Term Structure Modelling and Forecasting with Neural Structured State-Space Models

Pedro Amaral Amorim Oliveira and Márcio Poletti Laurini
FEARP - University of São Paulo

March 19, 2026

Abstract

This paper studies the yield curve as a machine learning problem and proposes the use of neural structured State Space Models (SSMs), implemented through the Mamba architecture, as a flexible alternative to traditional parametric and non parametric term structure models. Rather than imposing fixed functional forms for factor loadings and linear Gaussian dynamics, the SSM approach learns latent states, transition dynamics, and cross-maturity interactions directly from the data. This allows the model to adapt to nonlinearities, regime dependence, and time-varying relationships that may be difficult to capture with pre-specified parametric structures. To incorporate economic structure, we augment the architecture with a B-spline layer that represents the cross-sectional shape of the yield curve and enables the imposition of smoothness and no-arbitrage restrictions. In particular, we introduce a specification that penalizes violations of monotonic discount factors and negative forward rates, ensuring that predicted term structures remain economically consistent. This framework embeds financial theory directly into the learning objective, in a manner conceptually similar to physics-informed neural networks, where structural constraints guide the estimation while preserving the flexibility of deep learning models. We evaluate the approach through Monte Carlo simulations and an empirical application to U.S. Treasury Constant Maturity yields from 2019 to 2025. Across a range of data-generating processes and maturities, the SSM specification delivers systematically lower out-of-sample forecast errors than the Diebold–Li baseline model, with statistically significant improvements in most cases. The gains are particularly pronounced when the underlying dynamics deviate from linear, time-invariant factor structures. Our findings suggest that structured neural state-space models provide a principled and powerful machine learning framework for modeling and forecasting the yield curve, bridging modern sequence modeling techniques with the econometric tradition of dynamic term structure analysis while allowing the incorporation of economically meaningful no-arbitrage constraints.

1 Introduction

The term structure of interest rates, commonly represented by the yield curve, describes the relationship between the yield of debt instruments and their time to maturity (Filipović, 2009). It is a fundamental concept in finance, reflecting the market's expectations of future interest rates, inflation, and economic conditions. Understanding the term structure is crucial for a wide range of applications, including pricing fixed-income securities, managing interest rate risk, valuing derivatives, and conducting monetary policy analysis. However, the yield curve is not static; it evolves over time in response to macroeconomic shocks, monetary interventions, and market sentiment (Diebold and Rudebusch, 2013). Consequently, capturing its dynamics requires stochastic models capable of describing both the level, slope, and curvature of the curve and their temporal dependencies. Dynamic models, such as affine term structure models (Duffie and Kan, 1996; Dai and Singleton, 2000; Duffee, 2002), Heath-Jarrow-Morton frameworks (Heath et al., 1992; Musiela and Rutkowski, 2005), and more recently, structured state space sequence models (Gu et al., 2022, 2020; Gu and Dao, 2024), provide the necessary tools to simulate, forecast, and analyze the evolution of yields across maturities, offering both theoretical consistency and practical predictive power.

In traditional approaches to modeling the term structure of interest rates, such as affine term structure models, Heath-Jarrow-Morton frameworks, and other parametric specifications, both the stochastic dynamics of the yield curve and the associated risk premia are typically assumed or chosen

a priori. These methods impose rigid functional forms on the level, slope, and curvature of the curve and the dynamics of forward rates, and often restrict the evolution of factors driving yields to linear or affine dynamics. While this allows for analytical tractability and ease of estimation, it also introduces strong parametric assumptions that may not adequately capture the richness and flexibility observed in real-world yield curves. As a result, these models may fail to fully accommodate nonlinearities, regime shifts, or time-varying volatility structures inherent in the term structure, potentially limiting their predictive performance and applicability in dynamic financial environments.

A first alternative to rigid parametric approaches is to employ more flexible statistical models for the term structure, such as the Nelson-Siegel (Nelson and Siegel, 1987) and the generalized Diebold-Li class of models (Diebold and Li, 2006; Christensen et al., 2011). These frameworks use and extend the classical three-factor specification to include additional latent factors, often incorporating four or five dynamic components, to better capture the level, slope, curvature, and higher-order movements of the yield curve. Other approaches use non-parametric estimations of the continuos time diffusions Stanton (1997), mixed parametric models (Almeida et al., 2017) or functional data analysis (Laurini, 2014; Kowal et al., 2019). From a mathematical perspective, these models can be interpreted as a non-parametric approximation of the term structure using Laguerre, B-Splines or similar orthogonal basis expansions, where the observed yields are expressed as a linear combination of basis functions weighted by time-varying latent factors. This representation allows for a more flexible and parsimonious characterization of the yield curve while maintaining a tractable, factor-driven dynamic structure suitable for forecasting and risk management applications.

Although these statistical factor models offer greater flexibility compared to fully parametric specifications, they still impose an *a priori* functional form on the term structure through the choice of basis functions and the number of latent factors. This assumption can become limiting in practice, particularly in periods of pronounced changes in risk premia or when the overall shape of the yield curve evolves in ways that are not well captured by the pre-specified level, slope, and curvature components. Consequently, even generalized Diebold-Li models may fail to fully adapt to abrupt shifts or nonlinearities in the dynamics of yields, highlighting the need for models capable of learning the term structure more flexibly from data without relying on rigid parametric forms.

To address these limitations, fully data-driven or structured state-space approaches offer an alternative by learning the dynamics of the yield curve directly from the data. Rather than imposing a fixed functional form or predetermined number of factors, these models can represent the term structure through latent states whose evolution is governed by flexible stochastic dynamics, potentially including non-linearities and time-varying volatility. Structured state space models (SSMs) (Gu et al., 2022, 2020; Gu and Dao, 2024), in particular, allow for the integration of known temporal dependencies while retaining the capacity to adapt to abrupt changes in risk premia and overall curve shape. By combining the interpretability of state-space representations with the expressive power of modern machine learning architectures, these models provide a framework capable of capturing complex yield curve dynamics that traditional parametric or semi-parametric factor models may miss.

Structured state-space models (SSMs) provide a principled framework to model time series with latent dynamics, separating the observed data from unobserved states that evolve over time according to stochastic processes. In the context of the term structure of interest rates, SSMs allow the yield curve to be expressed as a function of a low-dimensional latent state, capturing level, slope, and curvature factors, while the evolution of these factors is modeled through transition dynamics. Recent advances integrate neural networks into this framework (Krishnan et al., 2015; Rangapuram et al., 2018; Chen et al., 2019), using them to flexibly parameterize both the mapping from latent states to observed yields and the latent state transition itself . This approach enables the model to learn the intrinsic shape of the yield curve directly from data, without imposing rigid parametric forms, while retaining the interpretability and structure of a state-space representation. By combining SSMs with deep learning, one can capture non-linearities, time-varying risk premia, and complex interactions between maturities, providing a powerful tool for forecasting and scenario analysis in fixed income markets.

The Mamba framework (Gu and Dao, 2024; Dao and Gu, 2024) represents a recent advancement in structured state-space modeling, specifically designed to handle long sequences efficiently while capturing complex temporal dependencies. At its core, Mamba parameterizes a linear state-space model using (deep) neural networks, allowing for expressive and flexible transition and observation functions, while maintaining the computational advantages of SSMs. One of its main characteristics is

the use of fast parallelizable recurrence algorithms that scale linearly with the sequence length, enabling the modeling of long-term dependencies without incurring the quadratic cost typical of standard RNNs (Gu and Dao, 2024; Dao and Gu, 2024). These properties make Mamba particularly suitable for applications like term structure modeling, where sequences are long, non-linearities are important, and interpretability of the latent dynamics is essential.

In the context of modeling the term structure of interest rates, the Mamba framework provides a powerful tool to capture the dynamic evolution of yields without imposing rigid parametric forms. By learning the latent state-space structure directly from data, Mamba can model complex, non-linear interactions between maturities over time, including shifts in level, slope, and curvature. Its ability to incorporate neural network modules, such as LSTMs (Hochreiter and Schmidhuber, 1997), allows the model to adaptively encode temporal dependencies and stochastic volatility patterns, while the structured state-space representation ensures interpretability of the latent factors driving the yield curve. This flexibility makes Mamba particularly advantageous over traditional affine or Nelson-Siegel-type approaches, as it can account for time-varying risk premia and structural changes in the curve's shape, providing more accurate and robust out-of-sample forecasts for financial and risk management applications.

Compared to classical approaches such as the Diebold-Li or affine term structure models, Mamba offers a fundamentally different paradigm by not requiring the specification of factor loadings, risk premia, or the parametric form of the yield curve a priori. While Diebold-Li models impose a three-factor structure and affine models assume linear dynamics under risk-neutral measures, Mamba learns both the latent state evolution and the mapping to observed yields directly from data. This allows it to flexibly capture non-linear interactions between level, slope, and curvature factors, adapt to time-varying volatility, and respond to structural changes in risk premia. As a result, Mamba provides more accurate in-sample fits and more robust out-of-sample forecasts, particularly in environments where traditional parametric assumptions are violated or when sudden shifts in the shape of the yield curve occur.

Building on the standard Mamba framework (Gu and Dao, 2024; Dao and Gu, 2024), our paper also introduces several extensions that enhance its flexibility and forecasting performance for term structure modeling. First, we integrate a recurrent LSTM module before the structured state space, allowing the model to capture complex temporal dependencies and memory effects in yield evolution. Second, we incorporate B-spline layers at the output, enabling the network to approximate the yield curve as a smooth function over maturities, rather than relying solely on discrete factor loadings. Under this framework, we impose additional no-arbitrage constraints on SSM predictions through shape constraints on the discount and forward curves associated with the observed yield curve, thus combining the flexibility of neural network models with theoretical constraints derived from asset pricing theory. Third, we introduce an alternative explicit smoothness penalty in the training objective, which encourages the predicted yield curves to be smooth across maturities, reflecting realistic economic behavior. Together, these innovations allow Mamba to learn both the latent state space dynamics and the functional shape of the yield curve in a data-driven manner, providing a flexible yet structured framework that adapts to changes in risk premia and curve format, while maintaining interpretability and robustness.

A relevant practical advantage of the Mamba architecture in this context is its ability to deliver strong performance without requiring the massive datasets typically associated with deep learning models for time series. Unlike transformer-based architectures, which rely heavily on large-scale data to estimate attention patterns and avoid overfitting, structured state space models (SSMs) such as Mamba impose an inductive bias tailored to sequential data through their state-space formulation. This structure allows the model to efficiently capture temporal dependencies, persistence, and decay dynamics that are intrinsic to financial time series, even in relatively small samples. In the case of yield curves, where data availability is limited by frequency and historical depth, this property is particularly important. Moreover, the incorporation of cross-sectional structure via B-splines and economically motivated penalizations further reduces the effective complexity of the learning problem by constraining the space of admissible solutions. As a result, the Mamba + No-Arbitrage B-Spline framework achieves a favorable bias–variance trade-off, leveraging domain structure to compensate for limited data and avoiding the data inefficiency that often characterizes more generic deep learning approaches.

In this paper, we investigate the use of Mamba structured state space models for out-of-sample

forecasting of the term structure of interest rates. We evaluate the performance of the Mamba approximation through extensive Monte Carlo experiments, considering a variety of data generating processes (DGPs) that are standard in the term structure literature. Specifically, we simulate yields using the Nelson-Siegel/Diebold-Li framework (Nelson and Siegel, 1987; Diebold and Li, 2006), multifactor Heath-Jarrow-Morton models with stochastic volatility and jumps Heath et al. (1992); Musiela and Rutkowski (2005); Cheridito et al. (2007), as well as multifactor affine term structure models (Duffie and Kan, 1996; Dai and Singleton, 2000), including versions with regime switching (Ang and Bekaert, 2002; Dai et al., 2007). For each DGP, we compare the out-of-sample forecasts obtained from the Mamba SSM with those produced by the corresponding Diebold-Li models, assessing the ability of the Mamba architecture to capture the latent dynamics and curve shapes under different structural and stochastic assumptions. This experimental setup allows us to rigorously evaluate the flexibility and accuracy of the Mamba approach in approximating a wide range of realistic yield curve dynamics.

In addition to the Monte Carlo simulations, we evaluate the Mamba structured state space model on real-world yield curve data in order to assess both its practical forecasting performance and its ability to accurately reconstruct the observed term structure of interest rates. Using publicly available sovereign yield data across multiple maturities, we construct an empirical exercise that combines an in-sample fitting analysis with an out-of-sample forecasting evaluation.

First, we perform an out-of-sample forecasting exercise. The Mamba model is trained on historical observations and used to predict yields at future horizons, which are then compared with the realized yield curves. To assess predictive performance, we benchmark the Mamba forecasts against those produced by the standard Diebold–Li model estimated over the same rolling historical windows. This setup allows for a direct comparison of predictive accuracy across maturities.

Second, the model is estimated using historical observations of the yield curve, allowing us to examine how well the proposed architecture reproduces the cross-sectional structure of observed yields. The estimation incorporates the B-spline representation of the yield curve together with the smoothness and forward-rate penalization terms described in the previous sections. These penalties introduce economically motivated regularization by encouraging smooth yield curves and discouraging violations of basic no-arbitrage conditions. The strength of the penalization parameters is selected using Bayesian optimization, ensuring that the model balances goodness-of-fit with economically consistent curve shapes. The resulting in-sample fitted curves closely track the observed yields across maturities while displaying a smoother structure than the raw data, reducing local irregularities and providing a stable representation of the term structure.

This real-data exercise complements the controlled Monte Carlo study by illustrating the model's ability to adapt to complex and time-varying structures in observed term structures, including changes in risk premia, shifts in the slope of the yield curve, and nonlinear dynamics that are difficult to capture with rigid parametric specifications. At the same time, the inclusion of smoothness and no-arbitrage penalties ensures that the fitted and predicted curves remain economically plausible, demonstrating how the proposed Mamba + No-Arbitrage B-Spline framework combines the flexibility of modern neural sequence models with the structural discipline of financial term structure theory.

