
Learning Time-Inhomogeneous Markov Dynamics in Financial Time Series via Neural Parameterization

Jan Rovirosa¹ Jesse Schmolze¹

¹University of Wisconsin – Madison, Madison, WI, USA

Abstract

Modeling non-stationary dynamics in noisy, continuous sequential data requires balancing structural interpretability with representational flexibility. In domains such as finance, discretizing continuous signals into regime-based states is a common analytical technique; however, estimating transition probabilities through direct empirical tabulation often proves to be a degenerate exercise, yielding sparse and high-variance matrices due to limited sample support. In this paper, we address this challenge by framing the sequence as a feature-conditioned, time-inhomogeneous Markov chain. We parameterize the transition dynamics via a neural network that maps the current state and concurrent features to a probability vector on the simplex. This formulation transforms the learning problem from discrete counting to continuous conditional density estimation. We demonstrate that this neural-Markov parameterization acts as a regularized estimator, recovering dense, coherent transition structures from data where empirical matrices are otherwise sparse. Furthermore, through a controlled ablation study, we show that explicit conditioning on the current state serves as a critical inductive bias: it stabilizes the learning process and significantly improves temporal generalization compared to state-free baselines. The resulting framework enables the rigorous analysis of local regime dynamics and multi-step uncertainty propagation without sacrificing the flexibility of deep learning models.

the resulting matrices are sparse and high-variance due to the lack of repeated samples for rare transitions [2].

In this work, we propose that the first-order Markov assumption can be utilized as a structural inductive bias to solve this estimation problem. Rather than viewing the Markov property as a restriction, we use it to regularize the learning of time-varying dynamics. We model the sequence as a feature-conditioned, time-inhomogeneous Markov chain, where the transition operators are parameterized by a neural network.

By replacing discrete counting with neural parameterization, we transform the task into continuous conditional density estimation. This allows the model to "filter" market noise and recover dense, stable transition structures that remain hidden in raw empirical data.

Our contributions include the demonstration that the neural-Markov parameterization acts as a regularized estimator, recovering smooth, dense transition structures where empirical counting fails. Through a state-free ablation study, we also show that explicit conditioning on the Markov state stabilizes the learning process and improves temporal generalization, proving that the Markovian structure provides a helpful constraint in high-noise environments.

Crucially, this framework enables us to move beyond one-step heuristics and investigate the internal consistency of the learned dynamics through multi-step composition.

1 Introduction

Learning the dynamics of non-stationary systems is a central challenge in machine learning. A fundamental tension exists in modeling such data: classical stationary Markov models offer interpretable structural constraints but fail to adapt to regime shifts [1], while modern deep neural networks capture complex temporal dependencies but often do so through opaque latent representations that lack probabilistic clarity [11].

This tension is most acute in settings where the data is continuous, noisy, and heavy-tailed—a combination that renders traditional discrete estimation methods ineffective. Financial return series serve as the canonical example of this regime [9]. When continuous returns are discretized into a fine state space \mathcal{S} to study distributional shifts, estimating transition probabilities through empirical counting becomes a degenerate exercise;

2 Related Work

2.1 Neural-Markov Architectures

The intersection of neural networks and Markov processes has historically focused on generative tasks. For instance, Awiszus and Rosenhahn [7] introduced "Markov Chain Neural Networks" to simulate stochastic behaviors (e.g., random walks) within a deep learning framework. However, their work focuses on inducing stochasticity into deterministic models for simulation purposes. In contrast, our work focuses on the inverse problem: *estimation*. We utilize the neural network not to generate random noise, but to recover a structured, regularized transition operator from noisy, sparse empirical observations.

2.2 Time-Inhomogeneity in Finance

In quantitative finance, the limitations of homogeneous Markov chains are well-documented. Mettler et al. [8] explicitly model exchange rates as time-inhomogeneous chains, demonstrating that dynamic transition matrices better capture market volatility than static ones. However, such approaches typically rely on statistical counting or rolling-window estimation. As noted by Lando and Skodeberg [2], these discrete estimation methods degenerate in fine-grained state spaces due to data sparsity. Our work advances this lineage by replacing discrete statistical estimation with *continuous neural parameterization*, allowing for density estimation that remains robust even when empirical transition counts are sparse or zero.

