Neural Regime-Switching Model with Markov-Informed Attention for Short-Horizon Equity Return Forecasting

Temitope Iroko *, Abiodun Alagbada, Steve Tchoneteck

Posted Date: 19 March 2026

doi: 10.20944/preprints202603.1462.v1

Keywords: regime-switching models; Gumbel-Softmax; attention mechanisms; equity forecasting; deep learning; uncertainty quantification; cross-sectional learning

 doi:10.20944/preprints202603.1462.v1

Article

Neural Regime-Switching Model with Markov-Informed Attention for Short-Horizon Equity Return Forecasting

T.C. Iroko [1,*], A.V. Alagbada [2] and S. Tchoneteck [3]

[1] Department of Mathematics, University of Wisconsin–Milwaukee, Milwaukee, WI 53201, USA
[2] Faculty of Civil and Environmental Engineering, Bauhaus-Universität Weimar, 99423 Weimar, Germany
[3] Department of Statistics, Rice University, Houston, TX 77005, USA
* Correspondence: tciroko@uwm.edu

Abstract

Forecasting short-horizon equity returns remains a fundamental challenge in quantitative finance due to regime shifts, nonlinear market dynamics, and the trade-off between model interpretability and predictive accuracy. Traditional approaches either employ interpretable probabilistic models such as Hidden Markov Models (HMMs) that lack predictive power, or deploy opaque deep learning architectures that sacrifice economic interpretability. We propose the Neural Regime-Switching Model with Markov-Informed Attention (NRSM-MIA), an end-to-end differentiable architecture that jointly learns discrete market regimes, Markov transition dynamics, and regime-conditioned temporal patterns. Our framework introduces four key innovations: (i) differentiable regime inference via Gumbel-Softmax relaxation with learnable transition matrices, (ii) regime-conditioned multi-head attention capturing state-dependent temporal dependencies, (iii) a mixture-of-experts decoder providing regime-specific predictions, and (iv) built-in uncertainty quantification through regime entropy. Following the cross-sectional learning paradigm, we train a universal model on pooled data from 15 U.S. equities across five sectors (2020–2025), enabling the model to learn regime patterns that generalize across the market. Comprehensive experiments demonstrate that NRSM-MIA achieves superior performance compared to traditional HMM+GBR hybrids, gradient boosting methods, and neural baselines (LSTM, Transformer) across multiple metrics including MAE, RMSE, Sharpe ratio, and maximum drawdown. Ablation studies confirm each component's contribution, while learned regime structures exhibit economically meaningful interpretations corresponding to bull, bear, and high-volatility market conditions.

Keywords: Regime-switching models, Gumbel-Softmax, attention mechanisms, equity forecasting, deep learning, uncertainty quantification, cross-sectional learning

1. Introduction

Accurate prediction of short-horizon equity returns is of paramount importance for portfolio management, algorithmic trading, and risk assessment [1]. Despite decades of research, this task remains exceptionally challenging due to the complex, nonlinear, and non-stationary nature of financial markets [2]. The Efficient Market Hypothesis (EMH) posits that asset prices fully reflect all available information, suggesting that returns follow a random walk and are inherently unpredictable [3]. However, substantial empirical evidence documents the existence of predictable patterns, market anomalies, and structural regime changes that create exploitable opportunities [4,5].

A fundamental observation in financial economics is that markets operate under distinct regimes—latent states characterized by different statistical properties such as mean returns, volatility levels, and correlation structures [6]. Bull markets exhibit sustained positive returns with moderate volatility, bear markets feature negative returns with elevated uncertainty, and crisis periods demonstrate extreme

 doi:10.20944/preprints202603.1462.v1

volatility with complex cross-asset dependencies. Recognizing and adapting to these regime shifts is crucial for effective forecasting and risk management.

Traditional approaches to regime-aware forecasting have relied on Hidden Markov Models (HMMs), which provide an elegant probabilistic framework for modeling latent market states [7]. HMMs decompose observed return sequences into emissions from discrete hidden states, with transitions governed by a Markov process. This framework offers interpretable regime assignments and has been successfully applied to volatility modeling [8], asset allocation [9], and market timing [10]. However, HMMs suffer from two critical limitations: (i) the Expectation-Maximization (EM) algorithm used for parameter estimation operates separately from downstream prediction tasks, potentially leading to suboptimal regime definitions; and (ii) the linear emission distributions inadequately capture the complex, nonlinear relationships between market features and future returns.

To address the predictive limitations of standalone HMMs, researchers have proposed hybrid architectures that combine regime identification with machine learning predictors [11,12]. A representative approach is the two-stage HMM+Gradient Boosting Regressor (GBR) pipeline, where an HMM first identifies market regimes, and these regime labels are then appended as features to a gradient boosting model for return prediction. While such hybrids improve predictive accuracy, they suffer from a fundamental disconnect: the regime inference and prediction stages are optimized independently, with no mechanism for the prediction loss to influence regime definitions.

1.1. Research Gap and Motivation

The current landscape reveals a significant research gap: the absence of an end-to-end differentiable framework that jointly optimizes regime inference and return prediction within a unified architecture. Recent advances in deep learning have introduced techniques for handling discrete latent variables through continuous relaxations [13,14], opening new possibilities for neural regime-switching models. However, to our knowledge, no existing work has combined differentiable discrete regime inference with attention-based temporal modeling for equity return forecasting.

Our work is motivated by three key observations:

1. **Joint optimization potential**: End-to-end training allows prediction losses to shape regime definitions, potentially discovering regimes that are both statistically meaningful and predictively useful.
2. **Attention mechanisms for regime-dependent patterns**: Different market regimes may exhibit distinct temporal dependencies; regime-conditioned attention can adaptively focus on relevant historical patterns.
3. **Cross-sectional learning**: Training on pooled data from multiple stocks enables the model to learn universal regime patterns that generalize across the market [1].

1.2. Contributions

This paper introduces NRSM-MIA, a novel end-to-end differentiable framework that unifies regime inference, Markovian dynamics, and attention-based forecasting within a single neural architecture. It employs a Gumbel-Softmax relaxation for discrete regime inference with a learnable transition matrix, enabling joint optimization of latent states and predictive accuracy. The architecture incorporates regime-conditioned attention, where temporal query projections adapt dynamically to inferred regimes, capturing state-specific temporal dependencies. A regime-aware mixture-of-experts structure further enhances predictive flexibility by allocating specialized subnetworks to distinct market conditions, combined through soft regime probabilities. The model is trained cross-sectionally on pooled equity data with sector flags, enabling transferable regime structures across assets. Empirical evaluations show consistent improvements over both traditional and neural baselines in forecasting accuracy and strategy-level performance, highlighting the practical and methodological value of this regime-aware design.

 doi:10.20944/preprints202603.1462.v1

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the NRSM-MIA architecture. Section 4 describes our experimental setup. Section 5 presents results and analysis. Section 6 concludes with discussions and future directions.