Our results indicate that the Mamba class of structured state space models provides a robust improvement in out-of-sample forecasting performance relative to traditional approaches. The model's ability to learn latent representations of the yield curve allows it to capture complex and time-varying dynamics that are difficult to represent with conventional parametric or factor-based models. By approximating the underlying latent structure and the intricate shape of the term structure, the Mamba approach demonstrates both flexibility and predictive accuracy, suggesting its practical relevance for financial applications requiring reliable forecasts of interest rates across multiple maturities.

2 Neural Structured State Space Models for Term Structure Modeling

2.1 State Space Representation of Dynamic Term Structure Models

Dynamic term structure models (DTSMs) can be naturally formulated in state space form (Kalman, 1960; Durbin and Koopman, 2012), which provides a flexible framework to model the joint dynamics of latent factors and observed yields (Diebold and Rudebusch, 2013). Let $x_t \in \mathbb{R}^k$ denote a vector of

latent state variables driving the evolution of the yield curve. The general linear Gaussian state space representation is given by

$$x_{t+1} = \mu + \Phi x_t + \Sigma \varepsilon_{t+1}, \quad \varepsilon_{t+1} \sim \mathcal{N}(0, I_k), \tag{1}$$

$$y_t = \Lambda(\tau) x_t + \eta_t, \quad \eta_t \sim \mathcal{N}(0, H), \tag{2}$$

where $y_t \in \mathbb{R}^n$ is the vector of observed yields at maturities $\tau = (\tau_1, \ldots, \tau_n)'$, μ is a drift vector, Φ is the state transition matrix, and Σ controls the volatility of the latent factors. The matrix $\Lambda(\tau)$ contains the factor loadings that map the latent states into yields at different maturities. The measurement errors η_t capture pricing errors or microstructure noise and are typically assumed to be independent of the state innovations.

Within this framework, no-arbitrage restrictions can be imposed by specifying the pricing kernel and deriving the bond prices implied by the dynamics of the state vector under the risk-neutral measure. A widely used class of models satisfying no-arbitrage is the affine term structure model (ATSM) (Duffie and Kan, 1996; Dai and Singleton, 2000; Duffee, 2002), in which bond prices are affine functions of the state variables. As a leading example, consider an affine-exponential term structure model. Let the short rate be given by

$$r_t = \delta_0 + \delta_1' x_t, \tag{3}$$

and assume that under the risk-neutral measure $\mathbb{Q}$ the state dynamics follow

$$x_{t+1} = \mu^{\mathbb{Q}} + \Phi^{\mathbb{Q}} x_t + \Sigma \varepsilon_{t+1}^{\mathbb{Q}}. \tag{4}$$

Under standard regularity conditions, the price of a zero-coupon bond with maturity τ takes the exponentially affine form

$$P(t, \tau) = \exp\left(A(\tau) + B(\tau)' x_t\right), \tag{5}$$

where $A(\tau)$ and $B(\tau)$ satisfy a system of Riccati difference (or differential) equations determined by the parameters of the model (Filipović, 2009). Consequently, the yield to maturity is affine in the state vector:

$$y(t, \tau) = -\frac{1}{\tau} \log P(t, \tau) = -\frac{A(\tau)}{\tau} - \frac{B(\tau)'}{\tau} x_t. \tag{6}$$

This affine-exponential structure implies that yields are linear functions of the latent factors, while bond prices are exponential-affine in the states. The state space representation thus provides a convenient econometric formulation for estimation and forecasting, typically via the Kalman filter or Bayesian methods, while maintaining consistency with no-arbitrage restrictions implied by asset pricing theory.

The Heath–Jarrow–Morton (HJM) (Heath et al., 1992; Musiela and Rutkowski, 2005) framework provides a general arbitrage-free methodology for modeling the evolution of the entire forward rate curve directly. Let $f(t, T)$ denote the instantaneous forward rate at time t for maturity $T \geq t$. In its general form under the risk-neutral measure $\mathbb{Q}$, the HJM dynamics are given by

$$df(t, T) = \alpha(t, T)\, dt + \sum_{k=1}^{d} \sigma_k(t, T)\, dW_t^{(k)}, \tag{7}$$

where $\{W_t^{(k)}\}_{k=1}^d$ are Brownian motions and $\sigma_k(t, T)$ are forward-rate volatility functions. The absence of arbitrage imposes a crucial drift restriction:

$$\alpha(t, T) = \sum_{k=1}^{d} \sigma_k(t, T) \int_t^T \sigma_k(t, u)\, du. \tag{8}$$

Thus, once the volatility structure is specified, the drift is uniquely determined by the no-arbitrage condition. This is a defining feature of the HJM framework: the modeler specifies the volatility of the forward curve, and arbitrage-freeness automatically determines the drift. In affine models, the entire

yield curve is determined by a finite set of Markov state variables. This structure ensures analytical tractability and closed-form bond pricing formulas. However, it imposes strong restrictions: the term structure dynamics are necessarily finite-dimensional and Markovian, and volatility is typically state-dependent but not directly maturity-specific. In contrast, HJM models the forward curve itself as the primitive object. The state variable is infinite-dimensional (the full forward curve), and Markovianity is not required unless explicitly imposed. While certain choices of volatility functions reduce HJM to an affine finite-factor representation, the general framework allows much richer dynamics. The HJM framework offers a broader and more flexible arbitrage-free modeling environment capable of incorporating non-Markovian features, stochastic volatility, and jumps in a natural and maturity-sensitive manner.

In parallel to no-arbitrage affine and HJM specifications, a large strand of the literature has developed purely statistical models for the term structure of interest rates, designed to provide flexible approximations of the yield curve without necessarily imposing structural pricing restrictions. These models typically represent yields as smooth functions of maturity driven by a small number of latent factors, with the functional form chosen to ensure tractable estimation and accurate cross-sectional fit, by using splines and smoothing splines (McCulloch, 1971; Fisher et al., 1995; Laurini and Moura, 2010; Laurini, 2014) or extensions of the Nelson-Siegel and Diebold-Li models Svensson (1994); Laurini and Hotta (2010, 2014).

By relaxing the tight economic constraints of affine-exponential models, statistical approaches offer greater flexibility in capturing local curvature, hump-shaped patterns, and time-varying shapes of the yield curve. As a result, they are particularly well suited for interpolation across maturities, construction of hedging portfolios, and short- to medium-term forecasting exercises. Although they may not enforce strict no-arbitrage conditions Christensen et al. (2011), their adaptability and empirical performance make them attractive tools in practical fixed-income applications.

Among statistical representations, the Nelson-Siegel (Nelson and Siegel, 1987) model has become one of the most widely used parametric specifications for the cross-section of yields. It represents the yield curve as a linear combination of level, slope, and curvature components, with maturity-dependent loadings that generate flexible shapes such as monotonic or hump-shaped curves. Formally, the yield at maturity τ is expressed as

$$y(t,\tau) = \beta_{1t} + \beta_{2t}\left(\frac{1-e^{-\lambda\tau}}{\lambda\tau}\right) + \beta_{3t}\left(\frac{1-e^{-\lambda\tau}}{\lambda\tau} - e^{-\lambda\tau}\right), \tag{9}$$

where β_{1t}, β_{2t}, and β_{3t} are time-varying latent factors and λ controls the decay rate of the loadings. The Diebold–Li extension (Diebold and Li, 2006; Diebold and Rudebusch, 2013) recasts this specification into a dynamic framework by modeling the latent factors as autoregressive processes, thereby embedding the Nelson-Siegel structure within a state space system suitable for estimation via the Kalman filter and for out-of-sample forecasting (Diebold et al., 2006). Owing to their parsimony, interpretability, and strong empirical performance, the Nelson-Siegel and Diebold-Li models have become benchmark statistical approaches for interpolation, risk management, and forecasting of the term structure.

While the Gaussian affine specification provides analytical tractability and economic interpretability, it imposes strong functional form restrictions on both the transition dynamics and the cross-sectional mapping from states to yields. In particular, the affine structure implies that yields are linear in the latent factors and that factor dynamics evolve according to linear autoregressive processes. These restrictions may be overly rigid in environments characterized by time-varying risk premia, nonlinear feedback effects, stochastic volatility, or regime changes. A natural extension is to generalize the state space system by allowing nonlinear transition and measurement equations,

$$x_{t+1} = f_\theta(x_t) + \Sigma_\theta(x_t)\varepsilon_{t+1}, \tag{10}$$

$$y_t = g_\theta(x_t,\tau) + \eta_t, \tag{11}$$

where $f_\theta(\cdot)$ and $g_\theta(\cdot)$ are flexible function classes parameterized by θ, potentially represented by neural networks. In this formulation, the latent state dynamics and the yield loading structure are learned directly from the data, rather than imposed a priori.

Structured State Space Models (SSMs) (Gu et al., 2022, 2020) implemented through neural architectures provide a scalable and computationally efficient mechanism to learn high-dimensional latent

representations and long-range temporal dependencies. By replacing fixed affine mappings with learned operators, neural SSMs retain the sequential structure of classical DTSMs while substantially increasing their capacity to approximate complex and time-varying term structure dynamics.

Structured State Space Sequence (S4) models (Gu et al., 2022, 2020) represent a recent class of neural architectures designed to efficiently model long-range dependencies in sequential data. Unlike traditional recurrent neural networks (RNNs) or standard Transformer architectures, S4 models leverage continuous-time state space representations, which allow them to capture both local and global temporal patterns with favorable computational and memory complexity.

At the core of S4 is the *linear state space model* (SSM), which describes a hidden state $x(t) \in \mathbb{R}^d$ evolving over time according to

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t), \qquad y(t) = Cx(t) + Du(t), \tag{12}$$

where $u(t)$ is the input sequence, $y(t)$ is the output, and A, B, C, D are matrices defining the system dynamics. This formulation provides a mathematically principled way to model sequences, drawing upon decades of control theory and signal processing literature. In discrete time, the system can be equivalently written as

$$x_{k+1} = \tilde{A}x_k + \tilde{B}u_k, \qquad y_k = \tilde{C}x_k + \tilde{D}u_k, \tag{13}$$

where $\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}$ are the discretized counterparts of the continuous-time matrices, computed via analytical methods such as the matrix exponential.

The innovation of structured SSMs lies in *parameterizing A in a highly structured form*, for example using the *HiPPO* (High-order Polynomial Projection Operator) framework (Gu et al., 2020), which ensures stability and efficient representation of long-range dependencies. By imposing a special structure on A, the model can maintain long-term memory without the vanishing gradient issues commonly encountered in RNNs. Additionally, convolutional techniques (Lecun et al., 1998) are often used to compute the hidden-to-output sequence transformations in $\mathcal{O}(L \log L)$ time for sequences of length L, enabling scalable training on very long sequences.

Structured SSMs have demonstrated remarkable performance in diverse domains such as natural language processing, time series forecasting, and speech processing. Their continuous-time formulation allows them to handle irregularly sampled data and multivariate sequences naturally. In financial applications, for instance, S4 models can be employed to model term structures of interest rates or high-frequency trading data, capturing both short-term fluctuations and long-term trends in a unified framework.

Structured state space sequence models are particularly well-suited for financial applications where capturing both short-term fluctuations and long-term dependencies is essential. One prominent example is the modeling of the yield curve, which describes the term structure of interest rates as a function of maturity τ. Let $y_t(\tau)$ denote the continuously compounded zero-coupon yield at time t and maturity τ. Traditional models, such as the Nelson-Siegel or affine term structure models parameterize the yield curve using a small number of latent factors. However, these approaches often struggle to incorporate high-frequency dynamics or nonlinear temporal patterns across maturities.

S4 and structured SSMs can extend this framework by treating the entire yield curve as a multivariate sequence and learning its evolution over time:

$$\mathbf{y}_t = f_\theta(\mathbf{y}_{t-1}, \mathbf{y}_{t-2}, \ldots, \mathbf{y}_{t-L}) + \boldsymbol{\epsilon}_t, \tag{14}$$

where $\mathbf{y}_t \in \mathbb{R}^{n_\tau}$ represents yields across n_τ maturities at time t, L is the look-back sequence length, f_θ is a structured SSM (e.g., Mamba or S4) parameterized by θ, and $\boldsymbol{\epsilon}_t$ captures measurement noise or market microstructure effects.

The structured nature of the hidden dynamics allows the model to efficiently propagate information from short to long maturities, naturally capturing level, slope, and curvature effects of the yield curve. Furthermore, by incorporating stochastic volatility or jump components into the state evolution, these models can replicate realistic market behaviors such as sudden spikes in short-term rates or regime shifts in long-term yields. Mathematically, the hidden factor evolution can be expressed as

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \boldsymbol{\eta}_t, \qquad \mathbf{y}_t = C\mathbf{x}_t + D\mathbf{u}_t + \boldsymbol{\epsilon}_t, \tag{15}$$

where $\mathbf{x}_t$ encodes latent factors governing the term structure, $\mathbf{u}_t$ may include exogenous macro-financial variables, and $\boldsymbol{\eta}_t$ represents process noise or stochastic volatility components. The matrices A, B, C, D

are structured to ensure stability and interpretable factor evolution, often leveraging the HiPPO parameterization for long-range memory retention.

By combining the rigorous foundations of linear dynamical systems with deep learning flexibility, S4 models provide a unified framework for modeling complex financial sequences. They facilitate accurate out-of-sample forecasting, scenario analysis, and risk management, all while maintaining computational efficiency even for large numbers of maturities or high-frequency observations.