2.3 Feature-Dependent Dynamics

While the majority of Markov literature assumes time-homogeneous transition probabilities, there is a distinct line of work focusing on feature-dependent chains. Boyd and Vandenberghe [3] propose a framework for fitting transition matrices where each row is a function of a feature vector. However, their approach relies on convex optimization and logistic regressors to guarantee convergence. While mathematically elegant, this restricts the model to linear relationships between covariates and transition dynamics.

Similarly, early work on Input-Output HMMs by Bengio and Frasconi [5] established the potential for conditioning state transitions on input sequences. Our work revisits these foundational ideas through the lens of modern deep learning. We demonstrate that replacing linear or shallow parameterizations with deep neural networks is not merely an increase in complexity, but a necessary step to capture the highly non-linear interactions between macroeconomic indicators and market regimes [11].

2.4 Sequence Modeling and Density Estimation

Standard neural sequence models (e.g., LSTMs, Transformers) typically optimize for point-prediction accuracy using Mean Squared Error (MSE). As noted by Bishop [4] in the context of Mixture Density Networks (MDNs), point estimates are insufficient for processes with multi-modal or stochastic outputs.

In the domain of probabilistic time series forecasting, recent approaches like DeepAR [6] address this by predicting the parameters of a fixed distribution (e.g., Gaussian or Negative Binomial) at each time step. While effective for stationary data, these parametric assumptions can be limiting in finance, where return distributions are notoriously heavy-tailed and skewed [9].

Our approach differs by combining the flexibility of MDNs with the structure of discretized states. By outputting a probability vector over a finite state space \mathcal{S} , our model performs *categorical density estimation*. This avoids imposing a specific

parametric shape (like a Gaussian) on the returns, allowing the model to learn arbitrary distributional shapes (including fat tails and multi-modalities) purely from the data, similar to the distributional perspective in reinforcement learning [10].

3 Problem Formulation

3.1 Return Process and State Space

Let $\{P_t\}_{t \geq 0}$ denote the sequence of adjusted daily closing prices for a given equity. We define the daily percentage return r_t as:

$$r_t = \frac{P_t - P_{t-1}}{P_{t-1}} \quad (1)$$

To be able to form a discrete Markovian framework, the continuous return r_t is mapped to a discrete state X_t through a discretization function $D : \mathbb{R} \rightarrow \mathcal{S}$. The state space $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$ consists of n separate bins representing ranges of returns.

3.2 Time-Dependent Markov Assumption

We model the evolution of the discretized return states $\{X_t\}$ using a feature-conditioned, time-inhomogeneous Markov framework [8]. Unlike a stationary Markov chain [1], where the transition probabilities are constant, we assume that the transition dynamics are governed by a sequence of time-varying transition matrices $\{A_t\}_{t \geq 0}$, where $A_t \in [0, 1]^{n \times n}$ and $\sum_j a_{ij}(t) = 1$ for all i, t .

We assume that the transition dynamics are governed by a set of observable features $F_t \in \mathbb{R}^d$, which include macroeconomic, market and fundamental indicators available at time t , similar to the covariate-dependent framework of Boyd and Vandenberghe [3]. The probability of transitioning from state s_i at time t to state s_j at time $t + 1$ is defined as:

$$a_{ij}(t) = \mathbb{P}(X_{t+1} = s_j \mid X_t = s_i, F_t) \quad (2)$$

This formulation assumes that while the process remains Markovian (conditioned on the current state and current exogenous information), the underlying transition dynamics are non-stationary and evolve over time.

3.3 Learning Objective

The objective is to approximate the time-varying transition dynamics of the return process. Specifically, we seek to identify a mapping $f : \mathcal{S} \times \mathbb{R}^d \rightarrow \Delta^{n-1}$ that estimates the conditional distribution of the next state X_{t+1} , given the current state X_t and the exogenous feature vector F_t .

