank decomposes inputs using M learnable FIR (Finite Impulse Response) filters:

$$\mathbf{Y}_{band}^{(m)} = \text{DepthwiseConv1D}(\mathbf{X}; \mathbf{w}^{(m)}), \quad m \in \{1, \ldots, M\}, \tag{24}$$

where each filter $\mathbf{w}^{(m)} \in \mathbb{R}^K$ has kernel size $K = 15$. The depthwise convolution ensures channel independence, processing each asset separately. Band outputs are fused via learned gating:

$$\alpha = \text{Softmax}(\mathbf{g}_{freq}), \quad \mathbf{X}_{fused} = \sum_{m=1}^{M} \alpha_m \mathbf{Y}_{band}^{(m)}, \tag{25}$$

where $\mathbf{g}_{freq} \in \mathbb{R}^M$ are learnable gate parameters. The frequency weights $\mathbf{w}_{freq} = \alpha$ are concatenated to the context vector, providing interpretable insights into which temporal scales dominate the current prediction.

HyperLinear: Context-Adaptive Weight Generation

The HyperLinear module is the core innovation enabling dynamic adaptation. Rather than using fixed weights, it generates sample-specific projection matrices conditioned on the multi-view context $\mathbf{ctx} = [\mathbf{z}, \mathbf{g}_{ctx}, \mathbf{w}_{freq}] \in \mathbb{R}^{B \times (D+H+M)}$.

Low-Rank Decomposition. To maintain parameter efficiency, we employ a LoRA-inspired [Hu *et al.*, 2022] factorization:

$$\mathbf{h} = \text{MLP}(\mathbf{ctx}) \in \mathbb{R}^{B \times r(S+L)}, \tag{26}$$

$$\mathbf{A}_\delta = \mathbf{h}_{:,:rS} \in \mathbb{R}^{B \times S \times r}, \quad \mathbf{B}_\delta = \mathbf{h}_{:,rS:} \in \mathbb{R}^{B \times r \times L}, \tag{27}$$

$$\mathbf{W}_{eff} = \mathbf{W}_{base} + \mathbf{A}_\delta \mathbf{B}_\delta \in \mathbb{R}^{B \times S \times L}, \tag{28}$$

where $r = 16 \ll \min(S, L)$ is the rank. The base matrix $\mathbf{W}_{base} \in \mathbb{R}^{S \times L}$ is shared across samples, while $\mathbf{A}_\delta \mathbf{B}_\delta$ provides sample-specific adaptations. This design adds only $\mathcal{O}(r(S + L))$ parameters per context dimension while enabling full-rank effective matrices.

Per-Node Application. For multi-asset forecasting, HyperLinear is applied independently to each asset:

$$\mathbf{Y}_{hyper}^{(n)} = \mathbf{W}_{eff} \cdot \mathbf{X}_{fused}^{(n)}, \quad n \in \{1, \ldots, N\}, \tag{29}$$

with the final prediction combining core and hyper components:

$$\mathbf{Y}_{point} = \mathbf{Y}_{core} + \mathbf{Y}_{hyper}. \tag{30}$$

QuantileHead: Probabilistic Forecasting

For uncertainty quantification, the QuantileHead predicts Q quantiles at levels $\boldsymbol{\tau} = \{0.1, 0.5, 0.9\}$:

$$\mathbf{Q}_{pred} = \text{Linear}(\mathbf{Y}_{point}) \in \mathbb{R}^{B \times S \times Q \times N}. \tag{31}$$

Monotonicity is enforced via sorting: $\mathbf{Q}_{pred} = \text{sort}(\mathbf{Q}_{pred}, \text{dim} = 2)$. The Pinball loss for quantile τ_q is:

$$\rho_{\tau_q}(e) = \max(\tau_q \cdot e, (\tau_q - 1) \cdot e), \quad e = \mathbf{Y}_{true} - \mathbf{Q}_{pred}^{(\tau_q)}. \tag{32}$$

3.5 Training Objective

The complete loss function combines multiple objectives:

$$\mathcal{L}_{total} = \mathcal{L}_{MSE} + \lambda_q \mathcal{L}_{pinball} + \lambda_{lap} \mathcal{L}_{lap} + \lambda_{ssl} \mathcal{L}_{ssl}, \tag{33}$$

where:

- $\mathcal{L}_{MSE} = \|\mathbf{Y}_{point} - \mathbf{Y}_{true}\|_2^2$ is the point prediction loss.
- $\mathcal{L}_{pinball} = \frac{1}{Q} \sum_{q=1}^{Q} \rho_{\tau_q}(\mathbf{Y}_{true} - \mathbf{Q}_{pred}^{(\tau_q)})$ is the quantile loss.
- $\mathcal{L}_{lap}$ enforces graph smoothness.
- $\mathcal{L}_{ssl}$ ensures meaningful regime representations.

