A Deep Learning Framework for Time-Dependent Markov Chain Modeling of Equity Prices

Jesse Schmolze [1] **Albert Lee** [1] **Frederik Pedersen** [1] **Jan Rovirosa Illa** [1]

Abstract

Markov chains provide a simple framework for modeling the evolution of stochastic processes, but their formulation assumes fixed transition probabilities that do not adapt to changing conditions. In financial markets, this assumption is often violated, as asset return dynamics depend on time-varying market factors. In this project, we propose a deep learning framework for modeling next-day equity return dynamics through a time-dependent Markov chain. Daily returns of JPMorgan Chase ($JPM) are discretized into a set of states, and a multilayer perceptron is trained as a multi-class classifier to estimate conditional transition probabilities given the current return state and observable fundamental and macroeconomic features. The resulting model produces a probability distribution over future return states rather than a single point forecast, allowing uncertainty to be explicitly quantified. We compare the learned transition structure to an empirical Markov chain constructed directly from historical data and analyze the resulting transition matrices and simulated price paths. While the model is not intended for short-horizon trading, it provides an interpretable probabilistic framework for understanding short-term return dynamics, risk, and regime behavior in equity markets.

1. Introduction

Over the past two decades, the quantitative finance industry has embraced deep learning to decompose noise into actionable patterns across virtually every asset class. In this paper, we take a distinct approach on this premise by attempting to construct a discrete-space, time-inhomogeneous Markov chain to model the day-to-day price evolution of JPMorgan Chase ($JPM) using deep learning.

A Markov chain is a stochastic process $\{X_k\}$ satisfying the Markov property

$$\mathbb{P}(X_k = x_k \mid X_{k-1}, X_{k-2}, \ldots, X_0) = \mathbb{P}(X_k = x_k \mid X_{k-1}).$$

In other words, it is a sequence of random variables used to represent an evolving random system under the assumption that future states depend only on the current state. If we denote the probability of transitioning from state i to state j in one time step by a_{ij}, we can form a transition matrix A whose rows each sum to one. Concretely, the matrix takes the form

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}, \qquad \sum_{j=1}^{n} a_{ij} = 1 \;\; \forall i.$$

Each row of A represents the conditional probability distribution of tomorrow's return state given today's return state. Visually, one may interpret a single row as a probability mass function over possible next-day outcomes, while the full matrix encodes how these distributions change across different current return regimes. In our setting, each state corresponds to the daily percentage change in $JPM, with the time period between transitions being one trading day. Importantly, once the transition matrix of a Markov chain is known, the full dynamics of the process are determined, provided the initial distribution is given.

Formally, let $\{r_t\}_{t\geq 0}$ denote the sequence of daily returns of $JPM, where

$$r_t = \frac{P_t - P_{t-1}}{P_{t-1}},$$

and P_t denotes the adjusted closing price on trading day t. We discretize the continuous return space into a finite set of n bins,

$$\mathcal{S} = \{s_1, s_2, \ldots, s_n\},$$

and define the Markov state variable X_t by assigning each realized return r_t to the corresponding bin in $\mathcal{S}$.

[1]University of Wisconsin–Madison, Madison, WI, USA. Correspondence to: Jesse Schmolze <jschmolze@wisc.edu>, Albert Lee <aslee32@wisc.edu>, Frederik Pedersen <flpedersen@wisc.edu>, Jan Rovirosa Illa <rovirosa@wisc.edu>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

More generally, the assumption of stationarity can be relaxed by allowing the transition matrix to vary with time. A time-inhomogeneous Markov chain is defined by a sequence of transition matrices $\{A_t\}_{t\geq 0}$, where

$$\mathbb{P}(X_{t+1} = j \mid X_t = i) = a_{ij}(t), \qquad \sum_{j=1}^{n} a_{ij}(t) = 1 \;\; \forall i, t.$$

Under this formulation, the evolution of the state distribution $\pi_t \in \mathbb{R}^n$ is governed by

$$\pi_{t+1} = \pi_t A_t,$$

where $\pi_t(i) = \mathbb{P}(X_t = i)$. Unlike the stationary case, closed-form characterizations of long-run behavior generally do not exist, and the dynamics depend explicitly on the temporal sequence $\{A_t\}$. This formulation is particularly well suited to financial markets because structural changes and evolving information invalidate the assumption of time-invariant transition probabilities.