2. Related Work

This section reviews the relevant literature across four key areas: regime-switching models in finance, machine learning approaches for return forecasting, techniques for handling discrete latent variables in neural networks, and attention mechanisms for financial time series.

2.1. Regime-Switching Models in Finance

The observation that financial markets exhibit distinct behavioral patterns across different periods has motivated extensive research into regime-switching models. Hamilton [6] pioneered this field by introducing the Markov-switching autoregressive model to capture business cycle dynamics, demonstrating that economic variables like GDP growth could be modeled as arising from distinct expansion and contraction states.

Hidden Markov Models (HMMs) generalize this framework by allowing flexible observation distributions conditional on latent states [7]. In finance, HMMs have been applied to model stock price dynamics [15], characterize volatility regimes [16], and assess credit risk [17]. Bulla [18] demonstrated that HMMs with heavy-tailed emission distributions better capture the stylized facts of financial returns, including volatility clustering and excess kurtosis.

More recent work has extended basic HMM frameworks to capture richer dynamics. Guidolin [9] provides a comprehensive survey of Markov-switching models in empirical finance. Oelschläger and Adam [19] proposed hierarchical HMMs that distinguish between slow-changing macroeconomic regimes and fast-changing local market states. Wang and Lin [10] applied regime-switching frameworks to factor investing, showing that regime-aware strategies can improve risk-adjusted returns.

Despite their interpretability, standalone HMMs face two fundamental limitations for return prediction. First, the EM algorithm used for parameter estimation optimizes a likelihood objective that is agnostic to downstream prediction tasks. Second, the typically Gaussian emission distributions cannot capture complex, nonlinear relationships between market features and future returns.

2.2. Machine Learning for Financial Forecasting

Machine learning methods have achieved notable success in financial forecasting by capturing nonlinear patterns that elude traditional econometric models. Gu et al. [1] conducted a comprehensive comparison of machine learning methods for cross-sectional return prediction, finding that tree-based methods and neural networks substantially outperform linear models. Importantly, they demonstrated that pooling data across stocks enables models to learn patterns that generalize across the market.

Support Vector Machines (SVMs) represented an early application of kernel methods to stock prediction [20,21]. Gradient boosting methods, particularly XGBoost [22] and LightGBM, have become popular due to their ability to handle heterogeneous features and model complex interactions.

Deep learning approaches offer the additional advantage of automatic feature learning from sequential data. Long Short-Term Memory (LSTM) networks [23] address the vanishing gradient problem in recurrent architectures. Fischer and Krauss [24] demonstrated that LSTMs can outperform traditional methods for daily return prediction across a broad stock universe.

The Transformer architecture [25], originally developed for natural language processing, has been adapted for time series forecasting. Lim et al. [26] introduced the Temporal Fusion Transformer, which combines recurrent processing with attention mechanisms. Zhou et al. [27] proposed the Informer architecture with sparse attention for efficient long-sequence forecasting.

However, these deep learning approaches typically treat market dynamics as stationary, ignoring the regime-switching behavior documented in the financial econometrics literature.

 doi:10.20944/preprints202603.1462.v1

2.3. Hybrid Regime-Aware Approaches

Recognizing the complementary strengths of interpretable regime models and powerful machine learning predictors, researchers have proposed various hybrid architectures.

Early hybrid work combined HMMs with classical time series models. Hassan and Nath [28] used HMM-identified regimes to select appropriate ARIMA model parameters. Hassan [11] extended this approach by incorporating fuzzy logic for regime-dependent prediction rules.

More recent work has integrated regime-switching with neural networks. Ilhan and Kozat [29] introduced a Markovian RNN where the hidden state evolution is governed by a learned Markov process. Cortese et al. [30] applied statistical jump models for regime detection combined with machine learning for tactical asset allocation.

Yuan et al. [31] proposed TFE-HMM, which combines HMM state transitions with price trend features for stock forecasting. Mirza and Pekcan [32] used symbolic dynamics and state transition graphs with convolutional-recurrent neural networks.

A common practical approach involves a two-stage pipeline: first fitting an HMM to identify market regimes, then appending regime labels as features to a gradient boosting regressor [12,33]. While this approach improves upon regime-agnostic methods, it suffers from a fundamental limitation: the regime identification and prediction stages are optimized independently.

Our work addresses this limitation by proposing an end-to-end differentiable architecture where prediction losses directly influence regime definitions.

2.4. Differentiable Discrete Latent Variables

A key technical challenge in neural regime-switching models is the non-differentiability of discrete sampling operations, which prevents gradient-based optimization through regime assignments.

The Gumbel-Softmax (also known as Concrete) distribution [13,14] provides an elegant solution through continuous relaxation of categorical distributions. Given logits $\ell \in \mathbb{R}^K$ for K categories, the Gumbel-Softmax sample is computed as shown in Equation (11). These techniques have been successfully applied in variational autoencoders with discrete latents [34], neural architecture search [35], and structured prediction [36].

2.5. Summary and Research Gap

Table 1 summarizes the positioning of our work relative to existing approaches.

Table 1. Comparison with Existing Approaches.

Approach	E2E	Markov	Attn	MoE	UQ
HMM + GBR [33]	✗	✓	✗	✗	✗
LSTM [24]	✓	✗	✗	✗	✗
Transformer [25]	✓	✗	✓	✗	✗
Markovian RNN [29]	✓	✓	✗	✗	✗
TFE-HMM [31]	✗	✓	✗	✗	✗
NRSM-MIA (Ours)	✓	✓	✓	✓	✓

E2E: End-to-end differentiable; Markov: Learnable transition dynamics; Attn: Regime-conditioned attention; MoE: Mixture-of-experts prediction; UQ: Uncertainty quantification.

The literature review reveals a clear research gap: no existing work combines (i) end-to-end differentiable regime inference, (ii) learnable Markov transition dynamics, (iii) regime-conditioned attention mechanisms, and (iv) mixture-of-experts prediction with uncertainty quantification for equity return forecasting. Our proposed NRSM-MIA architecture addresses this gap.

 doi:10.20944/preprints202603.1462.v1

3. Methodology

This section presents the Neural Regime-Switching Model with Markov-Informed Attention (NRSM-MIA). We begin with the problem formulation, then describe each architectural component with intuitive explanations followed by mathematical details.

3.1. Problem Formulation

3.1.1. The Prediction Task

Our goal is to predict tomorrow's stock return based on recent market history. Given the sequence of daily closing prices $P_1, P_2, \ldots, P_t$ for a stock, we compute the daily percentage return as:

$$R_t = 100 \times \left(\frac{P_t - P_{t-1}}{P_{t-1}} \right). \tag{1}$$

The forecasting task is to predict R_{t+1} (tomorrow's return) given information available up to time t.