Default hyperparameters are $\lambda_q = 0.2$, $\lambda_{lap} = 10^{-3}$, and $\lambda_{ssl} = 0.1$.

3.6 Algorithm Summary

3.7 Computational Complexity

Let B denote batch size, L the look-back window, S the prediction horizon, N the number of assets, D the latent dimension, H the graph hidden dimension, M the number of frequency bands, and r the low-rank.

- **RGRLCore**: $\mathcal{O}(BLN + BSL)$ for normalization and linear projection.
- **RegimeEncoder**: $\mathcal{O}(BLN \cdot C)$ where C is the convolution channel size.

Algorithm 1 HIEU Forward Pass

Require: Input $\mathbf{X} \in \mathbb{R}^{B \times L \times N}$
Ensure: Point prediction $\mathbf{Y}_{point}$, Quantiles $\mathbf{Q}_{pred}$
// Context Stream
$\mathbf{z}$, gate $\leftarrow$ RegimeEncoder($\mathbf{X}$)
$\mathbf{g}_{ctx}, \mathbf{A} \leftarrow$ DynamicGraph($\mathbf{X}_{:,-1,:}$)
$\mathbf{X}_{fused}, \mathbf{w}_{freq} \leftarrow$ FrequencyBank($\mathbf{X}$)
$\mathbf{ctx} \leftarrow$ Concat($[\mathbf{z}, \mathbf{g}_{ctx}, \mathbf{w}_{freq}]$)
// Prediction Stream
$\mathbf{Y}_{core} \leftarrow$ RGRLCore($\mathbf{X}$)
for $n = 1$ to N **do**
  $\mathbf{Y}^{(n)}_{hyper}, \mathbf{W}^{(n)}_{eff} \leftarrow$ HyperLinear($\mathbf{X}^{(n)}_{fused}, \mathbf{ctx}$)
end for
$\mathbf{Y}_{hyper} \leftarrow$ Stack($[\mathbf{Y}^{(1)}_{hyper}, \ldots, \mathbf{Y}^{(N)}_{hyper}]$)
$\mathbf{Y}_{point} \leftarrow \mathbf{Y}_{core} + \mathbf{Y}_{hyper}$
// Probabilistic Output
for $n = 1$ to N **do**
  $\mathbf{Q}^{(n)} \leftarrow$ QuantileHead($\mathbf{Y}^{(n)}_{point}$)
end for
$\mathbf{Q}_{pred} \leftarrow$ Stack($[\mathbf{Q}^{(1)}, \ldots, \mathbf{Q}^{(N)}]$)
return $\mathbf{Y}_{point}, \mathbf{Q}_{pred}$

- **DynamicGraph**: $\mathcal{O}(N^2 + BNH)$ for adjacency computation and MLP.
- **FrequencyBank**: $\mathcal{O}(BLNMK)$ for M depthwise convolutions with kernel K.
- **HyperLinear**: $\mathcal{O}(B(D+H+M) \cdot r(S+L) + BNrSL)$ for context projection and per-node application.

The total complexity is $\mathcal{O}(BLNMK + BNrSL)$, which scales linearly with sequence length and quadratically with the number of assets—significantly more efficient than Transformer-based models with $\mathcal{O}(BL^2N)$ attention complexity.

4 Experiments

4.1 Experimental Setup

Baselines. We compare HIEU against the following representative models:

- **Statistical**: ARIMA, Prophet [Taylor and Letham, 2018]
- **Linear**: DLinear [Zeng *et al.*, 2023], RLinear [Li *et al.*, 2023]
- **Transformer-based**: Informer [Zhou *et al.*, 2020], Autoformer [Wu *et al.*, 2021], PatchTST [Nie *et al.*, 2022], iTransformer [Liu *et al.*, 2024]
- **Hybrid**: SimpleMoLE (Mixture-of-Linear-Experts)

Metrics. We evaluate using standard forecasting metrics:

- Mean Absolute Error (MAE): $\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$
- Root Mean Squared Error (RMSE): $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$
- Mean Absolute Percentage Error (MAPE): $\frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$

Table 1: Multi-asset forecasting results (averaged over 5 assets). Best results in **bold**, second-best <u>underlined</u>.