If we let a financial asset's value over time be the sequence of random variables we are modeling, knowledge of its transition matrix would yield strong predictive power. Not only could one compute expected future prices using the expectation operator, but one could also deploy portfolio optimization techniques more effectively if such matrices were available across a wide range of assets. Numerous studies, however, have shown that the Markov property fails in empirical financial data (see, e.g., (Cecen et al.)). This implies that the price of a financial asset tomorrow depends on more than just its price today, which is intuitively unsurprising. For example, the trend in U.S. interest rate movements is often just as important as the level of rates themselves. See (Guyon & Lekeufack) for an empirical illustration of the failure of Markovian dependence in financial markets.

While financial markets exhibit no pure Markovian dependence, we argue that this failure can be reconciled through a time-dependent transition matrix whose structure evolves as new information arrives. The key insight is that price movements do not violate the spirit of Markovian behavior, but rather the assumption of a *stationary* transition matrix. Financial markets evolve as indicators change, their importance shifts over time, and new indicators continuously emerge. At its core, however, the act of buying a stock reflects the belief: "given the price today, the stock will go up or down tomorrow." This is the essence of the Markov property. Consequently, the task of the quantitative investor is to filter out exogenous information from this basic decision. This is precisely where our project derives its value.

The central idea is that, without deep learning, one effectively models

$$\mathbb{P}(x_{t+1} \mid x_t, \mathcal{F}),$$

where x_{t+1} denotes the price of \$JPM at time $t+1$, x_t its price at time t, and $\mathcal{F}$ is a vector of relevant information drawn from both the present and the past. In this project, we aim to reduce the dimensionality of this conditional probability to

$$\mathbb{P}(x_{t+1} \mid x_t)$$

by encoding the information contained in $\mathcal{F}$ directly into a time-evolving transition matrix. In practice, this would involve updating the transition matrix daily as new data arrive; however, due to time constraints, we do not perform forward testing in this study. Deep learning enables this temporal evolution to occur flexibly and adaptively, filtering noise and capturing nonlinear dependencies that are often missed by human-designed models.

We emphasize that the objective of this work is not to propose a trading strategy or to maximize predictive accuracy. Rather, our goal is to study whether a time-inhomogeneous Markov structure can be recovered and successfully interpreted when dynamics are allowed to evolve with flexible incoming information. This perspective motivates several modeling choices, such as discretizing returns and prioritizing the creation of a probabilistic structure instead of direct price prediction. These decisions change both the scope of the analysis and the way our model should be evaluated. With that being said, the contributions of this paper lie in the interpretability and structure of the learned transition dynamics.

An important advantage of the transition matrix representation is that it enables the computation of multi-step return distributions. Given an initial distribution π_t over states at time t, the distribution at horizon $t+k$ is given by

$$\pi_{t+k} = \pi_t A_t A_{t+1} \cdots A_{t+k-1}.$$

This recursive structure provides a mechanism for measuring uncertainty forward in time, allowing one to analyze a wide variety of useful modeling scenarios over arbitrary time horizons. These quantities are generally inaccessible when relying solely on point forecasts of future prices.

While direct price prediction is a natural extension of this framework, a transition matrix provides substantially richer information. By modeling the full conditional distribution over next-day return states, the framework explicitly represents uncertainty, tail risk and asymmetries in tomorrow's return dynamics. This representation is well suited to tools from stochastic calculus, stochastic control and stochastic optimization. We anticipate that the portfolio manager can use our transition dynamics as inputs to their optimization and risk management procedures. We discuss these potential applications and their limitations in the discussion section.