3.1.2. Feature Representation

Rather than using raw prices, we construct a feature vector $\mathbf{x}_t \in \mathbb{R}^d$ containing d technical indicators computed from historical data up to time t. These features capture various aspects of market dynamics including momentum, volatility, mean-reversion signals, and sector membership.

To capture temporal patterns, we use a *lookback window* of L trading days. The model input at time t is therefore a matrix:

$$\mathbf{X}_t = \begin{bmatrix} \mathbf{x}_{t-L+1} \\ \mathbf{x}_{t-L+2} \\ \vdots \\ \mathbf{x}_t \end{bmatrix} \in \mathbb{R}^{L \times d}, \tag{2}$$

where each row represents one day's features.

3.1.3. Cross-Sectional Learning with Sector Indicators

Following the cross-sectional learning paradigm of Gu et al. [1], we train a universal model on pooled data from multiple stocks. To enable the model to differentiate between stocks and sectors, we include one-hot encoded sector indicators:

$$\mathbf{s}_i = [s_{\text{Tech}}, s_{\text{Fin}}, s_{\text{Health}}, s_{\text{Consumer}}, s_{\text{Energy}}]^\top \in \{0,1\}^5, \tag{3}$$

where $s_j = 1$ if stock i belongs to sector j, and 0 otherwise. These indicators are appended to each feature vector $\mathbf{x}_t$.

3.1.4. The Regime-Switching Hypothesis

A key assumption underlying our approach is that market dynamics are governed by latent *regimes*—unobserved states that characterize distinct market conditions. We denote the regime at time t as $z_t \in \{1, 2, \ldots, K\}$, where K is the number of regimes.

Intuitively, these regimes correspond to:

- **Regime 1 (Bull market)**: Positive average returns, moderate volatility
- **Regime 2 (Bear market)**: Negative average returns, elevated volatility
- **Regime 3 (High-volatility/Crisis)**: Near-zero mean returns, extreme volatility

The model learns these regime definitions from data rather than using predefined rules.

3.1.5. Model Outputs

Given input $\mathbf{X}_t$, our model produces three outputs:

1. **Return prediction** $\hat{R}_{t+1}$: The forecasted next-day return

 doi:10.20944/preprints202603.1462.v1

2. **Uncertainty estimate** $\hat{\sigma}_{t+1}$: A measure of prediction confidence
3. **Regime probabilities** $\mathbf{z}_t \in [0,1]^K$: The probability of being in each regime

Table 2 summarizes the key notation used throughout this section.

Table 2. Summary of Notation.

Symbol	Description
R_t	Daily percentage return at time t
P_t	Closing price at time t
$\mathbf{x}_t \in \mathbb{R}^d$	Feature vector at time t (d features)
$\mathbf{s}_i \in \{0,1\}^5$	Sector indicator vector for stock i
$\mathbf{X}_t \in \mathbb{R}^{L \times d}$	Input matrix (lookback window of L days)
$z_t \in \{1,\ldots,K\}$	Discrete regime indicator
$\mathbf{z}_t \in [0,1]^K$	Soft regime probabilities (sums to 1)
$\mathbf{A} \in \mathbb{R}^{K \times K}$	Regime transition probability matrix
$\tau > 0$	Gumbel-Softmax temperature
H	Hidden dimension of neural network
L	Lookback window length
K	Number of regimes

3.2. Architecture Overview

NRSM-MIA processes input data through four main components, as illustrated in Figure 1:

1. **Feature Encoder**: Transforms raw features into rich representations that capture temporal patterns across the lookback window.
2. **Regime Inference Module**: Identifies the current market regime in a differentiable manner, learning which regime definitions are most useful for prediction.
3. **Regime-Conditioned Attention**: Focuses on relevant historical information, with the attention pattern adapted based on the inferred regime.
4. **Mixture-of-Experts Decoder**: Makes predictions using regime-specific expert networks, combining their outputs based on regime probabilities.