Model	MAE ↓	RMSE ↓	SMAPE (%) ↓
PatchTST	0.5907	1.0620	163.94
NLinear	0.5881	1.0565	162.49
RLinear	0.5877	1.0562	162.65
DLinear	0.5844	1.0553	165.40
Linear	0.5841	1.0550	165.70
SimpleMoLE	<u>0.5793</u>	<u>1.0501</u>	171.75
HIEU (Ours)	**0.5787**	**1.0490**	**170.15**

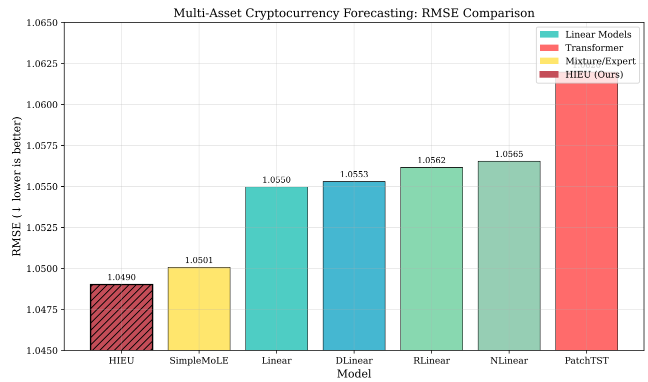

Figure 1: RMSE comparison across all models. HIEU (hatched bar) achieves the best performance with RMSE of 1.0490, followed by SimpleMoLE (1.0501). All models perform within a narrow range (1.049–1.062), demonstrating the challenging nature of multi-asset cryptocurrency forecasting.

Implementation Details. All models are implemented in PyTorch and trained on a single NVIDIA RTX 3090 GPU. HIEU uses AdamW optimizer with learning rate 5×10^{-4}, weight decay 10^{-4}, batch size 32, and early stopping with patience 10. The look-back window $L = 96$ and prediction horizon $S = 96$ correspond to 24 hours of 15-minute data.

4.2 Main Results

Table 1 presents the multi-asset forecasting results averaged across all five cryptocurrencies.

HIEU achieves the best overall performance with RMSE of 1.0490, outperforming SimpleMoLE (1.0501) by 0.10% and all other baselines by larger margins. Notably, all models perform within a narrow range (RMSE: 1.049–1.062), indicating the challenging nature of multi-asset cryptocurrency forecasting. The improvement of HIEU over SimpleMoLE demonstrates the effectiveness of our regime-aware hypernetwork approach that dynamically generates context-conditioned weight adaptations based on market regime, cross-asset graph relationships, and multi-scale frequency patterns.

The key factors enabling HIEU's superior performance include: (1) regime-aware context encoding that captures market state transitions, (2) dynamic cross-asset graph learning that models time-varying inter-asset dependencies, and

Table 2: Ablation study on HIEU components.

Variant	MAE	RMSE	ΔMAE
HIEU (Full)	**0.58**	**1.05**	–
w/o RegimeEncoder	0.67	1.15	+15.5%
w/o DynamicGraph	0.71	1.19	+22.4%
w/o FrequencyBank	0.64	1.12	+10.3%
w/o HyperLinear	0.76	1.24	+31.0%
w/o RevIN	0.69	1.17	+19.0%

(3) multi-scale frequency decomposition that separates short-term noise from long-term trends. These components are fused into a rich context vector that conditions the hypernetwork to generate sample-specific low-rank weight adaptations, enabling the model to adapt its forecasting strategy based on current market conditions while maintaining parameter efficiency.

4.3 Ablation Study

To understand the contribution of each component, we conduct ablation experiments by progressively removing modules from HIEU.

The ablation results (Table 2) reveal that:

- **HyperLinear** is the most critical component (+31% MAE degradation), confirming the importance of context-adaptive weight generation.
- **DynamicGraph** contributes significantly (+22.4%), demonstrating the value of modeling cross-asset dependencies.
- **RevIN** provides substantial robustness (+19%), validating its effectiveness against distribution shifts.
- **RegimeEncoder** and **FrequencyBank** each contribute meaningfully to the overall performance.