2. Related Work

Our work draws inspiration from the theoretical framework of *Markov Chain Neural Networks* proposed by Awiszus and Resenhahn, who demonstrate architectural methods for blending probabilistic Markov structures with neural networks. Although we do not directly implement their design in this project, we draw heavily from their core insight: that incorporating an explicit probabilistic component can significantly enhance the interpretability and structure of neural network outputs (Awiszus & Rosenhahn).

A closely related application appears in the use of hidden, time-inhomogeneous Markov chains for predicting mortality outcomes in life insurance modeling (Kiermayer & Weiß). In that setting, the authors reconstruct latent transition structures using stochastic gradient descent. A key distinction between their work and ours is that they aim to uncover endogenous effects in stochastic systems *after* they have occurred. In contrast, our objective is to use deep learning to predict future transition probability distributions directly. This task is inherently noisier, as it is impossible to incorporate all exogenous information shocks occurring between the present and future into the model.

Guyon and Julien address a closely related challenge by using machine learning techniques to quantify the extent to which volatility is endogenous to a financial asset (Guyon & Lekeufack). Their results confirm the intuition that recovering latent volatility structures is more reliable than predicting future volatility, although the latter remains feasible despite increased variability in accuracy. Our work aligns with the latter objective, seeking to forecast future probabilistic dynamics rather than retrospectively identify latent structures.

Wilinski's research on time-inhomogeneous Markov chains for financial forecasting further informs our approach, though his models rely on algorithmic updates with fixed functional bounds (Wilinski). We circumvent this limitation by employing deep learning, allowing for flexible, data-driven evolution of the transition matrix. While Wilinski's framework achieves robust results, we anticipate that our approach is better suited to handling the high levels of noise present in financial data. Moreover, the explicit use of transition matrices in our model provides additional utility for portfolio-level applications (see the Discussion section).

Taken together, this body of work suggests a tradeoff between flexibility, interpretability and predictive power in stochastic modeling. Approaches that seek to identify latent probabilities benefit from reduced noise in the system, while predictive models are forced to confront higher variance due to exogenous shocks to the dynamic system. Our methodology occupies a middle ground. Rather than inferring transition probabilities retrospectively or imposing static functional updates, we learn a time-varying matrix that is both probabilistic and explicit. This design provides easier interpretability on the state-to-state level while allowing the transition dynamics to evolve as new information gets incorporated into the system. This niche we occupy is particularly well suited to financial applications.

We close by analyzing a related study on exchange rates assuming randomly selected transition matrices drawn from an ergodic set (Mettle et al.). Their approaches can motivate alternative designs that relax the strict Markov assumption; they generally sacrifice interpretability and computational efficiency. In contrast, constructing an explicit transition matrix enables direct and efficient implementation in large-scale simulations and forward-testing frameworks, offering substantial advantages when evaluating millions of portfolio configurations or conducting extensive Monte Carlo analyses.

3. Data Acquisition and Preprocessing

All datasets used in this study were acquired from FactSet. We categorized the retrieved data into three primary groups: *macro*, *fundamental*, and *equity price* data. Macro-level data included high-yield and investment-grade bond rates, as well as the federal funds rate. These variables served as background information, capturing large-scale, aggregated signals that reflect broader trends in the U.S. economy. This layer of data provided the model with context regarding national economic conditions.

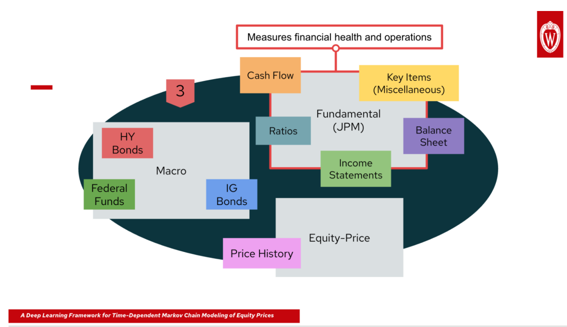

Figure 1. This chart shows the different categories of data we used to train our model. All data is sourced from FactSet.