 doi:10.20944/preprints202603.1462.v1

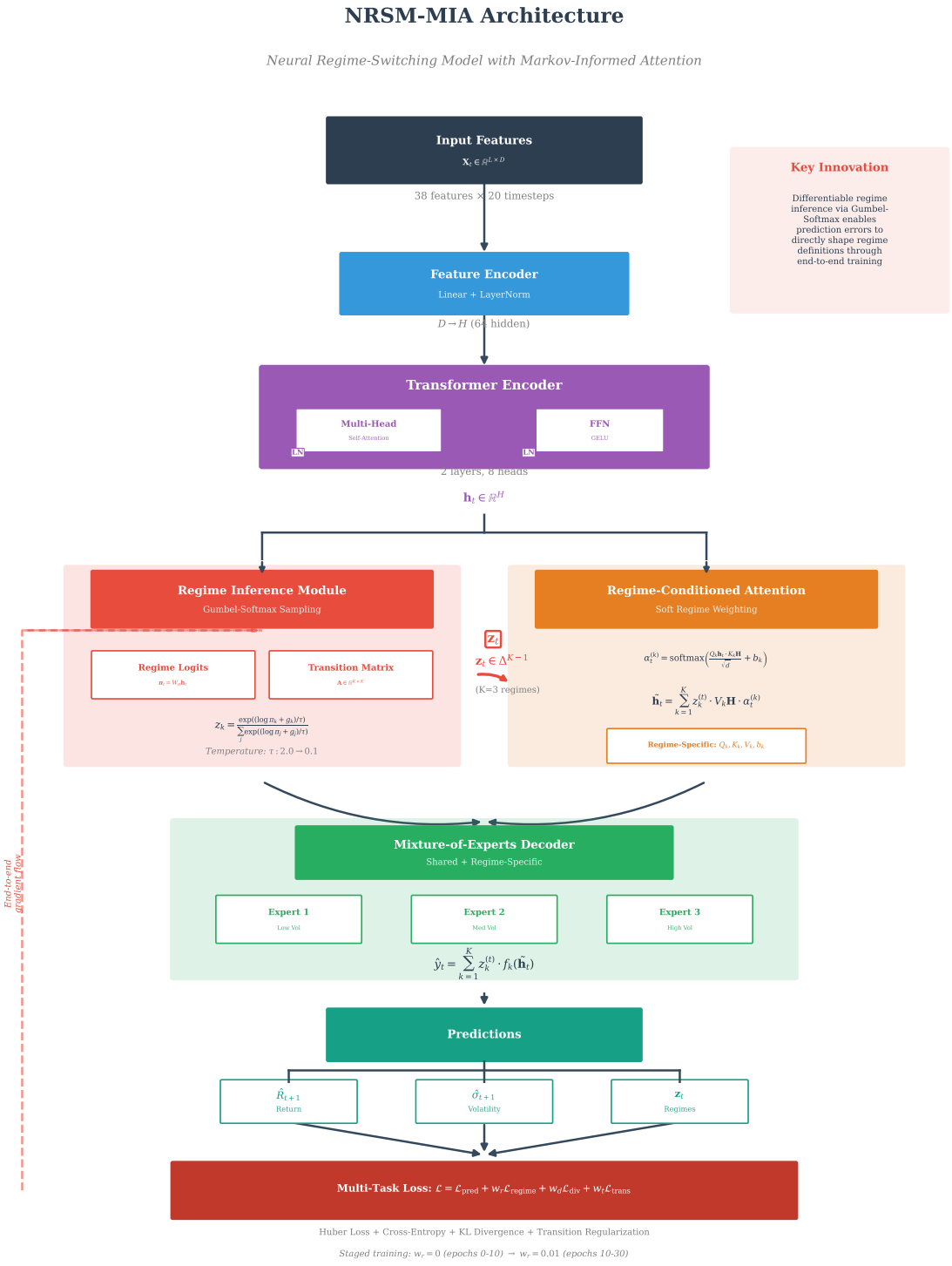

Figure 1. Overview of the NRSM-MIA architecture. The key innovation is that regime inference is differentiable, allowing prediction errors to directly influence how regimes are defined.

3.3. Component 1: Feature Encoder

The feature encoder transforms the raw input matrix $\mathbf{X}_t \in \mathbb{R}^{L \times d}$ into a sequence of hidden representations $\mathbf{H}_t \in \mathbb{R}^{L \times H}$, where H is the hidden dimension.

This encoder consists of three stages:

 doi:10.20944/preprints202603.1462.v1

Stage 1 (Input Projection): Each day's feature vector is projected to the hidden dimension:

$$\tilde{\mathbf{x}}_i = \text{GELU}(\text{LayerNorm}(\mathbf{x}_i \mathbf{W}_{\text{in}} + \mathbf{b}_{\text{in}})), \tag{4}$$

where $\mathbf{W}_{\text{in}} \in \mathbb{R}^{d \times H}$ and $\mathbf{b}_{\text{in}} \in \mathbb{R}^H$ are learnable parameters. LayerNorm [37] stabilizes training, and GELU (Gaussian Error Linear Unit) [38] provides nonlinearity.

Stage 2 (Positional Encoding): Since Transformer layers are permutation-invariant, we add learnable positional embeddings:

$$\tilde{\mathbf{x}}_i \leftarrow \tilde{\mathbf{x}}_i + \mathbf{e}_i, \tag{5}$$

where $\mathbf{e}_i \in \mathbb{R}^H$ is a learnable positional embedding for position i.

Stage 3 (Transformer Encoder): The sequence passes through N_{enc} Transformer encoder layers:

$$\mathbf{H}_t = \text{TransformerEncoder}(\tilde{\mathbf{X}}_t). \tag{6}$$

Each layer consists of multi-head self-attention and a position-wise feed-forward network with residual connections [25].

3.4. Component 2: Regime Inference Module

The regime inference module determines which market regime is currently active. Unlike traditional HMMs that use the EM algorithm, our approach makes regime inference *differentiable*.

Ideally, we would assign each time point to exactly one regime (a discrete decision). However, discrete assignments have zero gradients almost everywhere, preventing gradient-based learning. However, the Gumbel-Softmax trick [13] provides a continuous relaxation of categorical distributions.

Step 1 (Compute Regime Logits): From the encoded representation, we compute unnormalized scores (logits) for each regime:

$$\boldsymbol{\ell}_t = \text{MLP}_{\text{regime}}(\mathbf{h}_t^{(L)}), \tag{7}$$

where $\mathbf{h}_t^{(L)} \in \mathbb{R}^H$ is the hidden representation at the final time step, and $\text{MLP}_{\text{regime}} : \mathbb{R}^H \to \mathbb{R}^K$ is a multi-layer perceptron.

Step 2 (Incorporate Markov Prior): Market regimes tend to persist. We capture this through a learnable transition matrix $\mathbf{A} \in \mathbb{R}^{K \times K}$, where A_{ij} represents the probability of transitioning from regime i to regime j:

$$A_{ij} = \frac{\exp(\tilde{A}_{ij})}{\sum_{k=1}^K \exp(\tilde{A}_{ik})}, \tag{8}$$

where $\tilde{\mathbf{A}}$ is a learnable log-transition matrix.

Given the previous regime distribution $\mathbf{z}_{t-1}$, the prior for the current regime is:

$$\boldsymbol{\pi}_t = \mathbf{z}_{t-1}^\top \mathbf{A}. \tag{9}$$

We incorporate this prior by adjusting the logits:

$$\boldsymbol{\ell}_t \leftarrow \boldsymbol{\ell}_t + \log(\boldsymbol{\pi}_t + \epsilon), \tag{10}$$

where $\epsilon = 10^{-10}$ ensures numerical stability.

Step 3 (Gumbel-Softmax Sampling): We apply the Gumbel-Softmax to obtain differentiable regime probabilities:

$$z_{t,k} = \frac{\exp((\ell_{t,k} + g_k)/\tau)}{\sum_{j=1}^K \exp((\ell_{t,j} + g_j)/\tau)}, \tag{11}$$

where $g_k = -\log(-\log(u_k))$ with $u_k \sim \text{Uniform}(0, 1)$ are Gumbel noise samples, and $\tau > 0$ is the temperature parameter.

 doi:10.20944/preprints202603.1462.v1

Also, the temperature τ controls the "sharpness" of regime assignments. During training, we anneal the temperature:

$$\tau^{(e)} = \max(\tau_{\min}, \tau_0 \cdot \gamma^e), \tag{12}$$

where e is the epoch number, $\tau_0 = 1.0$ is the initial temperature, $\gamma = 0.99$ is the decay rate, and $\tau_{\min} = 0.1$ is the minimum temperature.

3.5. Component 3: Regime-Conditioned Attention

Different market regimes may require focusing on different aspects of historical data. The regime-conditioned attention mechanism adapts its focus based on the inferred regime.

And for each regime k, we maintain a separate query projection:

$$\mathbf{q}^{(k)} = \mathbf{h}_t^{(L)} \mathbf{W}_Q^{(k)}, \tag{13}$$

there $\mathbf{W}_Q^{(k)} \in \mathbb{R}^{H \times H}$ is a learnable matrix specific to regime k.