Formally, let $\mathbf{a}_i(t)$ denote the i -th row of the transition matrix A_t , representing the true (latent) conditional probabilities for a transition from state s_i at time t . The goal is to find an optimal parameterization θ such that the model output $\hat{\mathbf{p}}_t = f_\theta(X_t, F_t)$ serves as an estimator of the transition vector

$\mathbf{a}_{X_t, \cdot}(t)$. Within this framework, the learning task is framed as a conditional density estimation problem [4]. Rather than producing a deterministic point forecast, the objective is to recover the full probabilistic structure of the transition, thereby enabling the analysis of time-varying stochastic structure in the return dynamics.

4 Methods and Models

4.1 Data Sources and Feature Construction

All data is gathered from FactSet and FRED, and then grouped into three categories: (i) equity price history for the target asset; (ii) macroeconomic and market indicators, specifically the Federal Funds Rate, High-Yield (HY) bond indices, and Investment-Grade (IG) bond indices; and (iii) firm-level fundamentals, comprising Cash Flow, Ratios, Income Statements, Balance Sheet data, and miscellaneous key items. The daily price history provides the temporal index. Lower-frequency macro and fundamental features are aligned to the daily grid by (a) mapping each trading day to calendar `year` and fiscal `quarter` identifiers and (b) carrying forward the most recently observed value within the same calendar year.

Missing values are handled by a restricted imputation scheme designed to preserve interpretability. Within each calendar year, we apply linear interpolation for short gaps and last-observation-carried-forward/backward (LOCF) for boundary gaps. Features with gaps too large to perform interpolation on, are removed from the dataset.

4.2 State Discretization

We use the discretization scheme introduced in Section 3 to map each daily return to a state $X_t \in \mathcal{S}$. In our implementation we set the number of bins to $n = 55$ to balance granularity and per-class sample support. The learning target at time t is the next-day state label $Y_t := X_{t+1}$.

4.3 Neural Parameterization of Transition Rows

We model the time-varying transition row associated with the realized current state through a conditional classifier. Let $F_t \in \mathbb{R}^d$ denote the aligned feature vector available at time t , and let $e(X_t) \in \{0, 1\}^n$ denote the one-hot encoding of the current state. The model input is the concatenation

$$z_t := [e(X_t); F_t] \in \mathbb{R}^{n+d}. \quad (3)$$

A multilayer perceptron (MLP) with parameters θ maps z_t to logits $g_\theta(z_t) \in \mathbb{R}^n$, which are converted to a probability vector via softmax:

$$\hat{\mathbf{p}}_t = \text{softmax}(g_\theta(z_t)) \in \Delta^{n-1}. \quad (4)$$

By construction, $\hat{\mathbf{p}}_t$ estimates the row $\mathbf{a}_{X_t, \cdot}(t)$ of the latent transition matrix A_t .

4.4 Learning Objective and Optimization

Given a dataset of chronologically ordered observations $\{(X_t, F_t, X_{t+1})\}_{t=1}^{T-1}$, we fit θ by minimizing the negative log-likelihood of the realized next state:

$$\mathcal{L}(\theta) = - \sum_{t=1}^{T-1} \log \hat{\mathbf{p}}_t[X_{t+1}], \quad (5)$$

which is equivalent to multi-class cross-entropy.

4.5 Ablation Baseline (State-Free Model)

To isolate the contribution of explicit state conditioning, we also consider an ablated baseline that removes $e(X_t)$ from the input and predicts $\hat{\mathbf{p}}_t = \text{softmax}(g_\theta(F_t))$. This baseline retains the same output interpretation (a distribution over next-day states) but does not admit a direct interpretation as a Markov transition row.