Fundamental data served a similar purpose at a narrower scope, focusing specifically on JPMorgan Chase. These datasets, reported primarily on a quarterly basis, provided detailed information on JPM's financial health and operational performance. As a result, fundamental variables constituted the majority of our feature set. Finally, the equity-price dataset contained day-to-day information on JPM's stock price, including daily returns and short-term price trends, and represented the primary object of interest in our analysis.

Although the necessary datasets were available through FactSet, a central challenge was feature sparsity. In an earlier experiment aimed at rapidly validating the feasibility of our modeling framework, we adopted a naive preprocessing strategy that pruned all features containing missing values. As expected, this approach resulted in substantial data loss, particularly among JPM's fundamental variables and the high-yield and investment-grade bond series. For the final set of experiments, recovering this lost information was essential to unlocking the full potential of the revised model.

To address missing values, we synthesized data using a combination of linear interpolation and last observation carried forward/backward (LOCF) extrapolation. The appeal of this approach lies in its simplicity and its reliance on real-world observations, ensuring that synthesized values remain intrinsically tied to observed data. However, to preserve data fidelity, we imposed strict limitations on the scope of interpolation and extrapolation. In particular, we did not interpolate or extrapolate across calendar years. For example, a Q1 2023 value was not inferred from a Q4 2022 observation, as these periods correspond to distinct macroeconomic and firm-specific contexts. To maintain internal consistency and avoid obscuring latent relationships between data categories, we pruned any features with gaps too large to support reliable interpolation or extrapolation.

Feature selection was guided not only by availability but also by what indicators offered interpretability. We retained variables only if they had a clear economic interpretation and could realistically influence the dynamics of the next day's price. FactSet provides a wide range of fundamental, technical, and derived indicators, but many were excluded to avoid introducing data points that could complicate the interpretation of the learned transition dynamics. This design choice reflects our emphasis on structured probabilistic modeling rather than on maximizing our predictive performance.

Once synthetic values were generated within each dataset, the next step was to merge them into a unified training set. This posed a nontrivial challenge, as the datasets were recorded at different temporal frequencies: daily (equity prices), quarterly (fundamentals), and yearly (certain macro indicators). We used the daily price history dataset as the temporal baseline. While quarterly and yearly values could be reasonably extrapolated to a daily frequency, the reverse operation—aggregating daily prices or volumes into quarterly or yearly summaries—lacked a clear economic interpretation.

To facilitate merging, we engineered two additional features from the price history data: `quarter` and `year`. These variables mapped each trading day to its corresponding fiscal quarter and calendar year, allowing the remaining datasets to be joined on a common temporal index. In practice, this required reconstructing precise calendar dates, as the raw price history data recorded dates in a partially masked format (e.g., //). To resolve this issue, we employed the `bizdays` package in R, which provides robust tools for business-day calculations and calendar conversions.

Using these reconstructed date references, we aligned the macroeconomic and fundamental datasets to the daily price baseline, extrapolating quarterly and yearly values as needed to form the final feature matrix X. The final preprocessing step involved constructing features corresponding to the transition matrix bins. For an arbitrary trading day D, the *backward bin* was defined as the percentage change from day $D-1$ to D, while the *forward bin* corresponded to the percentage change from day D to $D+1$. Since the first and last observations lacked one of these quantities, they were removed from the dataset. The forward bin was subsequently excluded from the feature matrix and instead used to construct the label vector Y.

This initial data construction process produced a training set of dimension $4{,}181 \times 134$. A key limitation of this dataset was the relatively small number of features, a consequence of prioritizing sample size. Although FactSet provided additional variables, many were excluded due to shorter historical coverage. To study the trade-off between data quantity and feature richness, we constructed a second dataset that maximized the number of features, yielding a training set of dimension $2{,}369 \times 198$. This allowed us to evaluate whether a larger number of training samples with fewer features outperformed a smaller dataset with greater feature dimensionality.