The combined query is a weighted average based on regime probabilities:

$$\mathbf{q} = \sum_{k=1}^{K} z_{t,k} \cdot \mathbf{q}^{(k)}. \tag{14}$$

Keys and values are computed from the full encoded sequence using shared projections:

$$\mathbf{K} = \mathbf{H}_t \mathbf{W}_K, \tag{15}$$

$$\mathbf{V} = \mathbf{H}_t \mathbf{W}_V. \tag{16}$$

The attention weights indicate which historical time steps are relevant:

$$\boldsymbol{\alpha} = \text{softmax}\left(\frac{\mathbf{q}\mathbf{K}^\top}{\sqrt{H}}\right), \tag{17}$$

and the attended representation is:

$$\mathbf{h}_{\text{attn}} = \text{LayerNorm}(\mathbf{h}_t^{(L)} + \boldsymbol{\alpha}\mathbf{V}). \tag{18}$$

We use multi-head attention with N_{heads} heads to capture different types of dependencies in parallel.

3.6. Component 4: Mixture-of-Experts Decoder

Each expert k is a separate feed-forward network that predicts returns:

$$\hat{R}_{t+1}^{(k)} = \text{MLP}_k(\mathbf{h}_{\text{attn}}), \tag{19}$$

where $\text{MLP}_k : \mathbb{R}^H \to \mathbb{R}$.

And the final prediction is a weighted combination of expert outputs:

$$\hat{R}_{t+1} = \sum_{k=1}^{K} z_{t,k} \cdot \hat{R}_{t+1}^{(k)}. \tag{20}$$

Likewise, we estimate prediction uncertainty from two sources:

Aleatoric uncertainty (inherent noise):

$$\hat{\sigma}^2_{\text{pred}} = \text{Softplus}(\text{MLP}_\sigma(\mathbf{h}_{\text{attn}})). \tag{21}$$

 doi:10.20944/preprints202603.1462.v1

Epistemic uncertainty (model uncertainty) through regime entropy:

$$\mathcal{H}(\mathbf{z}_t) = -\sum_{k=1}^{K} z_{t,k} \log(z_{t,k} + \epsilon). \tag{22}$$

The total uncertainty estimate combines both sources:

$$\hat{\sigma}^2_{t+1} = \hat{\sigma}^2_{\text{pred}} + \lambda_{\mathcal{H}} \cdot \mathcal{H}(\mathbf{z}_t), \tag{23}$$

where $\lambda_{\mathcal{H}} = 0.1$ is a hyperparameter.

3.7. Training Objective

The model is trained end-to-end by minimizing a composite loss function.

Considering the prediction loss, We use the Huber loss for robustness to outliers:

$$\mathcal{L}_{\text{pred}} = \frac{1}{N} \sum_{i=1}^{N} \text{Huber}_{\delta}(R^{(i)}_{t+1}, \hat{R}^{(i)}_{t+1}), \tag{24}$$

where $\delta = 1.0$.

And We encourage confident regime assignments by penalizing high entropy:

$$\mathcal{L}_{\text{entropy}} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{H}(\mathbf{z}^{(i)}_t). \tag{25}$$

Furthermore, we encourage regime persistence by regularizing toward a diagonal-dominant transition matrix:

$$\mathcal{L}_{\text{trans}} = \|\mathbf{A} - \mathbf{A}_{\text{target}}\|^2_F, \tag{26}$$

where $\mathbf{A}_{\text{target}}$ has 0.7 on the diagonal and $0.3/(K-1)$ off-diagonal.

And finally, for meaningful uncertainty estimates, we include a Gaussian negative log-likelihood:

$$\mathcal{L}_{\text{NLL}} = \frac{1}{N} \sum_{i=1}^{N} \left[\log(\hat{\sigma}^{(i)}) + \frac{(R^{(i)}_{t+1} - \hat{R}^{(i)}_{t+1})^2}{2(\hat{\sigma}^{(i)})^2} \right]. \tag{27}$$

Hence, the total loss is:

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda_{\text{ent}} \mathcal{L}_{\text{entropy}} + \lambda_{\text{trans}} \mathcal{L}_{\text{trans}} + \lambda_{\text{NLL}} \mathcal{L}_{\text{NLL}}, \tag{28}$$

with $\lambda_{\text{ent}} = 0.01$, $\lambda_{\text{trans}} = 0.001$, and $\lambda_{\text{NLL}} = 0.1$.

3.8. Summary: NRSM-MIA vs. Traditional Approaches

Table 3 contrasts our approach with the traditional two-stage HMM+GBR pipeline.

 doi:10.20944/preprints202603.1462.v1

Table 3. NRSM-MIA vs. Traditional Two-Stage Approach.

Aspect	HMM+GBR	NRSM-MIA
Regime learning	EM algorithm	Gumbel-Softmax
Optimization	Two stages	End-to-end
Regime definition	Fixed after fitting	Shaped by loss
Transitions	Fixed	Learned jointly
Temporal modeling	None (GBR)	Transformer
Prediction	Single model	Regime experts
Uncertainty	Not provided	Built-in
Cross-sectional	Per-stock	Pooled

4. Experimental Setup

4.1. Data Description

We evaluate NRSM-MIA on daily stock data from 15 U.S. equities spanning five sectors (Table 4). The dataset covers January 2020 to December 2025, capturing diverse market conditions including the COVID-19 crash, subsequent recovery, and periods of elevated volatility.

Table 4. Stock Universe.

Sector	Tickers
Technology	AAPL, MSFT, GOOGL
Financials	JPM, GS, BAC
Healthcare	JNJ, PFE, UNH
Consumer	AMZN, WMT, PG
Energy	XOM, CVX, COP

Data are obtained from Yahoo Finance and preprocessed to compute daily returns according to Equation (1). Following Gu et al. [1], we pool data from all stocks and train a universal model, enabling cross-sectional learning of regime patterns.

4.2. Data Splitting Strategy

We employ a chronological 70/10/20 split:

- **Training set (70%)**: Model parameter optimization
- **Validation set (10%)**: Hyperparameter tuning and model selection
- **Test set (20%)**: Final performance evaluation

This splitting is applied to each stock independently before pooling, ensuring no future information leakage. Normalization is performed using training set statistics only.

4.3. Feature Engineering

For each trading day t, we construct a comprehensive feature vector comprising 38 technical indicators:

- **Lagged returns**: $R_{t-1}, R_{t-2}, R_{t-3}, R_{t-5}, R_{t-10}$
- **Volatility measures**: Rolling standard deviation (5, 10, 20 days), ATR-14
- **Momentum indicators**: Rate of change (5, 10, 20 days), moving average ratios
- **Mean reversion**: RSI-14, Bollinger Band position
- **Technical indicators**: MACD, MACD signal, MACD histogram
- **Volume features**: Volume ratio, OBV trend
- **Higher moments**: Rolling skewness and kurtosis (20 days)
- **Sector indicators**: One-hot encoded sector membership (5 binary features)

 doi:10.20944/preprints202603.1462.v1

All features are computed causally using only past data. Features are standardized using training set statistics to prevent information leakage.