5 Experiments

5.1 Datasets

We evaluate the proposed framework on a single-equity case study (JPMorgan Chase, \$JPM), which allows us to analyze learned transition dynamics in a controlled setting. Following the preprocessing described in Section 3 and Section 4, we construct two datasets in order to study the trade-off between sample size and feature richness: (i) a higher-sample dataset of size $4,181 \times 134$ (trading days \times features) and (ii) a higher-feature dataset of size $2,369 \times 198$. In both cases, the label at time t is the next-day state $Y_t = X_{t+1}$.

5.2 Compared Models

All experiments compare two instantiations of the same base architecture:

- **State-conditioned transition model.** Input $z_t = [e(X_t); F_t]$ and output $\hat{\mathbf{p}}_t \in \Delta^{n-1}$ interpreted as the estimated transition row $\mathbf{a}_{X_t, \cdot}(t)$.
- **State-free ablation baseline.** Input F_t only, with the same output space and loss.

This ablation isolates the role of explicit state conditioning in inducing an interpretable Markov-operator view of the dynamics.

5.3 Architecture

Both models use a fully-connected MLP with five hidden layers of widths

$$64 \rightarrow 128 \rightarrow 256 \rightarrow 128 \rightarrow 64, \quad (6)$$

GELU nonlinearities, and dropout with probability $p = 0.2$ after each hidden layer. The final layer is linear in \mathbb{R}^n followed by a softmax.

5.4 Chronological Splits and Training Procedure

To avoid look-ahead effects, we use a chronological split into training, validation, and test sets. Optimization is performed with Adam at learning rate $\eta = 0.001$ using cross-entropy loss. We select the final model by evaluating performance on the validation set after each epoch and retaining the epoch with the lowest validation loss.

5.5 Evaluation Focus: Transition Structure

Since the purpose of the framework is to recover stable and interpretable Markovian transition dynamics [1], we emphasize diagnostics of the learned transition structure. Specifically, we analyze:

- **Generalization via loss dynamics:** training vs. validation cross-entropy trajectories.
- **Regularity of the transition dynamics:** comparing an empirical transition matrix computed from held-out data to the model-implied transition matrices. This directly addresses the sparsity issues inherent in empirical estimation [2].
- **Local error geometry:** an error severity statistic defined as the absolute distance between predicted and realized bins, which treats near-miss predictions differently from distant misses.
- **Representation structure (optional):** qualitative analysis of learned embeddings (e.g., UMAP on hidden representations) to assess whether state conditioning changes the internal geometry.

6 Results

6.1 Generalization and Optimization Stability

Across both datasets, the state-conditioned model exhibits smoother generalization behavior than the state-free baseline when tracked through training and validation cross-entropy. Concretely, the baseline tends to reduce training loss while its validation loss increases earlier, consistent with learning dependencies that do not transfer forward in time. In contrast, conditioning on X_t constrains the hypothesis class to a row-selected transition operator, which empirically stabilizes training.

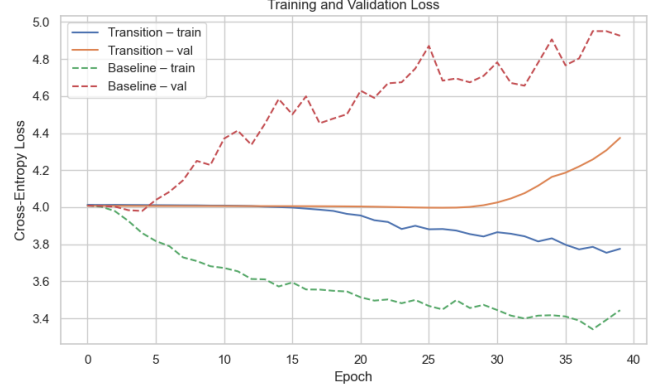


Figure 1: Training and validation cross-entropy loss trajectories for the state-conditioned transition model and the state-free baseline. Both models exhibit decreasing training loss, indicating adequate fit to the training data. However, the state-free baseline shows an early and sustained divergence between training and validation loss, consistent with unstable generalization over time. In contrast, explicit state conditioning yields more stable validation behavior, suggesting that the induced transition-dynamics structure acts as a regularizing constraint.