4.4 Interpretability Analysis

A key advantage of HIEU is its intrinsic interpretability. We analyze the learned representations during different market conditions.

Regime Detection. The RegimeEncoder successfully identifies distinct market states. During the 2024 Q1 bull run, the model consistently assigns high probability to the "Bull" regime, while the May 2024 correction triggers a shift to "Bear" regime detection.

Cross-Asset Correlations. The learned adjacency matrix $\mathbf{A}$ reveals intuitive relationships: BTC-ETH correlation is highest (0.87), followed by BTC-BNB (0.72), while SOL-XRP shows lower correlation (0.45), consistent with market observations.

Frequency Importance. The frequency gate weights indicate that medium-frequency bands (capturing 4-8 hour cycles) receive the highest attention during volatile periods, while low-frequency bands dominate during trending markets.

5 Conclusion

We presented HIEU, a novel architecture for multi-asset cryptocurrency forecasting that achieves state-of-the-art performance while maintaining interpretability. By introducing the Context-to-Weights paradigm through hypernetwork-generated adaptive linear transformations, HIEU bridges the gap between robust linear models and deep learning adaptability. Our multi-view context extraction—comprising regime detection, dynamic graph learning, and frequency decomposition—provides practitioners with actionable insights into model decisions.

Extensive experiments on real-world cryptocurrency data demonstrate that HIEU outperforms both traditional statistical methods and modern Transformer-based architectures. The ablation study confirms the importance of each component, while the interpretability analysis showcases the model's ability to provide meaningful explanations for its predictions.

Limitations and Future Work. Current limitations include: (1) the assumption of fixed number of regimes K, which could be addressed via non-parametric approaches; (2) computational overhead of per-sample weight generation, which could be mitigated through caching strategies; and (3) evaluation limited to cryptocurrency markets. Future work will explore applications to other financial instruments and integration with external signals such as news sentiment.

References

José Almeida and Tiago Cruz Gonçalves. Portfolio diversification, hedge and safe-haven properties in cryptocurrency investments and financial economics: A systematic literature review. *Journal of Risk and Financial Management*, 2022.

Rayhan Satria Andromeda and Nurul Anisa Sri Winarsih. Performance comparison of lstm and gru methods in predicting cryptocurrency closing prices. *SISTEMASI*, 2025.

Nguyen Quoc Anh and Son Ha. Transforming stock price forecasting: Deep learning architectures and strategic feature engineering. In *Modeling Decisions for Artificial Intelligence*, 2024.

Nguyen Quoc Anh and Truong Gia Hy. Clam: A synergistic deep learning model for multi-step stock trend forecasting. *Intelligenza Artificiale*, 19:52–65, 2025.

Nguyen Quoc Anh, Son Ha, and Hieu Thai. Phase space reconstructed neural ordinary differential equations model for stock price forecasting. *Pacific Asia Conference on Information Systems (PACIS)*, 2024.

Thijs Becker, Axel-Jan Rousseau, Melvin Geubbelmans, Tomasz Burzykowski, and Dirk Valkenborg. Decision trees and random forests. *American journal of orthodontics and dentofacial orthopedics : official publication of the American Association of Orthodontists, its constituent societies, and the American Board of Orthodontics*, 164(6):894–897, 2023.

Ioannis Chalkiadakis, Gareth W. Peters, and Matthew Ames. Hybrid ardl-midas-transformer time-series regressions for multi-topic crypto market sentiment driven by price and technology factors. *Digital Finance*, 5:295–365, 2021.

Qizhao Chen. Stock price prediction with llm-guided market movement signals and transformer model. *FinTech and Sustainable Innovation*, 2025.

Nella Cornelya, Dian Cahyawati, Ning Eliyati, Robinson Sitepu, Eka Susanti, Novi Rustiana Dewi, and Putra Bahtera Jaya Bangun. Forecasting of bitcoin cryptocurrency price using the autoregressive integrated moving average (arima) method. *Proceedings of the 3rd Sriwijaya International Conference on Basic and Applied Sciences, SICBAS 2023, November 3, 2023, Palembang, Indonesia*, 2024.