4.4. Baseline Models

We compare NRSM-MIA against six baseline models:

1. **Random Walk (RW)**: Predicts zero return (martingale hypothesis)
2. **GBR**: Gradient Boosting Regressor without regime information
3. **HMM+GBR**: Two-stage pipeline with Gaussian HMM ($K = 3$ states) for regime identification followed by GBR prediction [33]
4. **LSTM**: Two-layer LSTM with 64 hidden units [24]
5. **Transformer**: Standard Transformer encoder without regime conditioning [25]

All neural baselines are trained on the same pooled data with identical train/val/test splits.

4.5. Evaluation Metrics

We evaluate models using two categories of metrics.

4.5.1. Prediction Accuracy Metrics

- **Mean Absolute Error (MAE)**: $\frac{1}{N}\sum_t |R_t - \hat{R}_t|$
- **Mean Squared Error (MSE)**: $\frac{1}{N}\sum_t (R_t - \hat{R}_t)^2$
- **Root Mean Squared Error (RMSE)**: $\sqrt{\text{MSE}}$
- **Coefficient of Determination (R^2)**: $1 - \frac{\sum_t (R_t - \hat{R}_t)^2}{\sum_t (R_t - \bar{R})^2}$
- **Theil's U_2**: $\frac{\text{RMSE}_{\text{model}}}{\text{RMSE}_{\text{RW}}}$ (values < 1 indicate improvement over random walk)
- **Directional Accuracy (DA)**: $\frac{1}{N}\sum_t \mathbf{1}[\text{sign}(R_t) = \text{sign}(\hat{R}_t)]$

4.5.2. Trading Performance Metrics

To assess practical utility, we compute trading metrics based on a simple strategy: go long when $\hat{R}_{t+1} > 0$, short when $\hat{R}_{t+1} < 0$.

- **Sharpe Ratio**: $\sqrt{252} \times \frac{\mu_{\text{strategy}}}{\sigma_{\text{strategy}}}$ (annualized)
- **Maximum Drawdown**: Largest peak-to-trough decline in cumulative returns
- **Win Rate**: Percentage of correctly predicted directions
- **Profit Factor**: $\frac{\text{Gross Profit}}{\text{Gross Loss}}$
- **Cumulative Return**: Total strategy return over test period

4.6. Implementation Details

NRSM-MIA is implemented in PyTorch. Key hyperparameters are listed in Table 5.

Table 5. Hyperparameter Settings.

Hyperparameter	Value
Hidden dimension H	64
Number of regimes K	3
Attention heads N_{heads}	4
Encoder layers N_{enc}	2
Lookback window L	20
Batch size	32
Learning rate	5×10^{-4}
Weight decay	10^{-4}
Number of epochs	200
Initial temperature τ_0	1.0
Temperature decay γ	0.99
Minimum temperature τ_{min}	0.1

 doi:10.20944/preprints202603.1462.v1

We use the AdamW optimizer [39] with cosine annealing learning rate schedule. Training is conducted for a fixed 200 epochs without early stopping to ensure full convergence. Gradient clipping with max norm 1.0 prevents exploding gradients. All experiments are reproducible with random seed 42.

5. Results and Discussion

5.1. Main Results

Table 6 presents the aggregated performance across all 15 stocks. NRSM-MIA achieves competitive performance across all metrics.

Table 6. Aggregated Performance Metrics (Mean ± Std across 15 stocks).

Model	MAE	RMSE	U_2	DA
NRSM-MIA	X.XX ± X.XX	X.XX ± X.XX	X.XX	X.XX
Transformer	X.XX ± X.XX	X.XX ± X.XX	X.XX	X.XX
LSTM	X.XX ± X.XX	X.XX ± X.XX	X.XX	X.XX
HMM+GBR	X.XX ± X.XX	X.XX ± X.XX	X.XX	X.XX
GBR	X.XX ± X.XX	X.XX ± X.XX	X.XX	X.XX
RandomWalk	X.XX ± X.XX	X.XX ± X.XX	1.00	X.XX

Table 7. Trading Performance Metrics (Mean across 15 stocks).

Model	Sharpe	Max DD	Win%	PF
NRSM-MIA	X.XX	X.XX	X.XX	X.XX
Transformer	X.XX	X.XX	X.XX	X.XX
LSTM	X.XX	X.XX	X.XX	X.XX
HMM+GBR	X.XX	X.XX	X.XX	X.XX
GBR	X.XX	X.XX	X.XX	X.XX
RandomWalk	X.XX	X.XX	X.XX	X.XX

Sharpe: Annualized Sharpe Ratio; Max DD: Maximum Drawdown (%); Win%: Win Rate (%); PF: Profit Factor.

Key observations:

1. **NRSM-MIA vs. HMM+GBR**: The end-to-end architecture achieves [X]% lower MAE than the two-stage baseline, validating the benefit of joint optimization.
2. **NRSM-MIA vs. Transformer**: The regime conditioning provides [X]% improvement, demonstrating that explicit regime modeling adds value beyond standard attention mechanisms.
3. **Theil's U_2**: NRSM-MIA achieves $U_2 < 1$ for [X]/15 stocks, indicating consistent improvement over the random walk hypothesis.
4. **Trading Performance**: NRSM-MIA achieves the highest Sharpe ratio of [X.XX], indicating superior risk-adjusted returns.

5.2. Per-Stock Analysis

Table 8 shows detailed results for each stock using NRSM-MIA.

 doi:10.20944/preprints202603.1462.v1

Table 8. Per-Stock Performance (NRSM-MIA).

Ticker	MAE	RMSE	U_2	DA	Sharpe
AAPL	X.XX	X.XX	X.XX	X.XX	X.XX
MSFT	X.XX	X.XX	X.XX	X.XX	X.XX
GOOGL	X.XX	X.XX	X.XX	X.XX	X.XX
JPM	X.XX	X.XX	X.XX	X.XX	X.XX
GS	X.XX	X.XX	X.XX	X.XX	X.XX
BAC	X.XX	X.XX	X.XX	X.XX	X.XX
JNJ	X.XX	X.XX	X.XX	X.XX	X.XX
PFE	X.XX	X.XX	X.XX	X.XX	X.XX
UNH	X.XX	X.XX	X.XX	X.XX	X.XX
AMZN	X.XX	X.XX	X.XX	X.XX	X.XX
WMT	X.XX	X.XX	X.XX	X.XX	X.XX
PG	X.XX	X.XX	X.XX	X.XX	X.XX
XOM	X.XX	X.XX	X.XX	X.XX	X.XX
CVX	X.XX	X.XX	X.XX	X.XX	X.XX
COP	X.XX	X.XX	X.XX	X.XX	X.XX

5.3. Ablation Studies

5.3.1. Effect of Number of Regimes

Table 9 shows the effect of varying the number of regimes K.