The Mamba architecture (Gu and Dao, 2024; Dao and Gu, 2024) extends the class of Structured State Space Models (SSMs) by introducing a *selective* mechanism that allows the state transition and input interactions to depend on the input sequence itself. Classical SSMs such as S4 are based on linear time-invariant (LTI) dynamics of the form

$$h_t = Ah_{t-1} + Bu_t, \tag{16}$$
$$y_t = Ch_t, \tag{17}$$

where A, B, C are fixed matrices shared across time. While this structure enables efficient convolutional implementations and strong long-range modeling capacity, it limits expressiveness because the dynamics are independent of the content of the input sequence.

Mamba generalizes this formulation by allowing certain parameters of the SSM to become *input-dependent*. In particular, instead of fixed (A, B, C), the model introduces a selective mechanism where the state update becomes

$$h_t = A(u_t)h_{t-1} + B(u_t)u_t, \tag{18}$$
$$y_t = C(u_t)h_t, \tag{19}$$

where the dependence on u_t is implemented through lightweight learned projections. This transformation converts the LTI system into a *selective state space model*, preserving the recurrent structure while allowing the model to dynamically modulate how information is stored, forgotten, or propagated. The key insight is that selection acts as a form of content-based gating, analogous to attention, but implemented within a linear-time recurrent framework.

Crucially, Mamba maintains structured parameterizations of the state transition matrix A (typically diagonal plus low-rank or other structured forms ensuring stability and efficient discretization), so that the model preserves the favorable numerical properties and long-range memory of prior SSM approaches. The discretized dynamics can still be expressed in a form amenable to fast parallel scan algorithms, ensuring linear complexity in sequence length.

A central contribution of Mamba is the development of a hardware-aware parallel scan algorithm for selective SSMs. Classical convolution-based acceleration used in LTI SSMs no longer applies when parameters vary with time. Mamba instead reformulates the recurrence in a way that enables efficient prefix-scan computation, allowing parallelization across sequence elements while preserving exact recurrent semantics. This yields $\mathcal{O}(L)$ complexity with favorable constant factors, making the architecture competitive with Transformers in practice while avoiding quadratic attention costs.

The representation learned by Mamba can therefore be interpreted as a dynamically modulated latent state space, in which the hidden state evolves according to structured linear dynamics whose coefficients are selected at each time step. This dynamic formulation enables the model to capture long-range dependencies through structured state-space model (SSM) dynamics while simultaneously incorporating content-dependent gating mechanisms that are analogous to attention, all while maintaining linear-time computational scalability.

At the architectural level, a Mamba block replaces the self-attention module of a Transformer with a selective SSM layer. In a typical block, the input is first projected to expand the feature dimension, after which the selective SSM layer performs an input-dependent update of the hidden state. A gating mechanism then modulates the flow of information through the block, and the resulting features are projected back to the original model dimension. Finally, residual connections and normalization are applied to stabilize training and preserve gradient flow.

Unlike Transformers, Mamba does not compute pairwise token interactions. Instead, it relies on structured recurrent dynamics with learned selection, enabling strong performance on long sequences with significantly reduced memory overhead. The architecture thus unifies ideas from state space modeling, gated RNNs, and attention mechanisms into a single scalable framework.

Overall, the Mamba extension can be viewed as transforming classical structured SSMs from fixed linear dynamical systems into *selective, input-adaptive dynamical systems*, preserving stability and efficiency while dramatically increasing expressive power.

3 Mamba models architectures

To approximate the dynamic evolution of the yield curve in a data-driven manner, we employ a deep state-space model implemented as a PyTorch module, denoted as `YieldCurveMamba`. The architecture consists of three main components: an input projection layer, a recurrent state-space module, and an output projection layer. The input projection layer is a linear mapping from the vector of yields across $n_{\text{maturities}}$ to a higher-dimensional latent representation ($d_{\text{model}} = 64$), which serves as the embedding space for capturing complex interdependencies between maturities. The core module, `Mamba`, implements a recurrent state-space structure with $d_{\text{state}} = 16$ latent states, $d_{\text{conv}} = 4$ convolutional layers, and an expansion factor of 2, allowing the model to flexibly propagate and transform hidden state dynamics across the temporal sequence. This structure effectively generalizes classical AR(1) dynamics to a non-linear, high-dimensional latent space while preserving temporal coherence. Finally, the output projection layer maps the final hidden representation back to the original maturity space, producing a forecast for the next-step yield vector.

Formally, for an input sequence of past yields $X_{t-\text{seq\_len}:t}$, the forward pass of the network can be expressed as

$$\hat{y}_t = \text{OutputProj}\Big(\text{Mamba}(\text{InputProj}(X_{t-\text{seq\_len}:t}))\Big),$$

where InputProj and OutputProj are linear transformations, and Mamba encodes the recurrent state-space evolution. This architecture allows the model to capture both smooth yield curve evolution and abrupt changes, providing a flexible, non-parametric approximation to the underlying Nelson–Siegel dynamics. The combination of the parametric baseline (Nelson–Siegel) and the deep state-space approximation enables robust out-of-sample forecasting while preserving interpretability in terms of latent factors.

3.1 Enhanced Yield Curve Approximation with LSTM-Mamba Hybrid

In addition to the original Mamba architecture, we propose an enhanced baseline model that extends the original Mamba state-space network by incorporating a recurrent LSTM layer (Hochreiter and Schmidhuber, 1997) and deeper input-output projections to better capture complex temporal dependencies and non-linear interactions across maturities. The model operates on sequences of past yield curves, $X_{t-\text{seq\_len}:t} \in \mathbb{R}^{\text{seq\_len} \times n_{\text{maturities}}}$, and outputs the forecasted yield vector $\hat{y}_t$ for the next time step.

The initial projection consists of a two-layer feedforward network with ReLU activations and dropout regularization, mapping the input yields to a latent space of dimension d_{model}. This allows the model to encode non-linear relationships between short and long maturities and to provide a richer embedding for the subsequent recurrent processing.

Following the projection, the sequence is processed by an LSTM with *lstm_hidden* hidden units and *lstm_layers* layers. The LSTM captures temporal dependencies in the latent embedding space, enabling the model to recognize patterns such as persistence in level, slope, and curvature factors, as well as abrupt shifts in the term structure. Dropout is applied between LSTM layers to prevent overfitting when multiple layers are stacked.

The LSTM output feeds into the Mamba state-space module, which models the latent dynamics through d_{state} hidden states and d_{conv} convolutional transformations, with an expansion factor of expand. This hybrid structure combines the advantages of classical state-space models—such as interpretable latent factor dynamics—with the expressive power of deep convolutional networks, allowing non-linear interactions among latent states over time.

Finally, the Mamba output passes through a second deep projection, again using ReLU activations and dropout, to map back to the original maturity space. This output projection layer generates the forecasted yield vector $\hat{y}_t \in \mathbb{R}^{n_{\text{maturities}}}$, effectively approximating the conditional expectation of yields given the past sequence.

The model additionally registers forward hooks on the internal Mamba state-space layers to save latent matrices (A, B, C) and hidden states for analysis or potential regularization. This provides interpretability, as the learned matrices can be compared with theoretical state-space structures such as those in Nelson–Siegel or affine term structure models.

Formally, the forward pass of the model is

$$\hat{y}_t = \text{OutputProj}\Big(\text{Mamba}(\text{LSTM}(\text{InputProj}(X_{t-\text{seq\_len}:t})))\Big),$$

where InputProj and OutputProj are deep feedforward networks, LSTM captures temporal dependencies, and Mamba propagates latent state-space dynamics. The combination of deep projections, recurrent memory, and state-space approximation enables the model to handle both smooth evolution and abrupt shifts in the yield curve, providing a highly flexible and interpretable forecasting tool.

4 Yield Curve Approximation with BSpline-Mamba Hybrid

The BSpline-Mamba model extends the previous LSTM-Mamba architecture by incorporating a B-spline layer at the output, allowing the model to approximate the entire yield curve as a smooth functional surface rather than predicting individual maturities directly. This approach leverages both the temporal modeling capacity of recurrent networks and the flexibility of spline bases for functional approximation.

The input sequence of past yield curves, $X_{t-\text{seq\_len}:t} \in \mathbb{R}^{\text{seq\_len} \times n_{\text{maturities}}}$, is first projected through a deep feedforward network with ReLU activations, mapping the observed yields to a latent space of dimension d_{model}. This initial projection captures non-linear interactions between maturities and prepares the sequence for temporal processing.

Next, an LSTM layer with 64 hidden units processes the latent sequence to extract temporal dependencies, capturing persistent level, slope, and curvature movements in the yield curve. The LSTM output is then fed into the Mamba state-space module, which encodes latent factor dynamics via d_{state} states and d_{conv} convolutional transformations with expansion factor expand. This combination allows the network to model both smooth transitions and complex latent interactions over time.

Unlike previous models that project Mamba outputs directly to the yield vector, here the network predicts the coefficients of a set of n_{basis} B-spline functions, $c_t \in \mathbb{R}^{n_{\text{basis}}}$. The B-spline basis functions are defined over the maturity grid τ, with a cubic degree (3) and knots chosen to cover the full range of observed maturities. The spline layer reconstructs the forecasted yield curve as a linear combination of these basis functions:

$$\hat{y}_t(\tau) = \sum_{i=1}^{n_{\text{basis}}} c_{t,i} B_i(\tau),$$

where $B_i(\tau)$ are the pre-computed B-spline basis functions. This design ensures smoothness across maturities, enforces functional coherence, and reduces the effective dimensionality of the output space, providing both computational efficiency and regularization.

Formally, the forward pass can be summarized as:

$$\hat{y}_t = \text{BSplineLayer}\Big(\text{OutputProj}\big(\text{Mamba}(\text{LSTM}(\text{InputProj}(X_{t-\text{seq\_len}:t})))\big)\Big).$$

The combination of deep projection, recurrent memory, latent state-space dynamics, and B-spline functional approximation makes this architecture highly flexible for modeling yield curves, capturing temporal dependencies, latent factors, and smooth term structure variations while maintaining interpretability through the spline coefficients.

The inclusion of the B-spline layer also has important implications for term structure modeling and no-arbitrage conditions. By representing the yield curve as a smooth functional surface, the model naturally prevents unrealistic kinks or discontinuities between adjacent maturities that could violate basic arbitrage constraints (Laurini and Moura, 2010). Smooth curves ensure that forward rates derived from the projected yield curve remain well-behaved, reducing the likelihood of negative implied discount factors or locally inverted sections that would be inconsistent with no-arbitrage principles. Furthermore, the lower-dimensional B-spline representation imposes an implicit regularization that mitigates overfitting to noise in the observed yields, improving both the stability and economic plausibility of multi-step forecasts across the term structure.

Beyond ensuring smoothness, the B-spline framework can be leveraged to explicitly enforce no-arbitrage constraints in the construction of yield curves. By imposing linear inequality constraints on the spline coefficients, it is possible to guarantee that the implied discount function is monotonically decreasing and that forward rates remain positive, which are fundamental requirements for arbitrage-free pricing (Laurini and Moura, 2010). Additional regularization terms can be incorporated to penalize violations of convexity or local slope restrictions, ensuring that the reconstructed yield curve maintains economically consistent shapes. In this way, the B-spline layer functions not only as a smoothing mechanism but also as a means to embed structural financial constraints directly into the network outputs, enhancing both interpretability and adherence to fundamental no-arbitrage principles.

The B-spline layer also computes a matrix of second derivatives, B'', which is used to define a smoothness penalty:

$$\mathcal{L}_{\text{smooth}} = \frac{1}{n_{\text{batch}}} \sum_{i=1}^{n_{\text{batch}}} \left\| c_i B'' \right\|_2^2.$$

The total loss combines the mean squared error (MSE) with this smoothness penalty:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MSE}} + \lambda_{\text{smooth}} \mathcal{L}_{\text{smooth}},$$

where λ_{smooth} controls the strength of the smoothness regularization.

To ensure economically meaningful forecasts, the model penalizes deviations from no-arbitrage conditions. The predicted discount factor curve, $D(\tau) = \exp(-\tau \hat{y}(\tau))$, is enforced to be monotonically decreasing by penalizing negative derivatives, while the instantaneous forward rate, $f(\tau) = \hat{y}(\tau) + \tau \frac{\partial \hat{y}}{\partial \tau}$, is penalized if it becomes negative:

$$\mathcal{L}_{\text{fwd}} = \text{mean}(\text{ReLU}(-f(\tau))).$$

The final loss function therefore incorporates both smoothness and no-arbitrage penalties:

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{MSE}} + \lambda_{\text{smooth}} \mathcal{L}_{\text{smooth}} + \lambda_{\text{fwd}} \mathcal{L}_{\text{fwd}},$$

ensuring that the predicted yield curves are accurate, smooth, and consistent with fundamental economic constraints, creating a additional method (Mamba + No-Arbitrage B-Spline) that imposes no-arbitrage restrictions.

Conceptually, this approach is closely related to the idea of Physics-Informed Neural Networks (PINNs) (Raissi et al., 2019). In the PINN framework, neural networks are trained not only to fit observed data but also to satisfy governing physical equations, typically expressed as partial differential equations, through additional loss penalties. In an analogous manner, the present model embeds economic structure directly into the learning objective. Rather than enforcing physical laws, the model imposes financial equilibrium conditions derived from no-arbitrage theory. The resulting architecture can therefore be interpreted as an *Finance-Informed Neural Network - FINN*, in which arbitrage restrictions act as structural constraints guiding the learning process.

This analogy highlights an important methodological perspective: just as PINNs incorporate physical laws to improve generalization and ensure physically consistent predictions, the inclusion of no-arbitrage constraints provides economically coherent regularization for yield curve estimation. By embedding these structural restrictions directly into the training objective, the model combines the flexibility of deep learning architectures with the theoretical discipline of arbitrage-free term structure modeling.

5 Yield Curve Mamba with Smoothness Regularization

This variant of the YieldCurveMamba model builds upon the LSTM-Mamba architecture by explicitly incorporating smoothness constraints over maturities. The goal is to ensure that the predicted yield curve is not only accurate but also smooth, reflecting realistic term structure behavior where adjacent maturities exhibit gradual changes rather than abrupt fluctuations.