4. Model Comparison and Experimental Results

We evaluate two models on a diagnostic dataset: a *transition-based neural network* and an *ablated baseline model* in which the current state is removed from the input vector. Both models share the same multilayer perceptron (MLP) architecture in terms of depth, hidden-layer widths, activation functions, and dropout rates. The only architectural difference is whether information about the current state is included in the input.

4.1. Transition Model Architecture

The transition model takes as input the full feature vector concatenated with a one-hot encoding of the current state. This combined input is passed through a fully connected neural network with five hidden layers of sizes

$$64 \to 128 \to 256 \to 128 \to 64.$$

Each hidden layer uses the GELU activation function and dropout with probability $p = 0.2$. The final hidden representation is passed through a linear output layer that produces scores for each return bin. These scores are converted into probabilities via a softmax layer and trained using cross-entropy loss.

Because the model conditions explicitly on the current state, its output varies with the state and can be interpreted as a row of a time-varying transition matrix.

4.2. Baseline Model Architecture

The baseline model uses the same MLP architecture, including the same hidden-layer sizes, GELU activations, and dropout probability of 0.2. The only difference is that its input consists solely of the feature vector, with no information about the current state included.

4.3. Training Procedure

Both models are trained using the same chronological split into training, validation, and testing datasets to ensure a fair comparison. The Adam optimizer is used with a learning rate of 10^{-3}, and cross-entropy loss is employed as the training objective. Model selection is performed by monitoring the validation loss over training epochs and selecting the model with the lowest validation loss. The selected models are then evaluated on the held-out test set.

Figure 2. This graph shows the training and validation cross-entropy loss for both models over the training epochs. For both models, the training loss decreases steadily over the epochs, indicating that the models are able to fit the data. However, the baseline without information about the current state starts overfitting very early. While its training loss is decreasing, its validation loss starts growing early and keeps increasing steadily. This suggests that the baseline model learns patterns that do not generalize well to unseen data.

4.4. Error Severity and Predictive Performance

To assess predictive accuracy beyond exact classification, we examine the *error severity distribution*, defined as the absolute difference between the predicted return bin and the ground-truth bin. Smaller values correspond to predictions closer to the realized return.

Both models exhibit broadly similar error severity distributions, reflecting the inherent difficulty of predicting finely discretized financial returns in a noisy environment. However, the transition model places slightly more probability mass on smaller errors compared to the baseline model. This suggests that conditioning predictions on the current state provides a modest improvement in predictive accuracy and stability.

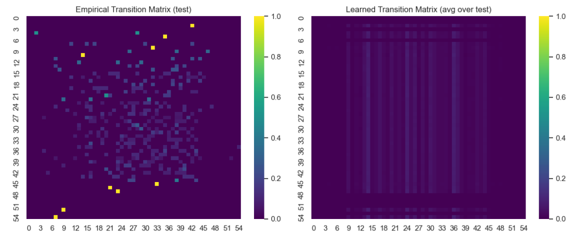

Figure 3. This heatmap compares the empirical transition matrix computed directly from the testing data with the transition matrix estimated by the transition model which is averaged over the test set. The empirical matrix is quite sparse and noisy which is in line with the limited number of observations for the large number of possible state transitions and the inherent randomness of the daily returns. In contrast to this, the transition matrix estimated by the model is much smoother. Instead of memorizing the specific transitions present in data, the model assigns probability mass across nearby states. This produces a smoother structure, reducing noise and better capturing general patterns instead of fitting random movement present in data.

4.5. Representation Analysis via UMAP

UMAP projections of the learned hidden representations for both models are broadly similar, which is expected given that the architectures and input features are nearly identical. In both cases, the embeddings form structured clusters rather than a random scattering of points, indicating that each network learns a meaningful low-dimensional representation of the data.