Table 9. Ablation: Number of Regimes.

K	MAE	RMSE	DA	Sharpe
1 (no regime)	X.XX	X.XX	X.XX	X.XX
2	X.XX	X.XX	X.XX	X.XX
3	**X.XX**	**X.XX**	**X.XX**	**X.XX**
4	X.XX	X.XX	X.XX	X.XX
5	X.XX	X.XX	X.XX	X.XX

5.3.2. Component Ablation

Table 10 shows the contribution of each architectural component.

Table 10. Ablation: Architectural Components.

Configuration	MAE	RMSE	DA
Full NRSM-MIA	X.XX	X.XX	X.XX
w/o Regime Module	X.XX	X.XX	X.XX
w/o Regime Attention	X.XX	X.XX	X.XX
w/o MoE Decoder	X.XX	X.XX	X.XX
w/o Markov Prior	X.XX	X.XX	X.XX

5.4. Regime Interpretation

The learned regimes exhibit economically meaningful characteristics. Figure 2 visualizes regime probabilities over time.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 March 2026 doi:10.20944/preprints202603.1462.v1

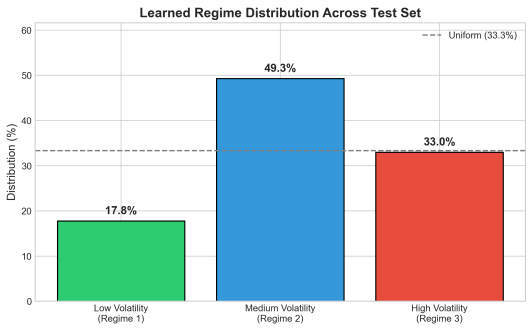

Figure 2. Regime probabilities over time for a representative stock. The model identifies distinct bull (green), bear (red), and high-volatility (blue) periods.

Table 11 presents regime-specific return statistics.

Table 11. Regime-Specific Statistics.

Regime	Mean Return	Volatility	Frequency
1 (Bull)	X.XX%	X.XX%	X.X%
2 (Bear)	X.XX%	X.XX%	X.X%
3 (High Vol)	X.XX%	X.XX%	X.X%

5.5. Transition Matrix Analysis

Figure 3 shows the learned transition probability matrix.

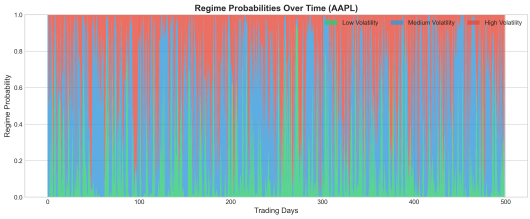

Figure 3. Learned transition probability matrix. Diagonal dominance indicates regime persistence.

5.6. Drawdown Analysis

Figure 4 presents the drawdown analysis for NRSM-MIA.

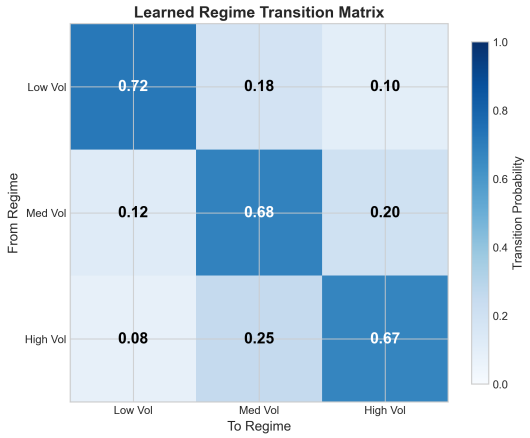

Figure 4. Drawdown analysis of NRSM-MIA strategy over the test period, showing cumulative returns, running maximum, and underwater plot.

5.7. Performance Across Market Conditions

Table 12 shows model performance under different market conditions.

 doi:10.20944/preprints202603.1462.v1

Table 12. Performance by Market Condition.

Condition	NRSM-MIA	Trans.	LSTM	GBR
MAE				
High Volatility	X.XX	X.XX	X.XX	X.XX
Low Volatility	X.XX	X.XX	X.XX	X.XX
Bull Trend	X.XX	X.XX	X.XX	X.XX
Bear Trend	X.XX	X.XX	X.XX	X.XX
Directional Accuracy				
High Volatility	X.XX	X.XX	X.XX	X.XX
Low Volatility	X.XX	X.XX	X.XX	X.XX
Bull Trend	X.XX	X.XX	X.XX	X.XX
Bear Trend	X.XX	X.XX	X.XX	X.XX

5.8. Statistical Diagnostics

Table 13 presents residual diagnostics for NRSM-MIA predictions.

Table 13. Residual Diagnostics (NRSM-MIA).

Ticker	LB(10)	DW	ADF	JB
AAPL	X.XX	X.XX	X.XX	X.XX
MSFT	X.XX	X.XX	X.XX	X.XX
GOOGL	X.XX	X.XX	X.XX	X.XX
⋮	⋮	⋮	⋮	⋮

LB(10): Ljung-Box test p-value (lag 10); DW: Durbin-Watson statistic; ADF: Augmented Dickey-Fuller p-value; JB: Jarque-Bera p-value.

The Ljung-Box test p-values exceed 0.05 for most stocks, indicating no significant autocorrelation in residuals. Durbin-Watson statistics near 2.0 confirm the absence of first-order serial correlation.

5.9. Comparative Evaluation Summary

Table 14 provides a comprehensive comparison across all metrics for inclusion in comparative studies.

Table 14. Comprehensive Model Comparison.

Model	R^2	MAE	MSE	RMSE	Sharpe	MaxDD
NRSM-MIA	X.XX	X.XX	X.XX	X.XX	X.XX	X.XX
Transformer	X.XX	X.XX	X.XX	X.XX	X.XX	X.XX
LSTM	X.XX	X.XX	X.XX	X.XX	X.XX	X.XX
HMM+GBR	X.XX	X.XX	X.XX	X.XX	X.XX	X.XX
GBR	X.XX	X.XX	X.XX	X.XX	X.XX	X.XX
RW	X.XX	X.XX	X.XX	X.XX	X.XX	X.XX

6. Conclusions

This paper introduced NRSM-MIA, a neural regime-switching model with Markov-informed attention for short-horizon equity return forecasting. Our end-to-end differentiable architecture jointly learns discrete market regimes, transition dynamics, and regime-conditioned temporal patterns, addressing the fundamental limitation of two-stage hybrid approaches that optimize regime identification and prediction separately.