6.2 Empirical vs. Model-Implied Transition Operators

A central diagnostic is the structure of the induced transition matrix. From a held-out segment we compute the empirical transition matrix

$$\hat{A}_{ij}^{\text{emp}} := \frac{\#\{t : X_t = s_i, X_{t+1} = s_j\}}{\#\{t : X_t = s_i\}}, \quad (7)$$

whenever the denominator is nonzero. With $n = 55$ and a limited number of observations per state, \hat{A}^{emp} is sparse and high-variance, illustrating the degeneracy of discrete estimation in fine-grained state spaces [2].

In contrast, the model produces, for each time index t , a dense row estimate $\hat{\mathbf{p}}_t$ that assigns mass to neighboring bins rather than concentrating on a small set of realized transitions. Aggregating these rows over the evaluation period yields a smoother average transition structure. This behavior is consistent with using the neural network as a regularized estimator of time-varying transition dynamics, rather than a direct tabulation of realized transitions. By smoothing out finite-sample noise, the model provides a conservative estimate of distributional risk, preventing the overfitting common in rolling-window estimators that react too sharply to idiosyncratic shocks.

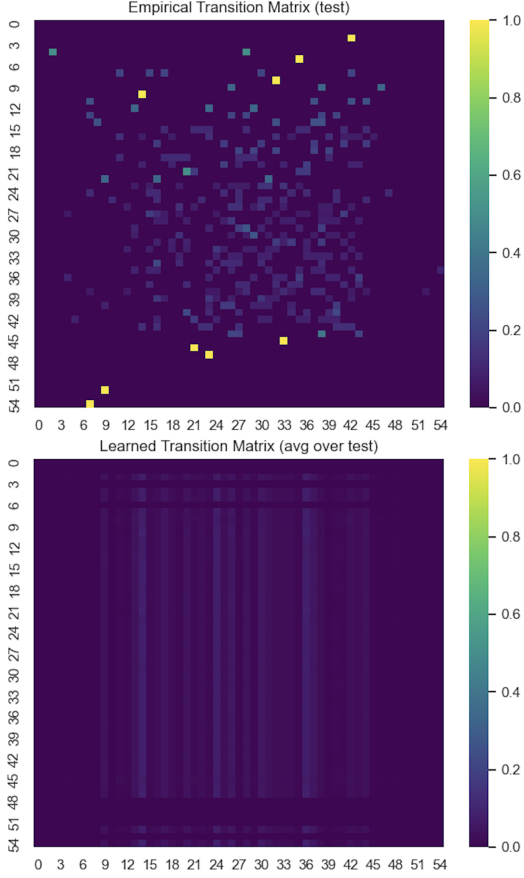


Figure 2: Comparison between the empirical transition matrix computed from held-out test data (left) and the model-implied transition matrix averaged over the same period (right). Due to the fine discretization and limited number of observations per state, the empirical estimator is sparse and high-variance. In contrast, the learned transition dynamics produce a dense and smoother structure, assigning probability mass across neighboring states. This behavior is consistent with the role of the neural model as a regularized estimator of time-varying transition dynamics rather than a direct tabulation of realized transitions.

6.3 Local Error Geometry (Severity)

Exact classification accuracy is a limited statistic when the label space is a fine discretization of a continuous variable. To capture the geometry induced by the discretization, we report an *error severity* statistic

$$\text{sev}(t) := \left| \arg \max_j \hat{\mathbf{p}}_t(j) - X_{t+1} \right|, \quad (8)$$

which counts the number of bins between the predicted mode and the realized next state. Both models concentrate most probability mass on moderate-to-large severities, reflecting the intrinsic noise and heavy tails of daily returns [9], but the

transition model shifts mass slightly toward smaller severities relative to the state-free baseline.