The transition model exhibits a slightly smoother embedding structure, although the difference is subtle. This suggests that state information influences the learned representations in a nuanced way, without fundamentally altering the organization of the feature space.

4.6. Generalization and Transition Dynamics

While both models are capable of fitting the training data, the baseline model begins to overfit relatively quickly. In contrast, the transition model generalizes more smoothly across epochs. Additionally, the learned transition matrix from the transition model displays a smoother and more regular structure than the empirical transition matrix, which is consistent with the limited and noisy nature of the available data.

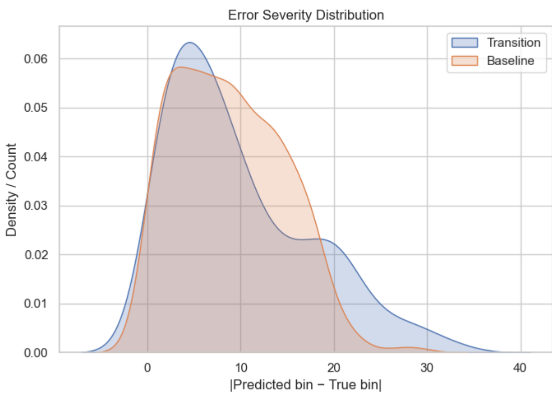

Figure 4. The error severity distribution shows the absolute difference between the predicted return bin and the ground truth, where smaller values correspond to predictions closer to the actual return. Both models show an overall similar distribution in errors, which reflects the difficulty in predicting narrow states in a noisy financial setting. However, the transition model does place slightly more mass towards the smaller errors compared to the baseline model. This suggests that conditioning the predictions on the current state does help somewhat with making more accurate and stable.

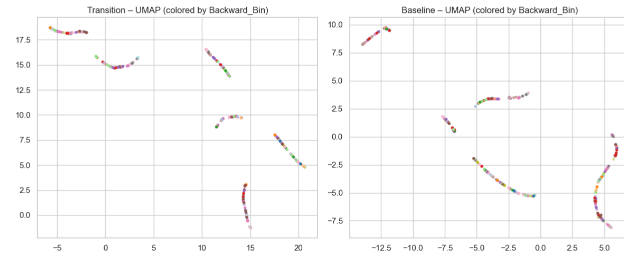

Figure 5. The UMAP projections for the two models are broadly similar, which reflects that both models share the same architecture and are trained on almost the same input features. For both models, the embedding forms some structured clusters rather than a random scattering of points. This indicates that each network learns a low-dimensional internal representation of the data. While the transition model does have a slightly smoother structure, it is very subtle, suggesting that the state information influences the learned representations in a subtle way that does not fundamentally change how the feature space is organized.

4.7. Test Set Results

On the test set of 356 observations, the transition model achieves an accuracy of

$$3.1\% \pm 0.9\%,$$

compared to

$$2.5\% \pm 0.8\%$$

for the baseline model. Given the very fine-grained discretization into 55 return bins and the intrinsic noise of financial data, these low accuracy levels are expected. Small fluctuations in returns can easily move observations across adjacent bins, making exact classification difficult. Random chance accuracy under uniform guessing is approximately $\frac{1}{55} = 1.8\%$, which places both models meaningfully above chance despite the noisy setting.

The uncertainty associated with the accuracy estimates is comparable to the observed difference between the two models, indicating that the improvement in accuracy is not statistically significant. Consequently, accuracy alone is not a reliable metric of model quality in this setting.

4.8. Summary

Overall, the experimental results suggest that conditioning the model on the current state leads to modest improvements in training stability, error severity, and the smoothness of learned transition dynamics. While the gains in raw accuracy are small and not statistically significant, the transition model appears to produce more robust and stable predictions, supporting the value of incorporating state information when modeling return transitions.