The input sequence $X_{t-\text{seq\_len}:t} \in \mathbb{R}^{\text{seq\_len} \times n_{\text{maturities}}}$ is first projected through a two-layer feedforward network with ReLU activations and dropout. This initial projection captures non-linear interactions between maturities while providing regularization to prevent overfitting. The projected sequence

is then processed by an LSTM network with $lstm\_hidden$ units and $lstm\_layers$ layers, which extracts temporal dependencies and long-memory effects in the evolution of the yield curve factors.

Following the LSTM, the latent sequence is passed to the Mamba state-space module, which encodes complex latent dynamics through d_{state} hidden states and d_{conv} convolutional transformations with expansion factor expand. The Mamba module enhances the network's ability to model smooth latent trajectories and capture persistent shifts in the yield curve's level, slope, and curvature.

The output of Mamba is projected to the maturity space through a multi-layer feedforward network, and a learnable bias vector $\mathbf{b} \in \mathbb{R}^{n_{\text{maturities}}}$ is added to allow global intercepts per maturity. This ensures that systematic differences across maturities are captured, providing better alignment with the average term structure.

Crucially, the model includes a smoothness penalty, defined as the mean squared second derivative of the predicted yields across maturities:

$$\mathcal{L}_{\text{smooth}} = \frac{1}{n_{\text{batch}}} \sum_{i=1}^{n_{\text{batch}}} \frac{1}{n_\tau} \sum_{j=1}^{n_\tau} \left(\frac{\partial^2 \hat{y}_i(\tau_j)}{\partial \tau^2} \right)^2,$$

which is added to the standard MSE loss during training. This penalty encourages the network to produce yield curves that vary smoothly with maturity, preventing unrealistic kinks or spikes, and reflecting the inherent continuity of financial term structures.

Formally, the forward computation of the model can be summarized as:

$$\hat{y}_t = \text{OutputProj}\big(\text{Mamba}(\text{LSTM}(\text{InputProj}(X_{t-\text{seq\_len}:t})))\big) + \mathbf{b},$$

with $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MSE}} + \lambda_{\text{smooth}} \mathcal{L}_{\text{smooth}}$ for a hyperparameter λ_{smooth} controlling the strength of the smoothness regularization.

This architecture effectively integrates temporal modeling, latent factor dynamics, and functional regularization, producing forecasts that are both accurate and consistent with economic intuition about smooth yield curves.

The Mamba model with a smoothness penalty and the BSpline-Mamba model share the same core building blocks—an input projection network, an LSTM for temporal dependencies, and the Mamba state-space module for latent factor dynamics—but they differ fundamentally in how the output yield curve is represented and regularized.

In the smoothness-penalized Mamba, the network directly predicts yields at each observed maturity. The smoothness of the curve is encouraged by adding a regularization term to the loss function, specifically the mean squared second derivative of the predicted yields across maturities. This approach ensures that the model discourages abrupt changes between adjacent maturities, enforcing gradual transitions that are consistent with realistic term structures. However, the smoothness is softly imposed via a penalty: the network still has the freedom to produce locally irregular behavior if it helps minimize the mean squared error, and the regularization strength must be carefully tuned with a hyperparameter λ_{smooth}.

By contrast, the BSpline-Mamba formulation replaces direct yield prediction with a functional representation of the curve through B-spline basis functions. Instead of predicting individual yields, the network outputs the coefficients of a set of pre-defined B-spline functions, and the yield curve is reconstructed as a linear combination of these basis functions. Smoothness is therefore hard-coded into the representation itself, as B-splines inherently produce continuous and smooth curves of the chosen degree (typically cubic). The BSpline-Mamba also allows additional structural constraints to be imposed on the coefficients, such as monotonicity of the discount function or positivity of forward rates, which directly enforce basic no-arbitrage conditions. This is in contrast to the smoothness penalty approach, which only indirectly encourages economically reasonable curves.

Another key difference lies in dimensionality and regularization. The smoothness-penalized Mamba predicts a yield value for each maturity independently, which can lead to high-dimensional outputs, whereas the BSpline-Mamba predicts a relatively small number of coefficients corresponding to the spline basis functions. This lower-dimensional representation acts as a natural regularizer, reducing the risk of overfitting to noise in the observed yields while maintaining flexibility to capture global shape and curvature.

5.1 Discussion and Relation to the Literature

Recent advances in machine learning have led to a growing interest in neural network architectures for modeling latent structures in economic and financial data. In particular, the work of Shen and Xiu (2024) on deep autoencoders for nonlinear factor models provides an important theoretical foundation for using neural networks as estimators of latent factor structures. Our approach, based on structured state space models implemented through the Mamba architecture, is closely related in spirit but differs in both modeling objectives and methodological design.

Shen and Xiu (2025) analyze autoencoders as estimators for nonlinear factor models in cross-sectional and panel settings. Their framework considers observations generated by a nonlinear factor structure of the form

$$X_{it} = \phi_i(F_t) + U_{it},$$

where F_t denotes a low-dimensional vector of latent factors and $\phi_i(\cdot)$ is a potentially nonlinear loading function. Within this setting, the encoder of the autoencoder extracts a low-dimensional representation that approximates the latent factors, while the decoder reconstructs the original observations from these embeddings. The main contribution of their work lies in establishing non-asymptotic theoretical guarantees for factor recovery and reconstruction accuracy, showing that deep autoencoders can consistently recover latent structures under general nonlinear factor models.

Our paper addresses a related but distinct problem. Rather than focusing on static nonlinear factor models, we study the dynamic evolution of the term structure of interest rates using structured state space models implemented via the Mamba architecture. In the standard dynamic term structure framework, the yield curve is driven by a vector of latent states x_t evolving according to

$$x_{t+1} = \mu + \Phi x_t + \Sigma \varepsilon_{t+1},$$

while observed yields across maturities are generated by

$$y_t = \Lambda(\tau) x_t + \eta_t.$$

Traditional models such as the Nelson–Siegel or affine term structure models impose strong parametric restrictions on both the factor loadings and the state dynamics. In contrast, our approach replaces these parametric assumptions with a neural structured state space model that learns both the latent dynamics and the mapping to observed yields directly from data.

Conceptually, both approaches rely on neural networks to uncover latent structures from high-dimensional data. However, the architectures differ fundamentally in how they treat the latent representation. In the autoencoder framework of Shen and Xiu (2024), the latent representation arises from a dimensionality reduction mechanism in which the encoder compresses the input data into a lower-dimensional embedding. The focus is primarily on recovering latent factors that explain cross-sectional variation. In contrast, our model treats the latent representation as a hidden state in a dynamic system. The Mamba architecture parameterizes the transition dynamics of these states through a structured recurrent state space module, allowing the model to capture temporal dependencies and long-range dynamics in the evolution of the yield curve.

Another key distinction concerns the role of economic structure. The autoencoder framework is largely model-agnostic, focusing on statistical recovery of latent factors without imposing domain-specific constraints. Our approach, by contrast, integrates financial structure directly into the learning problem. In particular, the proposed BSpline-Mamba hybrid incorporates smooth functional representations of the yield curve across maturities, while no-arbitrage considerations are enforced through shape constraints on discount and forward curves. This results in an finance-informed neural architecture that balances flexibility with theoretical consistency.

Despite these differences, the two approaches are complementary. The theoretical results in Shen and Xiu (2024) provide an important justification for using neural networks to estimate nonlinear latent factor structures. In many respects, the latent state representation learned by the Mamba state space model can be interpreted as a dynamic extension of the embeddings produced by autoencoders. Both methods rely on neural networks to approximate complex nonlinear mappings between observed data and low-dimensional latent structures.

From a methodological perspective, the main difference lies in the dimension along which flexibility is introduced. Autoencoders primarily address nonlinearities in cross-sectional relationships, while our approach focuses on nonlinear temporal dynamics and functional relationships across maturities.

Consequently, the two frameworks address different aspects of the modeling problem: nonlinear dimensionality reduction in the case of autoencoders, and nonlinear dynamic system identification in the case of structured state space models.

6 Monte Carlo Evidence

To evaluate the out-of-sample forecasting performance of the proposed architecture, we conduct a Monte Carlo experiment in which the data generating process (DGP) is first assumed to follow the Diebold–Li dynamic Nelson–Siegel specification. In each replication, synthetic yield curves are generated from the true Diebold–Li model with autoregressive latent factors, and we estimate competing forecasting models using only the simulated observable yields. We then compare the true Diebold–Li forecasting model, correctly specified under the DGP, with the proposed neural state space approximations: (i) a baseline Mamba model, (ii) a Mamba augmented with an LSTM pre-processing layer (Mamba+LSTM), (iii) a Mamba combined with a B-spline cross-sectional representation (Mamba+B-spline), (iv) the Mamba + No-Arbitrage B-Spline representation, and (v) a Mamba specification incorporating an explicit smoothness penalty on the second derivative of the yield curve. Forecast accuracy is evaluated using standard in and out-of-sample loss functions across maturities and horizons. This design allows us to assess whether the selective structured state space approximation can recover the underlying Diebold–Li dynamics and whether additional architectural constraints or smoothness regularization improve forecasting performance relative to both the correctly specified parametric benchmark and the flexible neural alternatives.

The Monte Carlo experiment consists of 200 independent replications, each generated using a distinct random seed to ensure reproducibility and statistical independence across simulations. In each replication, we simulate 1,000 time periods of yield curve data. The cross-section of yields is observed at seven maturities (0.1, 0.25, 0.5, 1, 2, 5, and 10 years). The deep learning model is trained using rolling input sequences of length 30, where each input contains the previous 30 yield curve observations and the target corresponds to the one-step-ahead yield curve.

For forecasting evaluation, the last 30 observations of each simulated sample are reserved as an out-of-sample period, while the remaining observations are used for model estimation and training. The neural network is reinitialized and trained from scratch in each replication using 200 epochs and the Adam optimizer with a learning rate of 0.001. The benchmark model is also re-estimated in every replication using only in-sample information, and its forecasts are generated recursively over the 30-period out-of-sample horizon.

Forecast accuracy is evaluated separately for each maturity using mean absolute error (MAE) and root mean squared error (RMSE), computed both in-sample and out-of-sample. For each replication and maturity, we additionally compute the Diebold–Mariano statistic with a Newey–West heteroskedasticity and autocorrelation consistent variance estimator to compare predictive accuracy across models at the 5% significance level. Final results are obtained by averaging all performance measures and test outcomes across the 200 Monte Carlo replications.

6.1 Baseline Nelson–Siegel (Diebold–Li) Model

The baseline model adopts the classical Nelson–Siegel specification for the term structure, representing yields as a function of three latent factors $\beta_1, \beta_2, \beta_3$ with fixed decay parameter $\lambda = 0.0609$, chosen to match typical medium-term curvature observed in empirical yield curves. Each factor evolves as a stationary AR(1) process with autoregressive coefficients $\phi = [0.95, 0.90, 0.85]$ and innovation standard deviations $\sigma = [0.03, 0.02, 0.02]$, reflecting the differing persistence and volatility of the level, slope, and curvature components of the yield curve. At each time step, the yield curve is generated by combining the three factors through the Nelson–Siegel functional form, optionally perturbed with small Gaussian noise (std = 0.05) to mimic market microstructure and measurement errors. This setup ensures realistic dynamics of the yield curve while preserving tractability for estimation. The Mamba neural network, equipped with a projection layer and a recurrent state-space module, is trained to map past yield sequences to the next-step yield, using mean-squared error as the objective and the Adam optimizer for efficient learning. Parallelly, the dynamic Nelson–Siegel procedure estimates factor values β_t at each step using nonlinear least squares, and fits independent AR(1) processes to forecast out-of-sample yields, providing a benchmark for comparison. This framework allows consistent evaluation

of parametric factor models against data-driven deep state-space representations, while the chosen parameters ensure factor stationarity, realistic volatility, and yield curve shapes across maturities ranging from one month to ten years.

Table 1: Monte Carlo results by maturity. DM rejection rate refers to the HAC-corrected Diebold–Mariano test at the 5% level.

	In-Sample				**Out-of-Sample**				**Forecast Comparison (OUT)**			
τ	MAE_M	MAE_{DL}	RMSE_M	RMSE_{DL}	MAE_M	MAE_{DL}	RMSE_M	RMSE_{DL}	DM Reject	DL Sup.	M Sup.	Mean DM
0.10	0.0503	0.0325	0.0632	0.0407	0.0504	0.0923	0.0627	0.1088	0.685	0.000	0.685	-2.582
0.25	0.0504	0.0338	0.0632	0.0423	0.0506	0.0916	0.0627	0.1084	0.660	0.000	0.660	-2.442
0.50	0.0503	0.0351	0.0631	0.0439	0.0514	0.0925	0.0640	0.1093	0.690	0.005	0.685	-2.500
1.00	0.0502	0.0360	0.0630	0.0451	0.0498	0.0917	0.0619	0.1084	0.690	0.010	0.680	-2.564
2.00	0.0501	0.0335	0.0628	0.0420	0.0498	0.0921	0.0622	0.1086	0.695	0.000	0.695	-2.616
5.00	0.0500	0.0220	0.0627	0.0276	0.0502	0.0906	0.0623	0.1070	0.655	0.000	0.655	-2.532
10.00	0.0499	0.0049	0.0625	0.0061	0.0506	0.0903	0.0628	0.1066	0.640	0.000	0.640	-2.533

The Monte Carlo results displayed in Table 1 the expected pattern under a correctly specified data-generating process. Since yields are generated from the classical Nelson–Siegel structure with AR(1) factor dynamics, the dynamic Diebold–Li (DL) model coincides with the true parametric specification. It therefore delivers superior in-sample performance across all maturities. Both MAE and RMSE are systematically lower for DL in-sample, particularly at longer maturities where the Nelson–Siegel factor loadings closely match the imposed functional form. This outcome reflects the standard efficiency property of correctly specified parametric models: when the model matches the DGP, least-squares estimation recovers the structure with minimal approximation error.

