Self-driving neural networks for yield curve modeling

Sicco Kooiker, Janneke van Brummelen, Julia Schaumburg, Marcin Zamojski

March, 2026

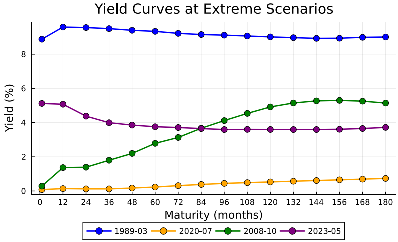
Vrije Universiteit Amsterdam

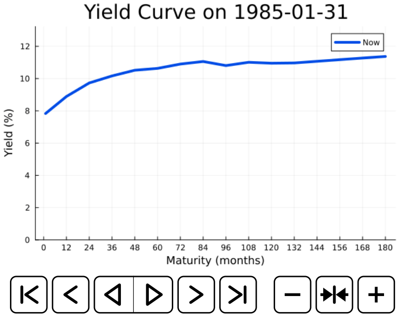
The Yield Curve

- The yield curve shows the relationship between interest rates and the maturities of government bonds.
- The yield curve is a central object in monetary policy and financial markets.
- It is often interpreted as a summary of:
  - growth expectations,
  - inflation expectations,
  - and risk premia.

Yield Curve Forecasting of US Treasury Bonds

- It often slopes upward, reflecting higher yields for longer maturities, but can flatten or invert in response to changing economic expectations.





Yield Curve Forecasting

- The workhorse models are Nelson–Siegel (NS) three-factor models (e.g. Nelson and Siegel (1987); Diebold and Li (2006); Diebold et al. (2006); Christensen et al. (2011)).
- They summarize the entire curve using three intuitive components:
  - Level
  - Slope
  - Curvature
- This structure is simple and interpretable.
- But it imposes a strong restriction:
  - the functional shapes of the components are fixed.

Contributions

1. We learn the shape of slope and curvature using neural networks within a factor model. (Unlike less interpretable machine learning approaches (e.g. Stock and Watson (2002); Bai and Ng (2013); Bianchi et al. (2021)).)

- The yield curve does not only move along factors — it reshapes:
  - Time-varying NS (e.g. Koopman et al. (2010); Hevia et al. (2014); Quaedvlieg and Schotman (2016))

2. We go from static neural networks to neural networks with observation-driven parameters.

3. This leads to the Self-Driving Neural Nelson–Siegel (SD-NNS) model, which improves forecast performance for the US yield curve.

Yield Curve Factor Models

Yield Curve Factor Models

- Yield curve factor models

$$\boldsymbol{y}_t = \Gamma(\boldsymbol{\tau}; \gamma_t)\, \boldsymbol{\beta}_t + \boldsymbol{\varepsilon}_t,$$

where

$$\begin{aligned}
\boldsymbol{y}_t &= \{y_{1t}, \ldots, y_{Mt}\}', \\
\boldsymbol{\varepsilon}_t &= \{\varepsilon_{1t}, \ldots, \varepsilon_{Mt}\}', \\
\boldsymbol{\tau} &= \{\tau_1, \ldots, \tau_M\}', \\
\Gamma(\boldsymbol{\tau}; \boldsymbol{\gamma}_t) &\in \mathbb{R}^{M \times K}, \quad K \ll M, \\
\boldsymbol{\beta}_t &\in \mathbb{R}^K.
\end{aligned}$$

- y_{it} is the yield corresponding to time-to-maturity τ_i, for $i = 1, \ldots, M$.
- We assume serially uncorrelated innovations with equal cross-sectional variance.

Specification of Factor Loadings

- Nelson-Siegel Model (NS) :

$$\Gamma_{ij}^{NS}(\tau_i;\lambda) = \begin{cases} 1, & j = 1, \\ \frac{1-e^{-\lambda\,\tau_i}}{\lambda\,\tau_i}, & j = 2, \\ \frac{1-e^{-\lambda\,\tau_i}}{\lambda\,\tau_i} - e^{-\lambda\,\tau_i}, & j = 3, \end{cases}$$

- These factor loading specifications make the factors interpretable as the: level, slope, and curvature.
- We want to be even more flexible.