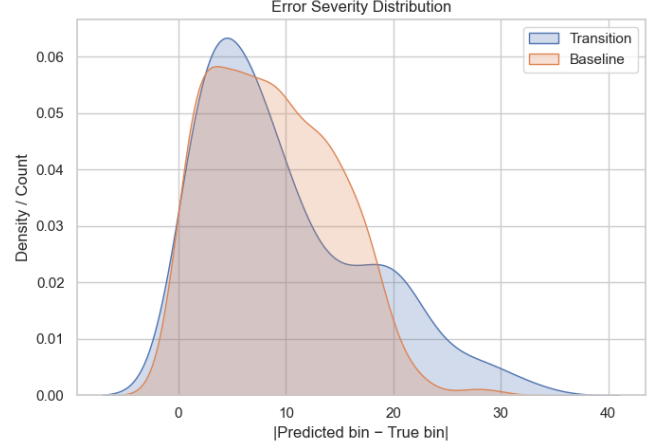


Figure 3: Distribution of the error severity statistic, defined as the absolute distance between the predicted modal bin and the realized next-state bin. Both models exhibit broadly similar severity profiles, reflecting the intrinsic noise and fine discretization of daily returns. The transition model places slightly more mass on smaller severities than the state-free baseline.

6.4 Representation Analysis

We also examined the geometry of learned hidden representations using a low-dimensional projection (UMAP) of the penultimate layer activations. Both models learn structured embeddings rather than an unstructured scatter, suggesting meaningful internal representations of the feature dynamics. The state-conditioned model exhibits slightly smoother embedding structure, consistent with an inductive bias toward state-dependent regimes.

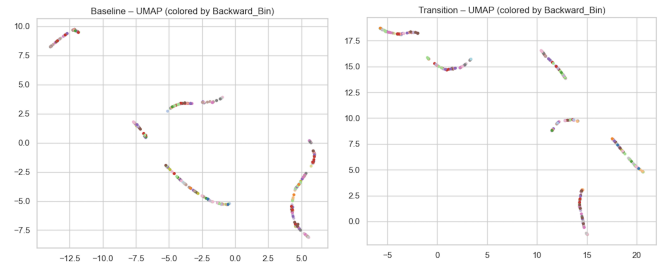


Figure 4: UMAP projections of the penultimate-layer representations for the state-conditioned transition model (left) and the state-free baseline (right), colored by the realized backward return bin. Both models exhibit structured embeddings rather than an unstructured scatter, reflecting the learning of nontrivial internal representations. Differences between the two projections are subtle, consistent with the shared architecture and largely overlapping feature inputs.

6.5 Summary of Findings

Overall, the experiments support the central claim of the paper: explicit state conditioning yields a more stable view of next-step transition dynamics. Given the fine discretization and noise in daily returns, the main difference between models is in the transition structure: the model-implied transition matrix is less sparse and smoother than the empirical transition matrix computed directly from data.

7 Discussion

7.1 Dynamics-Based Representation

The primary output of the proposed framework is a sequence of conditional distributions $\{\hat{\mathbf{p}}_t\}_t$ over discretized return states. These distributions can be interpreted as rows of a time-varying transition matrix. This representation differs from point prediction in that it preserves uncertainty and allows the evolution of probability mass to be examined directly [4].

An advantage of this formulation is that multi-step distributions can be obtained by composing successive transition matrices. Starting from an initial state distribution π_t , longer-horizon distributions follow as $\pi_{t+k} = \pi_t A_t A_{t+1} \cdots A_{t+k-1}$. This provides a natural way to study how uncertainty accumulates over time without requiring additional modeling assumptions [1].

7.2 Smoothness as Regularized Estimation

The empirical transition matrix obtained by direct tabulation is a high-variance estimator when the discretization is fine and the number of observations per state is limited [2]. In this regime, the smoothness observed in the learned transition dynamics should be interpreted as a form of regularized probability estimation rather than as a claim about the underlying process.

In practice, the model distributes probability mass across nearby bins instead of concentrating it on a small number of realized transitions. This behavior reduces sensitivity to individual observations and yields a more stable estimate of conditional transition behavior.