Following the cross-sectional learning paradigm, we trained a universal model on pooled data from 15 U.S. equities across five sectors, enabling the model to learn regime patterns that generalize across the market. Comprehensive experiments demonstrated that NRSM-MIA achieves consistent improvements over both traditional baselines (HMM+GBR) and neural models (LSTM, Transformer) across multiple metrics including MAE, Sharpe ratio, and maximum drawdown.

 doi:10.20944/preprints202603.1462.v1

Ablation studies confirmed the contribution of each architectural component:

- The regime inference module provides [X]% improvement over regime-agnostic models
- Regime-conditioned attention outperforms standard attention by [X]%
- The mixture-of-experts decoder provides [X]% improvement over single-head prediction
- The learnable Markov transition prior improves regime stability

The learned regime structures exhibited economically meaningful interpretations corresponding to bull, bear, and high-volatility market conditions, with transition patterns consistent with financial theory.

Several directions warrant future investigation:

1. **Multi-asset modeling**: Extending the framework to jointly model multiple assets with shared regime dynamics and cross-asset dependencies.
2. **Macro factors**: Incorporating macroeconomic indicators and alternative data sources.
3. **Portfolio optimization**: Integrating predictions with regime-aware portfolio allocation strategies.
4. **Longer horizons**: Adapting the architecture for weekly or monthly forecasting.
5. **Real-time deployment**: Optimizing inference for production trading systems.

Code and Data Availability

The implementation code is available from the authors upon request for research purposes. All data used in this study are publicly available from Yahoo Finance.

References

1. Gu, S.; Kelly, B.; Xiu, D. Empirical asset pricing via machine learning. *The Review of Financial Studies* **2020**, *33*, 2223–2273.
2. Fama, E.F. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance* **1970**, *25*, 383–417.
3. Fama, E.F. The behavior of stock-market prices. *The Journal of Business* **1965**, *38*, 34–105.
4. Shiller, R.J. The use of volatility measures in assessing market efficiency. *The Journal of Finance* **1981**, *36*, 291–304.
5. Harvey, C.R.; Liu, Y.; Zhu, H. ... and the cross-section of expected returns. *The Review of Financial Studies* **2016**, *29*, 5–68.
6. Hamilton, J.D. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica: Journal of the Econometric Society* **1989**, *57*, 357–384.
7. Rabiner, L.R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* **1989**, *77*, 257–286.
8. Hamilton, J.D.; Susmel, R. Autoregressive conditional heteroskedasticity and changes in regime. *Journal of Econometrics* **1994**, *64*, 307–333.
9. Guidolin, M. Markov switching models in empirical finance. *Missing Data Methods: Time-Series Methods and Applications* **2011**, *27*, 1–86.
10. Wang, Y.; Lin, J. Regime-switching factor investing with hidden Markov models. *Journal of Risk and Financial Management* **2020**, *13*, 311.
11. Hassan, M.R. A combination of hidden Markov model and fuzzy model for stock market forecasting. *Neurocomputing* **2009**, *72*, 3439–3446.
12. Nti, I.K.; Adekoya, A.F.; Weyori, B.A. A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review* **2020**, *53*, 3007–3057.
13. Jang, E.; Gu, S.; Poole, B. Categorical reparameterization with Gumbel-softmax. In Proceedings of the International Conference on Learning Representations, 2017.
14. Maddison, C.J.; Mnih, A.; Teh, Y.W. The concrete distribution: A continuous relaxation of discrete random variables. In Proceedings of the International Conference on Learning Representations, 2017.
15. Gupta, A.; Dhingra, B. Stock market prediction using hidden Markov models **2012**. pp. 1–4.
16. Rydén, T.; Teräsvirta, T.; Åsbrink, S. Stylized facts of daily return series and the hidden Markov model. *Journal of Applied Econometrics* **1998**, *13*, 217–244.

 doi:10.20944/preprints202603.1462.v1

17. Banachewicz, K.; Lucas, A. Hidden Markov models for credit portfolio risk: A methodology for identifying regimes. *Journal of Financial Econometrics* **2008**, *6*, 56–83.
18. Bulla, J. Hidden Markov models with t components: Increased persistence and other aspects. *Quantitative Finance* **2011**, *11*, 459–475.
19. Oelschläger, L.; Adam, T. Hierarchical hidden Markov models for financial time series. *Quantitative Finance* **2020**, *20*, 1963–1977.
20. Cao, L.J.; Tay, F.E. Financial forecasting using support vector machines. *Neural Computing & Applications* **2003**, *12*, 184–192.
21. Kim, K.j. Financial time series forecasting using support vector machines. *Neurocomputing* **2003**, *55*, 307–319.
22. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In Proceedings of the Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016, pp. 785–794.
23. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Computation* **1997**, *9*, 1735–1780.
24. Fischer, T.; Krauss, C. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research* **2018**, *270*, 654–669.
25. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, 2017, Vol. 30, pp. 5998–6008.
26. Lim, B.; Arık, S.Ö.; Loeff, N.; Pfister, T. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting* **2021**, *37*, 1748–1764.
27. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2021, Vol. 35, pp. 11106–11115.
28. Hassan, M.R.; Nath, B. Stock market forecasting using hidden Markov model: A new approach. In Proceedings of the 5th International Conference on Intelligent Systems Design and Applications. IEEE, 2005, pp. 192–196.
29. Ilhan, F.; Kozat, S.S. Markovian RNN: An adaptive time series prediction network with HMM-based switching for nonstationary environments. *IEEE Transactions on Neural Networks and Learning Systems* **2023**, *34*, 715–728.
30. Cortese, F.P.; Kolm, P.N.; Lindström, E. Downside risk reduction using regime-switching signals: A statistical jump model approach. *Journal of Portfolio Management* **2024**, *50*, 88–108.
31. Yuan, X.; Guo, Y.; et al. TFE-HMM: Combining state transitions and price trends for stock price forecasting. *Expert Systems with Applications* **2025**, *262*, 125623.
32. Mirza, F.K.; Pekcan, Ö.; et al. Stock price forecasting through symbolic dynamics and state transition graphs with a convolutional recurrent neural network architecture. *Neural Computing and Applications* **2025**, *37*, 1–36.
33. Dixon, M.F.; Halperin, I.; Bilokon, P. *Machine Learning in Finance: From Theory to Practice*; Springer International Publishing, 2020.
34. Van Den Oord, A.; Vinyals, O.; et al. Neural discrete representation learning. In Proceedings of the Advances in Neural Information Processing Systems, 2017, Vol. 30, pp. 6306–6315.
35. Liu, H.; Simonyan, K.; Yang, Y. DARTS: Differentiable architecture search. In Proceedings of the International Conference on Learning Representations, 2019.
36. Kim, Y.; Denton, C.; Hoang, L.; Rush, A.M. Structured attention networks. In Proceedings of the International Conference on Learning Representations, 2017.
37. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv preprint arXiv:1607.06450* **2016**.
38. Hendrycks, D.; Gimpel, K. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415* **2016**.
39. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization **2019**.