Specification of Factor Loadings

- Nelson-Siegel Model (NS) :

$$\Gamma_{ij}^{NS}(\tau_i;\lambda) = \begin{cases} 1, & j = 1, \\ \frac{1-e^{-\lambda\tau_i}}{\lambda\tau_i}, & j = 2, \\ \frac{1-e^{-\lambda\tau_i}}{\lambda\tau_i} - e^{-\lambda\tau_i}, & j = 3, \end{cases}$$

- These factor loading specifications make the factors interpretable as the: level, slope, and curvature.
- We want to be even more flexible.

Specification of Factor Loadings

- Nelson-Siegel Model (NS) :

$$\Gamma_{ij}^{NS}(\tau_i;\lambda) = \begin{cases} 1, & j = 1, \\ \frac{1-e^{-\lambda\,\tau_i}}{\lambda\,\tau_i}, & j = 2, \\ \frac{1-e^{-\lambda\,\tau_i}}{\lambda\,\tau_i} - e^{-\lambda\,\tau_i}, & j = 3, \end{cases}$$

- These factor loading specifications make the factors interpretable as the: level, slope, and curvature.
- We want to be even more flexible.

Neural Nelson-Siegel Model

- Our proposed model: Neural Nelson-Siegel Model (NNS) with *tiny* one-to-one neural networks (NN) as factor loadings.

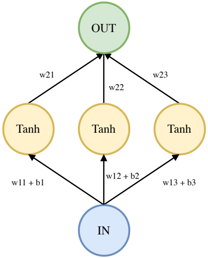
$$\Gamma_{ij}(\tau_i;\gamma_t) = \begin{cases} 1, & j=1, \\ \mathrm{NN}(\tau_i;\gamma_{1t}), & j=2, \\ \mathrm{NN}(\tau_i;\gamma_{2t}), & j=3, \end{cases} \qquad \gamma_t = \{\gamma_{1t}', \gamma_{2t}'\}'$$

where

$$\mathrm{NN}(\tau;\gamma) = \sum_{l=1}^{L} w_{2l}\, \phi\big(w_{1l}\,\tau + b_l\big),$$

$$\gamma = \{\, w_{1l},\ w_{2l},\ b_l\}_{l=1}^{L},$$

$$\phi : \mathbb{R} \to A, A \subseteq \mathbb{R}$$



Self-driving time-varying parameters

- We let the parameters of the neural networks be driven by the gradient of the Mean Squared Error (MSE), following the ideas in Creal et al. (2024).
- This is similar to the score-driven literature on Generalized Autoregressive Score models Creal et al. (2013). Instead of a log-likelihood, we use MSE.
- This way, we do not have to assume a likelihood and we are more robust against misspecification.

Identification and Interpretability

Identification and Interpretability

- Two goals:
  1. Unique identification of β factors. (*factor* $\times \mathrm{NN}(x)$ is not identified.)
  2. Interpretable factors

- Similar to NS, we want the three latent factors $\boldsymbol{\beta}_t$ to be associated with the *level*, *slope* and *curvature*.
- We use a combinations of anchoring and normalization to fix the location and scale of the neural networks.
- β_{1t}, $-\beta_{2t}$, β_{3t}, like Nelson-Siegel, are associated with the level, slope, and curvature.

Self-Driving Filter

Self-Driving Filter

- Initialize $\widehat{\gamma}_1$ and $\widehat{\boldsymbol{\beta}}_1$ at $t = 0$.

Self-Driving Filter

- Initialize $\widehat{\gamma}_1$ and $\widehat{\boldsymbol{\beta}}_1$ at $t = 0$.
- At $t = 1$, $\boldsymbol{y}_1$ enters, how do we update?