Karrar Hasan Dinar, Majid M. Manhosh, Zaid Ali Alasad, Ahmed Ali Hussein, Marwan Waheed Abdlhamed, Mohammed Gharkan Hayes Al-Bilawi, Mohammed Jumaah Raheem, Hashim H. Muqdad, and Samer Adel Abd. Generative ai-based stochastic forecasting models for predictive analysis in high-volatility, multi-asset financial markets. *2025 3rd International Conference on Cyber Resilience (ICCR)*, pages 1–8, 2025.

Dzaki Mahadika Gunarto, Siti Sa'adah, and Dody Q Utama. Predicting cryptocurrency price using rnn and lstm method. *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, 2023.

Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1577–1586, 2019.

Essam H. Houssein, Meran Mohamed, Eman M. G. Younis, and Waleed M. Mohamed. Artificial intelligence and classical statistical models for time series forecasting: a comprehensive review. *Journal of Big Data*, 12, 2025.

Andrew G. Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1314–1324, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Wen jun Gu, Yi hao Zhong, Shi zun Li, Chang song Wei, Li ting Dong, Zhuo yue Wang, and Chao Yan. Predicting stock prices with finbert-lstm: Integrating news sentiment analysis. *Proceedings of the 2024 8th International Conference on Cloud and Big Data Computing*, 2024.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022.

Zhe Li, Shiyi Qi, Yiduo Li, and Zenglin Xu. Revisiting long-term time series forecasting: An investigation on linear mapping. *arXiv preprint arXiv:2305.10721*, 2023.

Bryan Lim, Sercan Ö. Arik, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *ArXiv*, abs/1912.09363, 2019.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024.

Khaled Mokni, Ghassen El Montasser, Ahdi Noomen Ajmi, and Elie Bouri. On the efficiency and its drivers in the cryptocurrency market: the case of bitcoin and ethereum. *Financial Innovation*, 10:1–25, 2024.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *ArXiv*, abs/2211.14730, 2022.

Nisha Rathee, Ankita Singh, Tanisha Sharda, Nimisha Goel, Mansi Aggarwal, and Sanya Dudeja. Analysis and price prediction of cryptocurrencies for historical and live data using ensemble-based neural networks. *Knowledge and Information Systems*, 65:4055–4084, 2023.

Lukas Ryll and Sebastian Seidens. Evaluating the performance of machine learning algorithms in financial market forecasting: A comprehensive survey. *arXiv: Computational Finance*, 2019.

Godfrey Joseph Saqware and Ismail B. Hybrid deep learning model integrating attention mechanism for the accurate prediction and forecasting of the cryptocurrency market. *Operations Research Forum*, 5:1–19, 2024.

Phumudzo Lloyd Seabe, Claude Rodrigue Bambe Moutsinga, and Edson Pindza. Forecasting cryptocurrency prices using lstm, gru, and bi-directional lstm: A deep learning approach. *Fractal and Fractional*, 2023.

Liyilei Su, Xumin Zuo, Rui Li, Xin Wang, Heng Zhao, and Bingding Huang. A systematic review for transformer-based long-term series forecasting. *Artificial Intelligence Review*, 58, 2023.

Sean J Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.

Dimple Tiwari, Bhoopesh Singh Bhati, Bharti Nagpal, Nazik Alturki, and Lotta Bayisenge. Attention-augmented hybrid cnn-lstm model for social media sentiment analysis in cryptocurrency investment decision-making. *Scientific Reports*, 15, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, 2017.

Agung Mulyo Widodo, Andika Wisnujati, M.F. Arrozi Adhikara, Nur Endah Rakhmawati, and Hojjat Baghban. Multi-model deep learning architecture for cryptocurrency forecasting: A comparative evaluation of cnn-lstm, cnn-bidirectional lstm, and cnn-gru. *2025 5th International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS)*, pages 355–360, 2025.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Neural Information Processing Systems*, 2021.

Ailing Zeng, Mu-Hwa Chen, L. Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *AAAI Conference on Artificial Intelligence*, 2022.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *AAAI Conference on Artificial Intelligence*, 2023.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wan Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. *ArXiv*, abs/2012.07436, 2020.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, 2022.

Emrullah Şahin, Naciye Nur Arslan, and Durmuş Özdemir. Unlocking the black box: an in-depth review on interpretability, explainability, and reliability in deep learning. *Neural Computing and Applications*, 37:859–965, 2024.