The more informative evidence arises from the out-of-sample comparison. Despite not imposing the Nelson–Siegel functional form nor the AR(1) dynamics explicitly, the Mamba-based state-space neural architecture achieves substantially lower OOS errors across all maturities. The improvement is economically meaningful, as DL out-of-sample MAE and RMSE are consistently larger than those of Mamba. The Diebold–Mariano statistics are negative on average and the rejection frequencies lie between approximately 64% and 70%, indicating statistically significant predictive gains for Mamba in a large fraction of replications.

This contrast between in-sample dominance of DL and OOS superiority of Mamba can be interpreted through finite-sample considerations. The DL procedure relies on a two-step estimation strategy, first extracting factors cross-sectionally and then fitting AR(1) dynamics, which introduces estimation noise and parameter uncertainty that propagate into forecasts. By contrast, the Mamba architecture learns the full mapping from past yield sequences to future yields within a unified framework, jointly internalizing factor extraction and dynamic evolution. The recurrent structured state-space module, combined with nonlinear projections, allows the network to approximate complex dynamic interactions across maturities, even when the true DGP is linear. In finite samples, this integrated learning mechanism can effectively smooth estimation noise and reduce forecast error.

The gains are remarkably stable across the maturity spectrum, from short-term to long-term yields, suggesting that the neural state-space representation captures both short-end volatility and long-end persistence without overfitting specific segments of the curve. Overall, the results indicate that while correct parametric specification ensures in-sample efficiency, flexible deep state-space representations can yield robust improvements in out-of-sample forecasting by jointly approximating cross-sectional and temporal dynamics of the term structure.

Table 2: Monte Carlo results by maturity. DL results omitted.

	Mamba + LSTM				**Mamba + B-Spline**				**Mamba + No-Arb B-Spline**				**Mamba Penalized**			
τ	MAE_{IN}	RMSE_{IN}	MAE_{OUT}	RMSE_{OUT}	MAE_{IN}	RMSE_{IN}	MAE_{OUT}	RMSE_{OUT}	MAE_{IN}	RMSE_{IN}	MAE_{OUT}	RMSE_{OUT}	MAE_{IN}	RMSE_{IN}	MAE_{OUT}	RMSE_{OUT}
0.10	0.0504	0.0632	0.0507	0.0630	0.0503	0.0631	0.0505	0.0628	0.0665	0.0691	0.0898	0.1051	0.0853	0.1066	0.0849	0.1010
0.25	0.0504	0.0631	0.0508	0.0630	0.0504	0.0631	0.0506	0.0628	0.0666	0.0691	0.0901	0.1075	0.0852	0.1066	0.0847	0.1007
0.50	0.0503	0.0631	0.0516	0.0643	0.0503	0.0631	0.0514	0.0640	0.0664	0.0690	0.0886	0.1064	0.0852	0.1064	0.0854	0.1017
1.00	0.0502	0.0629	0.0500	0.0621	0.0502	0.0629	0.0498	0.0620	0.0665	0.0691	0.0882	0.1045	0.0849	0.1062	0.0848	0.1010
2.00	0.0501	0.0628	0.0502	0.0626	0.0500	0.0627	0.0499	0.0623	0.0673	0.0699	0.0842	0.1014	0.0846	0.1058	0.0847	0.1009
5.00	0.0500	0.0627	0.0505	0.0627	0.0500	0.0626	0.0503	0.0624	0.0669	0.0694	0.0876	0.1068	0.0840	0.1050	0.0838	0.0998
10.00	0.0499	0.0625	0.0510	0.0631	0.0499	0.0625	0.0507	0.0629	0.0667	0.0691	0.0811	0.0971	0.0830	0.1037	0.0834	0.0994

The out-of-sample results for the alternative Mamba specifications, reported in Table 2, provide useful insight into the role of temporal structure, cross-sectional smoothing, curvature regularization, and no-arbitrage constraints in this controlled DGP.

Starting with Mamba + LSTM, the OOS MAE and RMSE are extremely close to those of the baseline Mamba across all maturities. In most cases, the differences appear only at the third decimal place and are not systematic across the maturity spectrum. This suggests that adding an LSTM layer on top of the structured state-space module does not materially improve predictive performance in this environment. Given that the true DGP is driven by linear AR(1) dynamics in three latent factors, the baseline Mamba architecture already provides sufficient temporal expressiveness to capture the relevant persistence structure. The LSTM therefore introduces additional flexibility without yielding measurable forecasting gains, indicating that the structured state-space component is already well aligned with the underlying dynamics.

The Mamba + B-spline specification exhibits slightly more stable and marginally improved OOS errors relative to the LSTM extension, particularly at medium and longer maturities. By replacing or augmenting the implicit cross-sectional mapping with a smooth B-spline representation across maturities, the model imposes additional shape regularity on the yield curve. Even though the true DGP follows the Nelson–Siegel functional form, which is itself smooth and low-dimensional, the spline-based representation approximates this structure effectively while retaining flexibility. The small but consistent reduction in OOS RMSE at several maturities suggests that mild cross-sectional smoothing helps reduce estimation noise without constraining the dynamics excessively.

The Mamba + No-Arbitrage B-Spline specification, which incorporates no-arbitrage restrictions by enforcing monotonicity in the discount curve and positivity in the forward rates, produces slightly higher in-sample errors relative to the unconstrained B-spline model, reflecting the additional structural constraints. Out-of-sample performance, however, is notably more robust at longer maturities, with lower RMSE compared to the unconstrained penalized Mamba. This indicates that enforcing economic consistency can stabilize forecasts, particularly in regions of the yield curve where estimation variance is naturally higher. The Mamba + No-Arbitrage B-Spline strikes a balance between smoothness and arbitrage-free constraints, achieving competitive OOS accuracy while guaranteeing economically plausible yield curve shapes.

In contrast, the Mamba Penalized specification performs substantially worse both in-sample and out-of-sample. The imposition of a strong second-derivative penalty enforces excessive curvature regularization across maturities, effectively oversmoothing the yield curve. Because the simulated DGP already embodies a specific parametric curvature pattern driven by the Nelson–Siegel loadings, this excessive penalization distorts the implied factor structure and dampens legitimate cross-sectional variation. As a result, both MAE and RMSE increase markedly across all maturities. The degradation is uniform and persistent, indicating that the penalization strength likely imposes a bias that dominates any variance reduction benefits.

Taken together, the comparison highlights several important insights. Moderate structural guidance—through smooth B-spline basis expansions—can enhance stability and marginally improve OOS performance by controlling cross-sectional noise. Incorporating no-arbitrage constraints further improves robustness at longer maturities, producing economically consistent forecasts without substantial loss in accuracy. Excessive penalization, on the other hand, introduces bias and harms predictive performance. Meanwhile, augmenting the temporal component with an LSTM does not materially improve forecasts in this linear AR(1)-driven environment, suggesting that the core Mamba state-space architecture is sufficiently expressive to capture the relevant dynamics.

6.2 Simulation Model: Three-Factor HJM with Stochastic Volatility

In this simulation, the evolution of the forward rate curve is modeled using a three-factor Heath-Jarrow-Morton (HJM) framework with stochastic volatility. Let $f(t,\tau)$ denote the instantaneous forward rate at time t for maturity τ. The dynamics are given by

$$df(t,\tau) = \alpha(t,\tau)\,dt + \sigma_1(\tau)\,dW_1(t) + \sigma_2(\tau)\,dW_2(t) + \sigma_3(\tau)\,dW_3(t), \tag{20}$$

where $\alpha(t,\tau)$ is the drift term implied by the no-arbitrage HJM condition, $\sigma_i(\tau)$ are deterministic volatility functions for each factor, and $W_i(t)$ are correlated Brownian motions with correlation matrix

$$\mathbf{C} = \begin{pmatrix} 1.0 & 0.5 & 0.3 \\ 0.5 & 1.0 & 0.4 \\ 0.3 & 0.4 & 1.0 \end{pmatrix}.$$

The volatility functions are chosen to capture stylized features of the yield curve. Specifically, $\sigma_1(\tau) = 0.02\exp(-0.3\tau)$ primarily affects short maturities, $\sigma_2(\tau) = 0.015\tau\exp(-0.4\tau)$ exhibits a hump-shaped effect for medium maturities, and $\sigma_3(\tau) = 0.01\exp(-0.1\tau)$ captures long-maturity dynamics. To incorporate stochastic fluctuations in overall volatility, we define a log-AR(1) process

$$\log \nu_t = \phi_{\text{vol}} \log \nu_{t-1} + \sigma_{\text{vol}}\epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0,1), \tag{21}$$

with $\phi_{\text{vol}} = 0.95$ and $\sigma_{\text{vol}} = 0.1$. The stochastic volatility scaling is then $\nu_t = \exp(\log \nu_t)$, which multiplies the deterministic factor volatilities. This specification ensures persistent volatility shocks while preserving positivity, consistent with observed interest rate dynamics.

Forward rates are simulated on a monthly time step $\Delta t = 1/12$ using an Euler-Maruyama discretization:

$$f_t(\tau_i) = f_{t-1}(\tau_i) + \alpha_{t-1}(\tau_i)\Delta t + \nu_t \sum_{j=1}^{3} \sigma_j(\tau_i)\, \Delta W_j(t), \tag{22}$$

where the drift $\alpha_{t-1}(\tau_i)$ is approximated using trapezoidal integration of the volatility functions:

$$\alpha_{t-1}(\tau_i) = \sum_{j=1}^{3} \sigma_j(\tau_i) \int_0^{\tau_i} \sigma_j(s) ds. \tag{23}$$

The initial forward curve is set to $f_0(\tau) = 0.02 + 0.01\exp(-0.2\tau)$, producing realistic short-term and long-term rates. The dense maturity grid $\tau \in [0.1, 10]$ years with 150 points ensures accurate capture of the forward curve evolution. Finally, the simulated forward rates are converted into zero-coupon yields at standard maturities, generating synthetic yield curves suitable for calibration and forecasting analysis.

This parameterization is consistent with empirical observations: the deterministic volatility shapes reproduce typical curve movements, the stochastic volatility AR(1) process introduces realistic temporal dependence, and the correlation structure allows for coherent co-movements among the three factors, in line with standard practices in interest rate modeling Brigo and Mercurio (2006).

Table 3: Monte Carlo out-of-sample results by maturity. DM rejection rate refers to the HAC-corrected Diebold–Mariano test at the 5% level.

	MAE (OUT)		**RMSE (OUT)**		**Forecast Comparison**			
τ	Mamba	DL	Mamba	DL	DM Reject	DL Sup.	Mamba Sup.	Mean DM
0.10	0.0020	0.0933	0.0025	0.0934	0.995	0.000	0.995	-80.390
0.25	0.0102	0.0157	0.0112	0.0177	0.720	0.165	0.555	-0.703
0.50	0.0125	0.0558	0.0140	0.0580	0.875	0.030	0.845	-7.878
1.00	0.0101	0.0505	0.0117	0.0530	0.915	0.020	0.895	-6.814
2.00	0.0084	0.0321	0.0101	0.0355	0.845	0.005	0.840	-4.021
5.00	0.0142	0.0462	0.0157	0.0487	0.850	0.075	0.775	-5.417
10.00	0.0149	0.0179	0.0160	0.0204	0.685	0.255	0.430	0.444

The Monte Carlo results under the stochastic-volatility HJM specification (Table 3) reveal a markedly different pattern from the Nelson–Siegel DGP and strongly favor the Mamba architecture in out-of-sample forecasting.

First, the magnitude of the forecasting gains is economically and statistically substantial, particularly at the short end of the curve. At the 0.10 maturity, the difference is dramatic: DL exhibits MAE and RMSE near 0.093, while Mamba achieves errors close to 0.002–0.003. The Diebold–Mariano rejection rate reaches 99.5%, with an extremely large negative mean DM statistic, indicating overwhelming and systematic predictive superiority of Mamba. This result reflects a fundamental model misspecification problem for DL. The HJM DGP generates forward rates with maturity-dependent volatilities, correlated shocks, and—crucially—stochastic volatility scaling through the persistent log-AR(1) process. The standard dynamic Nelson–Siegel framework, based on linear Gaussian AR(1) factor dynamics with constant volatility, cannot reproduce time-varying volatility nor the nonlinear propagation of shocks across maturities implied by the HJM drift condition.

Across intermediate maturities (0.25 to 5 years), Mamba continues to dominate, with lower MAE and RMSE in nearly all cases and rejection rates typically between 72% and 91%. The gains are particularly pronounced around medium maturities, where the hump-shaped volatility component and the interaction between correlated Brownian motions generate richer cross-sectional dynamics. Because the HJM drift depends on the integral of volatility functions, the evolution of forward rates embeds nonlinear maturity interactions that are not representable within the linear Nelson–Siegel loading structure. The structured state-space neural architecture is able to approximate these nonlinear cross-maturity mappings directly from the data, which explains its strong OOS performance.

At the long end (10 years), the gap narrows and the DM rejection rate falls to 68.5%, with a slightly positive mean DM statistic. This suggests that for the longest maturities the DL approximation becomes comparatively more competitive. One possible explanation is that the long-maturity volatility function is smoother and more slowly decaying, making the induced dynamics closer to a low-dimensional persistent factor structure that Nelson–Siegel can partially approximate. Nevertheless, even in this region, Mamba still achieves lower MAE and RMSE on average.

Overall, these results highlight the sensitivity of parametric term structure models to structural misspecification. When the DGP departs from linear Gaussian factor dynamics and incorporates stochastic volatility and maturity-dependent nonlinearities, the dynamic Nelson–Siegel model lacks the flexibility to adapt. In contrast, the Mamba state-space architecture—through its recurrent structured dynamics and nonlinear projection layers—can approximate both time-varying volatility effects and nonlinear factor loadings across maturities. The large and persistent OOS gains therefore provide evidence that flexible deep state-space representations are particularly advantageous in environments with stochastic volatility and non-affine cross-sectional dynamics, where traditional low-dimensional parametric factor models become restrictive.