5. Discussion

5.1. Results and Broader Impact

The results in this project should be interpreted through the lens of interpretable probabilistic modeling rather than perfect point prediction. The main objective of the project was not to maximize next-day return accuracy, but to build a stable and interpretable conditional distribution over future return states. Given the level of noise in the financial markets, it is expected that modeling precision will be somewhat low, specifically if we are to discretize returns into bins. Therefore, the typical accuracy metrics are not sufficient to evaluate the effectiveness of the model.

Instead, the value of the model lies in the structure of the learned transition probabilities. By producing a full probability distribution over next-day returns, the model conveys substantially more information than a single-point forecast. This probabilistic output enables richer downstream analysis, particularly in risk management and portfolio construction. A straightforward application would be to compute the expected next-day price of a single stock using standard statistical methods; however, restricting the model to this use case underutilizes its potential. The primary value of the framework lies in the interpretability and flexibility afforded by an explicit transition matrix.

When such matrices are available across multiple assets, they can be combined with tools from stochastic calculus and stochastic control to inform more sophisticated portfolio management strategies. For instance, a portfolio manager could leverage asset-specific transition matrices to design hedging strategies that better account for exogenous volatility. Implementing such an approach would require estimating transition matrices for a broad universe of equities and potentially tailoring aspects of the model to investor-specific objectives. Nevertheless, the framework naturally extends to a wide range of applications, from long-term portfolio optimization to broader risk management settings.

In isolation, computing a transition matrix for a single asset offers limited advantages over traditional deep learning prediction methods. However, when aggregated across assets and analyzed jointly, transition matrices unlock substantially greater analytical power. While pursuing these extensions lies beyond the technical and mathematical scope of the present work, they represent promising directions for future research. These considerations naturally motivate a discussion of the framework's limitations, which are outlined in the following section.

5.2. Limitations

The following limitations arise from deliberately made choices to balance stability, data constraints, and interpretability, prioritizing the latter.

Although the model produces next-day probability distributions, it is not intended to function as a trading strategy. While such an application lies outside the scope and objectives of this project, a natural extension would be to use the learned transition probabilities to forecast daily price movements for $JPM and generate trading signals. For example, if the return space were reduced to two states—positive and negative—the model could suggest taking a long position when the probability of a positive return exceeds 50% and a short position otherwise. Despite its intuitive appeal, such a strategy would not be economically tradable in practice.

The primary obstacle is that the probability distributions evolve daily, requiring frequent position changes. This would lead to extremely high turnover, and the associated transaction costs—including bid–ask spread, slippage, and commissions—would exceed the expected daily returns, which are typically on the order of 5–10 basis points. Even with a perfectly accurate signal, such a strategy would be unprofitable once trading costs are taken into account. A second practical limitation is that the model cannot capture all sources of randomness in equity prices. A nontrivial portion of daily price movements arises from noise that is independent of the model's input features. In some cases, this noise may dominate the small predictive edge identified by the model, rendering the signal insufficient for generating alpha.

Beyond these practical constraints, there are also structural limitations inherent to the framework. One such limitation stems from discretizing continuous daily returns into a finite set of bins. While this step simplifies the problem and enables a classification-based approach, it necessarily removes information. Returns that are numerically close may fall into different bins, while returns that differ by several basis points may be grouped together. Consequently, the model cannot distinguish between movements within the same bin, reducing predictive precision. Decreasing bin widths mitigates this issue only up to a point: sufficiently small bins result in very few observations per class, leading to a large softmax output and unstable cross-entropy optimization. In the limit, the problem approaches a regression setting, which would no longer support probability distributions or a Markov transition structure. Thus, while discretization is essential for classification, it imposes a fundamental constraint on accuracy.

Another limitation arises from the nonstationary nature of financial markets. Relationships between input features and future returns can shift as macroeconomic conditions, monetary policy, or firm-specific factors evolve. Training a single neural network over a long historical window implicitly averages across multiple market regimes, potentially obscuring regime-specific patterns. A possible extension would involve explicitly modeling regime changes, either by training separate models for distinct market environments or by constructing an adaptive framework that adjusts as conditions evolve.