7.3 Stability Under Temporal Generalization

The reduced divergence between training and validation loss for the state-conditioned model indicates more stable temporal generalization. Here, stability refers to the consistency of the learned conditional distributions when evaluated on later time periods.

Conditioning explicitly on the current return state allows the model to estimate a local transition law for each state. This reduces the need for the network to encode state-dependent behavior implicitly in the feature representation, which appears to support generalization under chronological evaluation.

7.4 Interpreting Transition Structure

The learned transition dynamics admit interpretation at different levels of aggregation. Individual rows correspond to conditional distributions over next-day return states given the current state and features. Aggregating these rows over time yields average transition structures that summarize how probability mass is distributed across regimes.

Such summaries can be used to examine properties such as persistence of states, asymmetry between positive and negative returns, and the dispersion of transition probabilities [9]. These interpretations follow directly from the probabilistic structure of the model output and do not rely on additional assumptions.

7.5 Evaluation Scope

Consistent with the modeling objective, evaluation focuses primarily on the structure and stability of the learned transition dynamics. Diagnostics such as error severity are used to assess local behavior of the conditional distributions. The emphasis on interpretability reflects the intended use of the framework as a modeling tool for studying time-varying transition dynamics.

8 Conclusion

We presented a feature-conditioned, time-inhomogeneous Markov framework for modeling discretized return dynamics. The method outputs a sequence of simplex-valued transition rows $\{\hat{\mathbf{p}}_t\}_t$ that define time-varying transition dynamics on the state space.

This output represents the local evolution of probability mass across return regimes at each time step. It supports multi-step distributional propagation by composing the time-varying transition matrices.

Across our experiments, explicit state conditioning yields transition structures that are substantially more regular than empirical tabulations formed from held-out data. We observe that the state-conditioned model produces dense transition rows that allocate mass across neighboring bins, whereas the empirical estimator is sparse under fine discretization [2]. Chronological evaluation further shows improved training-validation stability for the state-conditioned parameterization.

The learned transition dynamics can be inspected row-by-row to study regime persistence, asymmetry, and dispersion conditional on (X_t, F_t) [9]. They can be aggregated over time windows to form average transition matrices that summarize how the dynamics change over time. They can also be composed to obtain horizon-dependent state distributions and to track how uncertainty evolves under the learned dynamics.

A natural extension of this framework is to examine the temporal consistency of the learned transition dynamics. In particular, the Chapman–Kolmogorov equation provides a way to compare multi-step transitions obtained by composing

one-step dynamics with direct estimates at longer horizons [1]. Such comparisons can help assess whether discrepancies arise from genuine time variation in the dynamics or from finite-sample estimation effects.

Taken together, this work illustrates how neural parameterizations can be used to model time-varying Markovian transition dynamics in an explicit and interpretable way.

References

- [1] Hamilton, J. D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica*, 57(2), 357–384.
- [2] Lando, D., & Skodeberg, T. M. (2002). Analyzing rating transitions and rating drift with continuous observations. *Journal of Banking & Finance*, 26(2), 423–444.
- [3] Boyd, S., & Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.
- [4] Bishop, C. M. (1994). Mixture density networks. Technical Report NCRG/94/004, Neural Computing Research Group, Aston University.
- [5] Bengio, Y., & Frasconi, P. (1995). An input output HMM architecture. *Advances in Neural Information Processing Systems*, 7, 427–434.
- [6] Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3), 1181–1191.
- [7] Awiszus, M., & Rosenhahn, B. (2018). Markov chain neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.
- [8] Mettle, F. O., Boateng, L. P., Quaye, E. N. B., Aidoo, E. K., & Seidu, I. (2022). Analysis of exchange rates as time-inhomogeneous Markov chain with finite states. *Journal of Probability and Statistics*, 2022.
- [9] Cont, R. (2001). Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 1(2), 223–236.
- [10] Bellemare, M. G., Dabney, W., & Munos, R. (2017). A distributional perspective on reinforcement learning. *International Conference on Machine Learning (ICML)*, 449–458.
- [11] Lim, B., & Zohren, S. (2021). Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194).