6.3 Simulation Model with Jumps

The previous three-factor HJM framework with stochastic volatility is extended here to incorporate occasional discrete jumps in the forward rate dynamics. The deterministic volatility functions $\sigma_1(\tau)$, $\sigma_2(\tau)$, and $\sigma_3(\tau)$, the stochastic log-AR(1) volatility process, and the correlation structure among the Brownian motions remain unchanged. The primary modification consists of a Poisson jump component added to the forward rate evolution. Specifically, at each time step Δt, a jump occurs with probability $\lambda_{\text{jump}}\Delta t$, where $\lambda_{\text{jump}} = 0.08$ corresponds to an annualized jump intensity. When a jump occurs, its size is drawn from a normal distribution with mean $\mu_{\text{jump}} = 0$ and standard deviation $\sigma_{\text{jump}} = 0.02$, reflecting small, symmetric shocks consistent with observed abrupt interest rate movements. The forward rate dynamics are thus modified as

$$f_t(\tau_i) = f_{t-1}(\tau_i) + \alpha_{t-1}(\tau_i)\Delta t + \nu_t \sum_{j=1}^{3} \sigma_j(\tau_i)\, \Delta W_j(t) + J_t \sum_{j=1}^{3} \sigma_j(\tau_i), \tag{24}$$

where J_t is zero if no jump occurs or the sampled jump value if a jump occurs. The inclusion of J_t scaled by the sum of factor volatilities ensures that jumps impact the entire curve proportionally across maturities. This mechanism introduces discrete, sudden movements into the simulated yield curves while preserving the continuous diffusive dynamics of the original model. The remaining numerical setup, including the discretization grid for maturities, the Euler-Maruyama integration of the forward rates, and the conversion to zero-coupon yields, is identical to the baseline model. An additional small Gaussian noise with standard deviation 0.2% is also added to the yields to mimic measurement errors or market microstructure effects. This jump-augmented HJM model better captures the observed leptokurtosis and occasional large movements in interest rates while retaining the stylized features of the three-factor stochastic volatility framework.

The introduction of Poisson jumps into the stochastic-volatility HJM framework further accentuates the structural misspecification of the dynamic Nelson–Siegel (DL) benchmark and reinforces the relative strength of the Mamba architecture in out-of-sample forecasting.

At the short end of the curve (0.10 maturity), the difference remains dramatic. Mamba achieves MAE and RMSE around 0.002–0.003, while DL errors remain close to 0.093. The Diebold–Mariano rejection rate is 99.5%, with an extremely large negative mean DM statistic. The presence of discrete jumps—affecting the entire curve proportionally—induces abrupt, non-Gaussian movements that cannot be captured by the linear Gaussian AR(1) factor dynamics embedded in the DL specification. Since

Table 4: Monte Carlo out-of-sample results by maturity. DM rejection rate refers to the HAC-corrected Diebold–Mariano test at the 5% level.

	MAE (OUT)		**RMSE (OUT)**		**Forecast Comparison**			
τ	Mamba	DL	Mamba	DL	DM Reject	DL Sup.	Mamba Sup.	Mean DM
0.10	0.0021	0.0925	0.0025	0.0926	0.995	0.005	0.990	-77.849
0.25	0.0098	0.0149	0.0108	0.0168	0.720	0.220	0.500	-1.023
0.50	0.0119	0.0533	0.0134	0.0553	0.870	0.050	0.820	-7.480
1.00	0.0093	0.0477	0.0109	0.0501	0.895	0.015	0.880	-6.763
2.00	0.0081	0.0307	0.0099	0.0339	0.860	0.000	0.860	-4.026
5.00	0.0139	0.0466	0.0155	0.0490	0.840	0.100	0.740	-5.348
10.00	0.0140	0.0186	0.0152	0.0211	0.685	0.255	0.430	-0.120

DL assumes continuous, homoskedastic innovations, it systematically underreacts to jump realizations and propagates forecast errors across maturities.

Across medium maturities (0.50 to 5 years), Mamba continues to dominate strongly. MAE and RMSE differences remain large, and rejection rates typically lie between 84% and 89%. The hump-shaped volatility component, combined with stochastic volatility and discrete jumps, generates rich nonlinear interactions across maturities. The drift term in HJM already introduces maturity-integrated volatility effects; adding jumps scaled by the sum of factor volatilities creates synchronized, state-dependent shifts of the entire curve. This produces leptokurtic return distributions and time-varying higher moments that a linear three-factor representation cannot reproduce. The structured state-space neural architecture, by contrast, can flexibly approximate these nonlinear and non-Gaussian dynamics directly from past observations.

At 0.25 maturity, the advantage remains but is more moderate, with a 72% rejection rate and some fraction of replications favoring DL. This suggests that for very short maturities with relatively smaller effective jump amplification, the DL approximation occasionally performs competitively in finite samples. Nevertheless, the average DM statistic still favors Mamba.

At the long end (10 years), the gap narrows further. Errors remain lower for Mamba, but the rejection rate declines to roughly 69%, and the mean DM statistic is close to zero. This mirrors the previous stochastic-volatility case without jumps. Long-maturity dynamics are smoother due to the slowly decaying volatility loading and lower sensitivity to short-lived shocks. As a result, the effective dynamics become closer to a persistent low-dimensional factor structure, which DL can approximate more adequately.

Overall, the jump-augmented experiment strengthens the central conclusion. When the data-generating process departs from linear Gaussian factor dynamics—through stochastic volatility, correlated innovations, and especially discrete jumps—the dynamic Nelson–Siegel model becomes structurally restrictive. It cannot account for time-varying higher moments or abrupt regime-like movements. The Mamba state-space architecture, through nonlinear projections and recurrent structured dynamics, adapts to these features and delivers robust improvements in predictive accuracy. The persistence of large OOS gains even under jump risk indicates that flexible deep state-space representations are particularly well suited for environments characterized by leptokurtosis, volatility clustering, and synchronized curve-wide shocks, which are well-documented features of empirical interest rate data.

6.4 Affine Arbitrage-Free Three-Factor Model with Risk Premia

The data generating process is based on a three-factor affine term structure model (Duffie-Kan) simulated under the physical measure $\mathbb{P}$ with an affine risk premium. The factors $X_t \in \mathbb{R}^3$ evolve according to mean-reverting dynamics with drift matrix K_P and long-run levels θ_P, while the factor volatilities are given by the diagonal matrix Σ. The chosen drift parameters K_P and θ_P reflect plausible speed-of-mean-reversion and level targets for short, medium, and long-term components of the term structure, ensuring realistic factor paths. An affine risk premium is introduced through λ_0 and λ_1, which capture both level-dependent and state-independent compensation for bearing factor risk. These risk premia transform the physical measure dynamics into the risk-neutral measure $\mathbb{Q}$ via $K_Q = K_P + \Sigma\lambda_1$ and $\theta_Q = K_Q^{-1}(K_P\theta_P - \Sigma\lambda_0)$, ensuring arbitrage-free pricing of zero-coupon bonds. The short rate is

modeled as $r_t = \delta_0 + \delta_1^\top X_t$, with $\delta_0 = 0.01$ representing a baseline short rate and δ_1 assigning factor loadings to the short rate. The factors are simulated under $\mathbb{P}$ using Euler discretization, incorporating both the mean-reverting drift and the stochastic diffusion $\Sigma\, dW_t$. To obtain zero-coupon yields, the model solves the Riccati ordinary differential equations under $\mathbb{Q}$ for each maturity τ to compute $A(\tau)$ and $B(\tau)$, so that bond prices follow $P(t,\tau) = \exp(A(\tau) - B(\tau)^\top X_t)$. Yields are then obtained as $y(t,\tau) = -\log P(t,\tau)/\tau$, and small Gaussian noise with standard deviation 0.1% is added to mimic measurement or market microstructure effects. The parameters are chosen to generate realistic yield curve dynamics, with mean-reversion ensuring stability of factor paths, volatilities consistent with observed term structure movements, and the affine risk premia capturing the empirically observed excess returns on bonds of different maturities.

Table 5: Monte Carlo out-of-sample results by maturity. DM rejection rate refers to the HAC-corrected Diebold–Mariano test at the 5% level.

	MAE (OUT)		**RMSE (OUT)**		**Forecast Comparison**			
τ	Mamba	DL	Mamba	DL	DM Reject	DL Sup.	Mamba Sup.	Mean DM
0.10	0.0051	0.0278	0.0063	0.0301	0.910	0.000	0.910	-4.957
0.25	0.0049	0.0275	0.0061	0.0297	0.915	0.000	0.915	-5.008
0.50	0.0047	0.0271	0.0059	0.0291	0.915	0.000	0.915	-5.127
1.00	0.0043	0.0263	0.0054	0.0282	0.905	0.000	0.905	-5.347
2.00	0.0038	0.0251	0.0047	0.0267	0.930	0.000	0.930	-5.847
5.00	0.0028	0.0236	0.0035	0.0244	0.980	0.000	0.980	-7.887
10.00	0.0020	0.0229	0.0025	0.0234	0.990	0.000	0.990	-11.660

The Monte Carlo results under the three-factor affine term structure (Duffie–Kan) DGP provide a particularly informative benchmark because the data are generated by an arbitrage-free affine model with mean-reverting Gaussian factors and an affine risk premium. This environment is structurally closer to the dynamic Nelson–Siegel (DL) framework than the previous HJM specifications with stochastic volatility and jumps. Nevertheless, the out-of-sample evidence strongly favors the Mamba architecture across all maturities.

At the short end (0.10–0.50 years), Mamba achieves MAE values around 0.0047–0.0051, whereas DL errors are roughly five to six times larger. The Diebold–Mariano rejection rates exceed 90%, with consistently negative and economically large mean DM statistics. Even though the affine model is linear in the factors and Gaussian under the physical measure, the mapping from factors to yields is exponential-affine in bond prices and nonlinear in yields through the Riccati solutions. Moreover, the presence of an affine risk premium implies that physical-measure factor dynamics differ from risk-neutral pricing dynamics, creating richer cross-sectional and intertemporal interactions than those captured by a simple three-factor Nelson–Siegel representation.

The dominance of Mamba becomes even more pronounced at medium and long maturities. At 5 and 10 years, rejection rates reach 98% and 99%, respectively, with very large negative mean DM statistics. This pattern suggests that DL struggles particularly at the long end when the pricing kernels implied by the Riccati equations generate maturity-dependent loadings that are not constrained to the Nelson–Siegel functional form. Although both models are effectively three-factor representations, the affine model produces maturity loadings determined endogenously by the solution of the ODE system under the risk-neutral measure. These loadings generally differ from the fixed exponential-polynomial structure imposed by Nelson–Siegel, especially when risk premia alter the relationship between physical and pricing dynamics.

The results therefore highlight an important distinction: dimensional equivalence does not imply functional equivalence. Even with three latent factors, the affine structure generates yield dynamics that embed nonlinear cross-maturity relationships and state-dependent pricing effects. The DL model approximates the curve through a fixed parametric loading structure and independent AR(1) factor forecasts, which introduces approximation error and potentially misaligns the evolution of cross-sectional loadings over time.

By contrast, the Mamba state-space architecture directly learns the joint mapping from past yields to future yields without imposing a specific functional form for factor loadings. Its recurrent structured dynamics can approximate the mean-reverting Gaussian factor process, while its nonlinear projection

layers can emulate the exponential-affine bond pricing relationship implicit in the Riccati solution. The very high rejection frequencies and large improvements in MAE and RMSE—especially at longer maturities—indicate that this flexible approximation capacity translates into substantial predictive gains.

Overall, even in an arbitrage-free affine environment with Gaussian factors—arguably favorable to low-dimensional parametric models—the dynamic Nelson–Siegel specification exhibits meaningful structural mismatch relative to the true DGP. The Mamba architecture, by not restricting the cross-sectional loading structure and by jointly learning temporal and cross-sectional dynamics, delivers robust and economically significant out-of-sample improvements across the entire maturity spectrum.

6.5 Regime-Switching Affine Three-Factor Model

The regime-switching affine three-factor model extends the standard affine Duffie-Kan specification by allowing the parameters governing factor dynamics, volatilities, and risk premia to change according to a two-state Markov chain. The latent regime process $S_t \in \{0, 1\}$ evolves with transition probabilities $p_{00} = 0.95$ and $p_{11} = 0.90$, reflecting a high degree of persistence in both the low-volatility and high-volatility states, consistent with observed term structure regimes in empirical data. For each regime, the mean-reversion matrices $K_P[s]$, long-run levels $\theta_P[s]$, volatility matrices $\Sigma[s]$, and affine risk premia $\lambda_0[s], \lambda_1[s]$ are specified differently, allowing the factors to exhibit distinct dynamics under normal and stressed market conditions. These parameters are chosen to generate realistic shifts in the yield curve, with regime 0 representing a low-volatility environment and regime 1 a high-volatility environment, while preserving mean-reversion and stability of factor paths. Under each regime, the risk-neutral parameters $K_Q[s]$ and $\theta_Q[s]$ are derived via the affine risk premium specification, ensuring arbitrage-free pricing of zero-coupon bonds. The short rate retains a linear dependence on the factors, $r_t = \delta_0 + \delta_1^\top X_t$, maintaining continuity with the single-regime affine model. Factor paths are simulated under $\mathbb{P}$ conditional on the current regime, incorporating both the regime-specific drift and stochastic diffusion. Zero-coupon bond prices are obtained by solving the Riccati equations separately for each regime to compute $A(\tau)$ and $B(\tau)$, and yields are calculated as $y(t, \tau) = -\log P(t, \tau)/\tau$. Small Gaussian measurement noise is added to mimic market microstructure effects. This regime-switching extension captures observed changes in yield curve volatility and factor behavior across market states while retaining the analytical tractability and affine structure of the standard three-factor model.