A further limitation arises from the conditional Markov assumption underlying the framework. At each point in time, the model conditions on the current state and the current information. This disregards the path dependence exhibited in many financial indicators. To circumvent this moving forward, we plan to introduce synthetic variables in the model to capture trends in the data beyond the information we currently have that is constrained to a single day. We anticipate that this will increase the accuracy of the model and allow for the neural network architecture to find more nuanced patterns in the data.

The final limitation concerns temporal mismatches in the data. Most market variables are observed daily, whereas accounting variables—such as balance sheets, income statements, and cash flow statements—are reported quarterly. To incorporate these variables, we interpolate them across trading days, creating a smooth daily progression that does not reflect the true timing of fundamental information releases. Although interpolation improves predictive performance, it weakens the interpretability of the input features and may reduce the model's ability to anticipate abrupt shifts in market behavior driven by discrete information shocks.

6. Conclusion

In this work, we introduced a deep learning framework for modeling equity returns using a time-dependent Markov chain with a discrete state space. We discretized the daily returns and trained a neural network to estimate conditional transition probabilities given both the current return state and observable market information. In doing so, we effectively shifted the model objective away from point prediction towards a more holistic view of $JPM's probabilistic dynamics. This approach emphasizes the interpretability of Markov chains, while relaxing the static assumption of them that limits their applicability in modern-day quantitative finance.

Our empirical results show that explicit conditioning on the current state leads to smoother transition matrices, improved stability and modest gains in the predictive power relative to a baseline model that ignores state information. While exact classification accuracy remains low, this is an expected outcome given the noise of daily equity returns. Regardless, the learned transition matrix structure exhibits meaningful regularities that are absent in empirical transition matrices constructed from data. These findings suggest that deep learning can be used as an effective mechanism for filtering out noise and building interpretable probabilistic models.

In this paper, we demonstrate that modeling equity prices through a learned, time-varying transition matrix provides a alternative to traditional deep learning forecasts. This representation naturally supports applications in risk analysis, simulation and portfolio construction, especially when extended to multiple assets. Several limitations remain, but the framework still offers a foundation for future use.

Potential extensions include adding regime dynamics, refining the discretization schemes and scaling the approach to mixed-asset settings. Generally speaking, this work illustrates how classic stochastic modeling can be successfully integrated with modern machine learning techniques to better capture probabilistic structures.

References

Awiszus, M. and Rosenhahn, B. Markov chain neural networks. URL http://arxiv.org/abs/1805.00784.

Cecen, A. A., Jain, P., and Xiao, L. Machines, memory and the markov property in stock returns: Deus ex machina? ISSN 1556-5068. doi: 10.2139/ssrn.2963418. URL https://www.ssrn.com/abstract=2963418.

Guyon, J. and Lekeufack, J. VOLATILITY IS (MOSTLY) PATH-DEPENDENT.

Kiermayer, M. and Weiß, C. Neural calibration of hidden inhomogeneous markov chains: information decompression in life insurance. 113(10):7129–7156. ISSN 0885-6125, 1573-0565. doi: 10.1007/s10994-024-06551-w. URL https://link.springer.com/10.1007/s10994-024-06551-w.

Mettle, F. O., Boateng, L. P., Quaye, E. N. B., Aidoo, E. K., and Seidu, I. Analysis of exchange rates as time-inhomogeneous markov chain with finite states. 2022:1–13. ISSN 1687-0042, 1110-757X. doi: 10.1155/2022/3524808. URL https://www.hindawi.com/journals/jam/2022/3524808/.

Wilinski, A. Time series modeling and forecasting based on a markov chain with changing transition matrices. 133: 163–172. ISSN 09574174. doi: 10.1016/j.eswa.2019.04.067. URL https://linkinghub.elsevier.com/retrieve/pii/S0957417419303033.