Table 6: Monte Carlo out-of-sample results by maturity. DM rejection rate refers to the HAC-corrected Diebold–Mariano test at the 5% level.

	MAE (OUT)		**RMSE (OUT)**		**Forecast Comparison**			
τ	Mamba	DL	Mamba	DL	DM Reject	DL Sup.	Mamba Sup.	Mean DM
0.10	0.0062	0.0385	0.0078	0.0414	0.900	0.000	0.900	-5.081
0.25	0.0061	0.0381	0.0076	0.0409	0.905	0.000	0.905	-5.121
0.50	0.0059	0.0373	0.0073	0.0401	0.915	0.000	0.915	-5.187
1.00	0.0055	0.0360	0.0069	0.0387	0.915	0.000	0.915	-5.285
2.00	0.0051	0.0337	0.0065	0.0361	0.940	0.000	0.940	-5.466
5.00	0.0052	0.0286	0.0064	0.0305	0.930	0.000	0.930	-5.716
10.00	0.0066	0.0238	0.0077	0.0254	0.830	0.025	0.805	-5.595

The regime-switching affine three-factor experiment introduces a qualitatively different source of nonlinearity relative to the single-regime affine model: structural breaks in drift, volatility, and risk premia governed by a persistent latent Markov chain. This setting is particularly challenging for linear Gaussian factor models because the conditional distribution of future yields depends on an unobserved regime state that alters both the speed of mean reversion and the volatility structure.

The out-of-sample results indicate that Mamba maintains clear and systematic predictive superiority across all maturities. At the short end (0.10–0.50 years), MAE and RMSE for Mamba are roughly six times smaller than those of DL, and Diebold–Mariano rejection rates are around 90–92%. The negative mean DM statistics confirm that the gains are not occasional but persistent across Monte Carlo replications. In this regime-switching environment, DL effectively fits a single linear AR(1)-type

dynamic to factors that in reality follow state-dependent mean-reverting processes. As a result, it averages across regimes and fails to adapt promptly when volatility or drift parameters shift.

The advantage of Mamba becomes even more pronounced at medium maturities (1–5 years), where rejection rates rise to 93–94%. These maturities are particularly sensitive to changes in both drift and volatility parameters under different regimes, because the affine loadings implied by the Riccati equations depend on regime-specific risk-neutral dynamics. When the regime switches, the entire term structure loading pattern changes. A fixed Nelson–Siegel loading structure combined with constant AR(1) factor dynamics cannot replicate this endogenous shift in pricing kernels. In contrast, the Mamba state-space architecture can implicitly encode regime-dependent behavior through its nonlinear recurrent dynamics, effectively approximating a latent state without explicitly modeling a Markov chain.

At the long end (10 years), the rejection rate decreases to 83%, and DL occasionally performs competitively. This attenuation is consistent with the smoothing effect of long maturities: yields at the far end reflect long-horizon expectations and are less sensitive to short-lived regime switches. Nonetheless, Mamba still achieves lower MAE and RMSE on average, and the mean DM statistic remains strongly negative, indicating robust superiority.

Overall, the results demonstrate that introducing regime dependence significantly amplifies the structural limitations of single-regime parametric factor models. Although the underlying DGP remains affine within each regime and arbitrage-free, the presence of discrete shifts in drift, volatility, and risk premia generates nonlinear and state-dependent yield dynamics that cannot be captured by a time-invariant Nelson–Siegel specification. The Mamba architecture, by jointly learning cross-sectional and temporal relationships through nonlinear state-space representations, adapts to regime changes and delivers substantial and consistent out-of-sample improvements across the maturity spectrum.

7 Empirical Application

The empirical application uses U.S. Treasury Constant Maturity Treasury (CMT) rates, commonly referred to as Treasury par yield curve rates. These data represent interpolated yields derived from the U.S. Treasury's daily par yield curve and are designed to provide standardized yields at fixed maturities, even when no outstanding Treasury security matches those exact maturities.

The underlying par yield curve relates the yield of a Treasury security to its time to maturity and is constructed using closing market bid prices of the most recently auctioned Treasury securities traded in the over-the-counter market. The price quotations used in the construction are indicative bid-side quotes rather than transaction prices. These quotations are collected by the Federal Reserve Bank of New York at or near 3:30 PM (Eastern Time) on each trading day. Based on these inputs, the U.S. Treasury estimates a smooth par yield curve from which yields at constant maturities are extracted.

The CMT series currently provides yields at the following fixed maturities: 1, 1.5, 2, 3, 4, and 6 months, and 1, 2, 3, 5, 7, 10, 20, and 30 years. Because these rates are interpolated from the fitted par yield curve, they ensure consistency across maturities and over time. In the empirical application, we use daily U.S. Treasury Constant Maturity (CMT) yields at the following maturities: 1, 2, 3, and 6 months, and 1, 2, 3, 5, 7, 10, 20, and 30 years. This selection provides a balanced coverage of the short, medium, and long segments of the yield curve, allowing the models to capture both near-term monetary policy expectations and long-horizon term premia dynamics.

The sample period spans from January 2, 2019 to December 31, 2025, yielding a total of 1,750 daily observations. This period includes episodes of substantial market stress and regime variation, ensuring a rich environment for evaluating forecasting performance under changing macro-financial conditions.

For estimation and evaluation, the models are trained using the initial portion of the sample, while the final 60 observations are reserved for out-of-sample forecasting. These 60 days constitute the evaluation window, allowing for a direct comparison of predictive accuracy across competing specifications. This split ensures that model parameters are estimated exclusively on historical data, with forecasting performance assessed on genuinely unseen observations.

The empirical out-of-sample results, presented in Table 7, provide strong evidence in favor of the Mamba specification relative to the state-space Diebold–Li (DL) benchmark with estimated decay parameter λ. Across almost the entire maturity spectrum, Mamba delivers substantially lower MAE and RMSE, and the Diebold–Mariano (DM) statistics indicate statistically significant improvements in predictive accuracy.

Table 7: Out-of-sample forecast comparison: Mamba vs Diebold–Li (state space, λ estimated).

	MAE		**RMSE**		**Diebold–Mariano Test**	
τ	Mamba	DL	Mamba	DL	DM Stat	p-value
0.0833	0.0426	0.1748	0.0527	0.2268	-2.909	0.0036
0.1667	0.0296	0.1565	0.0399	0.2097	-2.973	0.0030
0.2500	0.0331	0.1796	0.0454	0.2341	-3.119	0.0018
0.5000	0.0301	0.2020	0.0362	0.2252	-4.194	0.0000
1.0000	0.0543	0.1808	0.0615	0.2004	-4.264	0.0000
2.0000	0.0384	0.1317	0.0459	0.1553	-3.907	0.0001
3.0000	0.0449	0.1164	0.0544	0.1337	-4.513	0.0000
5.0000	0.1203	0.1431	0.1301	0.1578	-2.116	0.0344
7.0000	0.1214	0.1454	0.1293	0.1571	-2.031	0.0423
10.0000	0.1074	0.1454	0.1159	0.1551	-2.624	0.0087
20.0000	0.0478	0.0889	0.0568	0.1022	-3.267	0.0011
30.0000	0.0497	0.0624	0.0598	0.0780	-1.188	0.2349

At the short end of the curve (1 to 6 months), the gains are particularly pronounced. For example, at the 1-month maturity (0.0833 years), the MAE of Mamba (0.0426) is roughly one quarter of that of DL (0.1748), with similar reductions in RMSE. The DM statistics are strongly negative and highly significant, with p-values well below 1%. This pattern persists at 2- and 3-month maturities, suggesting that Mamba is substantially better at capturing short-run dynamics, which are often driven by rapidly evolving monetary policy expectations and high-frequency macro-financial news.

In the medium segment (1 to 5 years), Mamba continues to dominate. At 1, 2, and 3 years, both MAE and RMSE are markedly lower under Mamba, and the DM test rejects equal predictive accuracy at very high confidence levels. Even at 5 and 7 years, where the gap narrows somewhat, Mamba still achieves statistically significant improvements at the 5% level. These results indicate that the flexible state-space neural architecture effectively captures both the cross-sectional structure and the intertemporal persistence of yields in the belly of the curve, where slope and curvature dynamics are most pronounced.

At longer maturities (10 and 20 years), the superiority of Mamba remains statistically significant. The reductions in RMSE are economically meaningful, and the DM p-values remain below conventional thresholds. This suggests that the model is not merely improving short-term dynamics but is also capturing longer-horizon expectations and term premia movements more accurately than the parametric DL structure.

Only at the 30-year maturity does the statistical advantage weaken. Although Mamba still achieves lower MAE and RMSE, the DM statistic is not statistically significant at conventional levels (p-value near 0.23). This attenuation at the very long end is consistent with the greater smoothness and persistence of ultra-long yields, which are often well approximated by low-dimensional factor structures. In such segments, the flexibility of Mamba yields smaller marginal gains relative to DL.

Overall, the empirical evidence aligns closely with the Monte Carlo findings. While the dynamic Nelson–Siegel framework provides a parsimonious and interpretable factor representation, its fixed functional loadings and linear Gaussian state dynamics impose structural restrictions. The Mamba architecture, by jointly learning nonlinear temporal and cross-sectional dependencies, translates this flexibility into consistent and statistically significant forecasting improvements across nearly all maturities in real-world Treasury yield data.

8 Fitting the yield curve with the no-arbitrage penalizations

In this example, we apply the proposed Mamba + No-Arbitrage B-Spline architecture to the U.S. Treasury Constant Maturity Treasury (CMT) rates analyzed in the forecasting experiment. The goal of this exercise is to illustrate the adjustment properties of the model and to analyze the role of the penalty parameters that control the structural restrictions embedded in the learning process.

The proposed Mamba + No-Arbitrage B-Spline architecture addresses this limitation by combining

three complementary components. First, the Mamba state-space module captures temporal dependencies and nonlinear dynamics in the sequence of observed yield curves. Second, the B-spline representation provides a flexible nonparametric approximation of the cross-sectional yield curve. Third, the no-arbitrage penalties impose economically meaningful constraints on the resulting curves. Together, these components allow the model to learn a highly flexible representation of the yield curve while preserving key financial principles such as smoothness and admissible forward rate behavior.

The relative importance of the regularization terms is controlled by the hyperparameters λ_{smooth} and λ_{fwd}. Rather than fixing these values arbitrarily, we determine them using Bayesian optimization.

Let $\boldsymbol{\lambda} = (\lambda_{\text{smooth}}, \lambda_{\text{fwd}})$. We consider the validation loss

$$\mathcal{J}(\boldsymbol{\lambda}) = \frac{1}{M} \sum_{t \in \mathcal{V}} \left\| y_{t+1} - \hat{y}_{t+1}(\boldsymbol{\lambda}) \right\|^2 , \tag{25}$$

where $\mathcal{V}$ denotes a validation sample. Bayesian optimization treats $\mathcal{J}(\boldsymbol{\lambda})$ as an unknown objective function and constructs a probabilistic surrogate model over the hyperparameter space. At each iteration, the algorithm selects new candidate values of $\boldsymbol{\lambda}$ by maximizing an acquisition function that balances exploration and exploitation.

This approach is particularly attractive in the present setting because the objective function is non-convex and expensive to evaluate, requiring full neural network training for each candidate pair $(\lambda_{\text{smooth}}, \lambda_{\text{fwd}})$.

In the empirical implementation, the dataset is divided into three parts. The first portion of the sample is used to train the neural network parameters. A validation subset is then used to evaluate the candidate penalization parameters during the Bayesian optimization procedure. The final segment of the sample is reserved for out-of-sample evaluation. This temporal division respects the time-series structure of the data and prevents look-ahead bias. During the Bayesian optimization stage, each candidate pair of hyperparameters is evaluated by training the model on the training sample and computing the validation loss on the validation subset. The surrogate model is then updated and a new candidate pair of hyperparameters is proposed.

The search is conducted over a logarithmic domain for both regularization parameters, allowing the algorithm to explore several orders of magnitude of penalization strength. The optimal parameters obtained from this procedure are

$$\lambda_{\text{smooth}} = 2.01 \times 10^{-6}, \qquad \lambda_{\text{fwd}} = 8.34 \times 10^{-6}.$$

The two penalization parameters have natural economic interpretations. The smoothness parameter λ_{smooth} controls the degree of cross-sectional regularization imposed on the yield curve. A larger value enforces smoother curves, effectively limiting the ability of the model to generate high-frequency oscillations across maturities. From an economic perspective, this reflects the well-established empirical observation that yields at nearby maturities tend to move in a coherent manner.

The forward-rate penalty parameter λ_{fwd} regulates the extent to which the model enforces economically admissible forward rate structures. Higher values impose stronger penalties on negative forward rates and thus encourage yield curves that are consistent with standard no-arbitrage intuition.

Together, these penalties allow the neural state-space model to remain flexible while embedding economically motivated structure into the learning process. Rather than imposing a rigid parametric form for the yield curve, the model learns the functional representation directly from the data while respecting smoothness and financial plausibility.

Figure 1 presents the yield surfaces obtained using the optimal penalization parameters. The model provides a close adjustment to the observed yield curves across all maturities. At the same time, the spline representation introduces a degree of smoothing relative to the raw observed data, resulting in a curve with fewer local irregularities.

This behavior reflects the trade-off between goodness-of-fit and smoothness imposed by the penalization structure. While the model is flexible enough to capture the main movements of the yield curve, the smoothness and no-arbitrage penalties prevent the network from fitting high-frequency noise that may arise from measurement errors or market microstructure effects. In this sense, the regularization terms also act as a mechanism to reduce overfitting and improve the stability of the estimated yield surface.

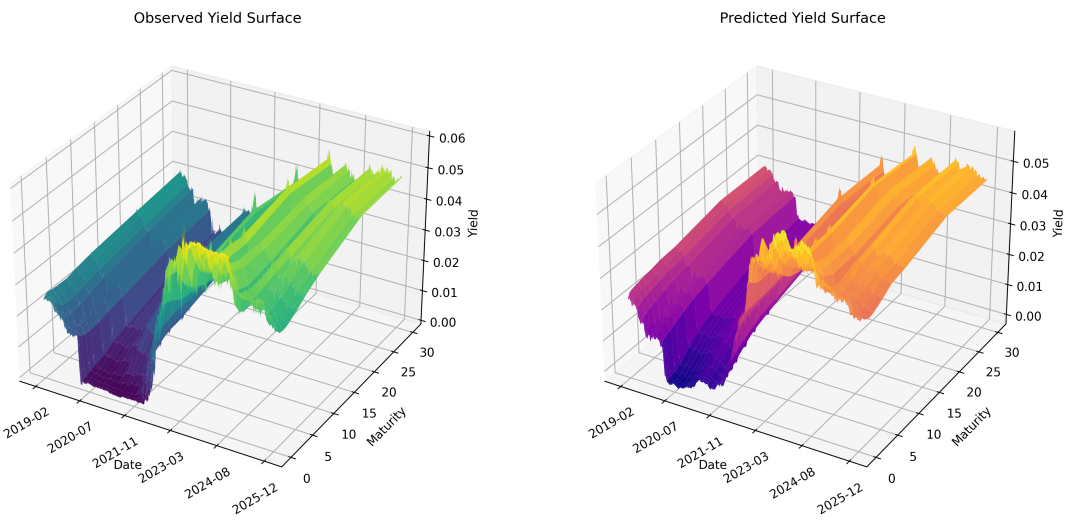

Figure 1: Observed and fitted yield surfaces obtained using the optimal penalization using Bayesian Optimization

From a broader methodological perspective, the proposed architecture can be interpreted as a Financial-Informed Neural Network (FINN). In contrast to purely data-driven neural models, FINNs incorporate domain-specific knowledge directly into the learning objective.

In the present case, financial theory provides a set of structural restrictions that characterize economically meaningful yield curves. These include smoothness across maturities and the absence of arbitrage opportunities implied by negative forward rates or non-monotonic discount factors. By embedding these restrictions as penalty terms in the loss function, the model integrates financial theory with the flexibility of modern neural architectures.

This perspective parallels the idea of Physics-Informed Neural Networks (PINNs), where physical laws guide the learning process through additional constraints. Similarly, the Mamba + No-Arbitrage B-Spline architecture incorporates financial equilibrium conditions to guide the estimation of the yield curve. The resulting framework combines the expressive power of deep learning with the theoretical discipline of arbitrage-free term structure modeling, producing estimates that are both accurate and economically interpretable.

9 Conclusion

This paper proposes a novel application of structured State Space Models (SSMs), implemented through the Mamba architecture, to the modeling and forecasting of the yield curve. The central objective is to assess whether a flexible, learnable state-space representation can improve predictive performance relative to traditional parametric term structure models, particularly the dynamic Nelson–Siegel (Diebold–Li) framework.

Our main contribution is threefold. First, from a modeling perspective, we introduce an SSM-based architecture capable of jointly learning temporal dynamics and cross-sectional relationships in the yield curve without imposing a fixed parametric loading structure. While classical affine and Nelson–Siegel models rely on predetermined functional forms for factor loadings and linear Gaussian state dynamics, the Mamba-based specification allows the data to determine both the effective state representation and the transition structure. This yields a unified framework that remains computationally tractable while being substantially more flexible.

The BSpline-Mamba Hybrid model can be interpreted as an finance-informed neural network. Similarly to Physics-Informed Neural Networks (PINNs), structural constraints derived from financial theory are incorporated directly into the loss function. In particular, no-arbitrage conditions such as monotonicity of the discount curve and positivity of forward rates are enforced through differentiable penalty terms. Rather than enforcing a governing partial differential equation as in classical PINNs, the

model imposes economically motivated shape constraints on the term structure representation. This approach allows the network to learn flexible yield curve dynamics while ensuring that the resulting curves remain consistent with fundamental arbitrage-free conditions.

Second, from a methodological standpoint, we design a comprehensive Monte Carlo evaluation strategy that compares forecasting performance under increasingly complex data-generating processes. These include a standard three-factor affine (Duffie–Kan) model, as well as extensions with regime switching, stochastic volatility, and nonlinear features. In each case, the Diebold–Li model serves as a structured parametric benchmark. The simulation design allows us to isolate the impact of nonlinearities, regime changes, and risk-premium dynamics on forecasting performance, thereby providing a controlled environment to evaluate structural robustness.

Third, we provide an empirical application using U.S. Treasury Constant Maturity yields over the 2019–2025 period. The empirical results confirm the simulation evidence: the Mamba architecture delivers systematic and statistically significant out-of-sample improvements in MAE and RMSE across nearly all maturities. The gains are particularly strong at short and medium maturities, where time variation, nonlinear interactions, and policy-driven dynamics are most pronounced, while remaining robust at longer maturities.

In addition to forecasting performance, we also investigate the in-sample adjustment properties of the proposed framework by combining the Mamba state-space architecture with a B-spline representation of the yield curve and economically motivated penalization terms. The B-spline layer provides a flexible nonparametric representation of the cross-sectional structure of the yield curve, while the smoothness and forward-rate penalties impose economically meaningful restrictions related to curve regularity and the absence of arbitrage opportunities. The strength of these penalization terms is determined using Bayesian optimization, allowing the model to balance goodness-of-fit with economically plausible curve shapes.

The empirical adjustment results show that the model closely tracks the observed yield curves across maturities while producing smoother surfaces than the raw data. In particular, the penalization terms reduce local irregularities and oscillations in the fitted curves, leading to a more stable representation of the term structure. This behavior reflects an important bias–variance trade-off: while the neural architecture retains sufficient flexibility to capture the main movements in the yield curve, the regularization terms prevent the model from fitting high-frequency noise that may arise from measurement errors or market microstructure effects. As a result, the penalized specification improves the robustness and interpretability of the estimated yield surface.

Across all experiments, a consistent pattern emerges. When the true data-generating process deviates from linear, time-invariant Gaussian factor dynamics—through regime switching, nonlinear pricing relationships, or richer volatility structures—the parametric restrictions of the Nelson–Siegel framework induce structural approximation error. Even when the DGP is affine and arbitrage-free, the fixed functional loading structure of Nelson–Siegel cannot perfectly replicate the maturity-dependent loadings implied by the Riccati system. In contrast, the SSM/Mamba architecture adapts flexibly to these features and translates representational capacity into forecasting gains.

More broadly, the combination of neural sequence modeling with structural penalization can be interpreted as an instance of a Financial-Informed Neural Network (FINN). Rather than relying exclusively on data-driven learning, the model incorporates domain knowledge from financial economics directly into the estimation objective. In this case, smoothness and no-arbitrage restrictions act as guiding principles that shape the learned representation of the yield curve. This approach parallels the logic of physics-informed neural networks, where governing laws are embedded into the learning process to improve generalization and ensure consistency with theoretical principles.

Importantly, this paper focuses primarily on forecasting performance and empirical adjustment properties. We do not attempt to provide a structural economic interpretation of the latent states learned by the SSM, nor do we derive explicit analytical representations for the implied factor loadings across maturities. Understanding how the learned internal states relate to traditional level, slope, and curvature factors—or whether they embed richer term-premium components—remains an open research question. Developing approximation results that connect SSM/Mamba representations to classical affine loadings is a promising direction for future work.

Moreover, we restrict attention to predictive accuracy and do not explore asset pricing or hedging implications. The potential use of SSM-based architectures in pricing fixed-income securities, extracting risk premia, constructing hedging portfolios, or performing scenario analysis under alterna-

tive measures (physical versus risk-neutral) represents a natural extension. Embedding no-arbitrage restrictions directly into the SSM framework, or combining it with pricing-kernel approaches, may further bridge the gap between flexible machine learning models and structural term structure theory.

In summary, the evidence suggests that structured neural state-space models offer a powerful and flexible alternative to traditional parametric yield curve models for forecasting purposes. At the same time, their theoretical interpretation and integration into pricing and risk-management applications remain open and promising avenues for future research.

References

Almeida, C., K. Ardison, D. Kubudi, A. Simonsen, and J. Vicente (2017, 02). Forecasting bond yields with segmented term structure models*. *Journal of Financial Econometrics 16*(1), 1–33.

Ang, A. and G. Bekaert (2002). Regime switches in interest rates. *Journal of Business & Economic Statistics 20*(2), 163–182.

Brigo, D. and F. Mercurio (2006, April). *Interest Rate Models - Theory and Practice*. Number 978-3-540-34604-3 in Springer Finance. Springer.

Chen, R. T. Q., Y. Rubanova, J. Bettencourt, and D. Duvenaud (2019). Neural ordinary differential equations.

Cheridito, P., D. Filipović, and R. Kimmel (2007). Market price of risk specifications for affine models: Theory and evidence. *Journal of Financial Economics 83*(1), 123–170.

Christensen, J. H. E., F. X. Diebold, and G. D. Rudebusch (2011). The affine arbitrage-free class of nelson–siegel term structure models. *Journal of Econometrics 164*(1), 4–20.

Dai, Q. and K. J. Singleton (2000). Specification analysis of affine term structure models. *Journal of Finance 55*(5), 1943–1978.

Dai, Q., K. J. Singleton, and W. Yang (2007). Regime shifts in a dynamic term structure model of u.s. treasury bond yields. *Review of Financial Studies 20*(5), 1669–1706.

Dao, T. and A. Gu (2024). Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning (ICML)*.

Diebold, F. X. and C. Li (2006). Forecasting the term structure of government bond yields. *Journal of Econometrics 130*(2), 337–364.

Diebold, F. X. and G. D. Rudebusch (2013). *Yield Curve Modeling and Forecasting*. Princeton, NJ: Princeton University Press.

Diebold, F. X., G. D. Rudebusch, and S. Borag˘an Aruoba (2006). The macroeconomy and the yield curve: a dynamic latent factor approach. *Journal of Econometrics 131*(1), 309–338.

Duffee, G. R. (2002). Term premia and interest rate forecasts in affine models. *Journal of Finance 57*(1), 405–443.

Duffie, D. and R. Kan (1996). A yield-factor model of interest rates. *Mathematical Finance 6*(4), 379–406.

Durbin, J. and S. J. Koopman (2012). *Time Series Analysis by State Space Methods* (2 ed.). Oxford: Oxford University Press.

Filipović, D. (2009). *Term-Structure Models: A Graduate Course*. Springer Finance. Berlin, Heidelberg: Springer.

Fisher, M., D. Nychka, and D. Zervos (1995). Fitting the term structure of interest rates with smoothing splines. Finance and Economics Discussion Series 95-1, Board of Governors of the Federal Reserve System (U.S.).

Gu, A. and T. Dao (2024). Mamba: Linear-time sequence modeling with selective state spaces.

Gu, A., K. Goel, and C. Ré (2022). Efficiently modeling long sequences with structured state spaces.

Gu, A., I. Johnson, K. Goel, K. Saab, T. Dao, A. Rudra, and C. Ré (2020). Hippo: Recurrent memory with optimal polynomial projections. *Advances in Neural Information Processing Systems*.

Heath, D., R. Jarrow, and A. Morton (1992). Bond pricing and the term structure of interest rates: A new methodology for contingent claims valuation. *Econometrica 60*(1), 77–105.

Hochreiter, S. and J. Schmidhuber (1997, November). Long short-term memory. *Neural Comput. 9*(8), 1735–1780.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering 82*(1), 35–45.

Kowal, D. R., D. S. Matteson, and D. Ruppert (2019). Functional autoregression for sparsely sampled data. *Journal of Business & Economic Statistics 37*(1), 97–109.

Krishnan, R. G., U. Shalit, and D. Sontag (2015). Deep kalman filters.

Laurini, M. P. (2014). Dynamic functional data analysis with non-parametric state space models. *Journal of Applied Statistics 41*(1), 142–163.

Laurini, M. P. and L. K. Hotta (2010). Bayesian extensions to diebold-li term structure model. *International Review of Financial Analysis 19*(5), 342–350.

Laurini, M. P. and L. K. Hotta (2014). Forecasting the term structure of interest rates using integrated nested laplace approximations. *Journal of Forecasting 33*(3), 214–230.

Laurini, M. P. and M. Moura (2010). Constrained smoothing b-splines for the term structure of interest rates. *Insurance: Mathematics and Economics 46*(2), 339–350.

Lecun, Y., L. Bottou, Y. Bengio, and P. Haffner (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*(11), 2278–2324.

McCulloch, J. H. (1971). Measuring the term structure of interest rates. *Journal of Business 44*(1), 19–31.

Musiela, M. and M. Rutkowski (2005). *Martingale Methods in Financial Modelling* (2 ed.). Berlin: Springer.

Nelson, C. and A. F. Siegel (1987). Parsimonious modeling of yield curves. *The Journal of Business 60*(4), 473–89.

Raissi, M., P. Perdikaris, and G. Karniadakis (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics 378*, 686–707.

Rangapuram, S. S., M. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski (2018). Deep state space models for time series forecasting. In *Advances in Neural Information Processing Systems*.

Shen, Z. and D. Xiu (2024, December). Deep autoencoders for nonlinear factor models: Theory and applications. Chicago Booth Research Paper 25-14, University of Chicago, Booth School of Business. Available at SSRN.

Stanton, R. (1997). A nonparametric model of term structure dynamics and the market price of interest rate risk. *The Journal of Finance 52*(5), 1973–2002.

Svensson, L. E. O. (1994). Estimating and interpreting forward interest rates: Sweden 1992–1994. *IMF Working Paper* (94/114).