Self-Driving Filter

- Initialize $\widehat{\gamma}_1$ and $\widehat{\boldsymbol{\beta}}_1$ at $t = 0$.
- At $t = 1$, $\boldsymbol{y}_1$ enters, how do we update?
- For $1, \ldots, T$ do:

  UPDATE:

  PREDICT:

Self-Driving Filter

- Initialize $\widehat{\gamma}_1$ and $\widehat{\boldsymbol{\beta}}_1$ at $t = 0$.
- At $t = 1$, $\boldsymbol{y}_1$ enters, how do we update?
- For $1, \ldots, T$ do:

  UPDATE:

  1. $\widehat{\gamma}_{t|t} = \widehat{\gamma}_t + A\nabla_t$

  PREDICT:

Self-Driving Filter

- Initialize $\widehat{\gamma}_1$ and $\widehat{\boldsymbol{\beta}}_1$ at $t = 0$.
- At $t = 1$, $\boldsymbol{y}_1$ enters, how do we update?
- For $1, \dots, T$ do:

  UPDATE:

  1. $\widehat{\gamma}_{t|t} = \widehat{\gamma}_t + A\nabla_t$
  2. $\widehat{\gamma}_{t|t} \Rightarrow \Gamma(\boldsymbol{\tau}; \widehat{\gamma}_{t|t}) \Rightarrow \widehat{\Gamma}_{t|t}$

  PREDICT:

Self-Driving Filter

- Initialize $\widehat{\gamma}_1$ and $\widehat{\boldsymbol{\beta}}_1$ at $t = 0$.
- At $t = 1$, $\boldsymbol{y}_1$ enters, how do we update?
- For $1, \dots, T$ do:

  UPDATE:

  1. $\widehat{\gamma}_{t|t} = \widehat{\gamma}_t + A\nabla_t$
  2. $\widehat{\gamma}_{t|t} \Rightarrow \Gamma(\boldsymbol{\tau}; \widehat{\gamma}_{t|t}) \Rightarrow \widehat{\Gamma}_{t|t}$
  3. $\widehat{\boldsymbol{\beta}}_{t|t} = (\widehat{\Gamma}'_{t|t}\widehat{\Gamma}_{t|t})^{-1}\widehat{\Gamma}'_{t|t}\boldsymbol{y}_t$

  PREDICT:

Self-Driving Filter

- Initialize $\widehat{\gamma}_1$ and $\widehat{\boldsymbol{\beta}}_1$ at $t = 0$.
- At $t = 1$, $\boldsymbol{y}_1$ enters, how do we update?
- For $1, \dots, T$ do:

  UPDATE:

  1. $\widehat{\gamma}_{t|t} = \widehat{\gamma}_t + A\nabla_t$
  2. $\widehat{\gamma}_{t|t} \Rightarrow \Gamma(\boldsymbol{\tau}; \widehat{\gamma}_{t|t}) \Rightarrow \widehat{\Gamma}_{t|t}$
  3. $\widehat{\boldsymbol{\beta}}_{t|t} = (\widehat{\Gamma}'_{t|t}\widehat{\Gamma}_{t|t})^{-1}\widehat{\Gamma}'_{t|t}\boldsymbol{y}_t$

  PREDICT:

  1. $\widehat{\gamma}_{t+1} = \omega_\gamma + B_\gamma \widehat{\gamma}_{t|t}, \quad \widehat{\boldsymbol{\beta}}_{t+1} = \omega_\beta + B_\beta \widehat{\boldsymbol{\beta}}_{t|t}$

Self-Driving Filter

- Initialize $\widehat{\gamma}_1$ and $\widehat{\boldsymbol{\beta}}_1$ at $t = 0$.
- At $t = 1$, $\boldsymbol{y}_1$ enters, how do we update?
- For $1, \ldots, T$ do:

  UPDATE:

  1. $\widehat{\gamma}_{t|t} = \widehat{\gamma}_t + A\nabla_t$
  2. $\widehat{\gamma}_{t|t} \Rightarrow \Gamma(\boldsymbol{\tau}; \widehat{\gamma}_{t|t}) \Rightarrow \widehat{\Gamma}_{t|t}$
  3. $\widehat{\boldsymbol{\beta}}_{t|t} = (\widehat{\Gamma}'_{t|t}\widehat{\Gamma}_{t|t})^{-1}\widehat{\Gamma}'_{t|t}\boldsymbol{y}_t$

  PREDICT:

  1. $\widehat{\gamma}_{t+1} = \omega_\gamma + B_\gamma \widehat{\gamma}_{t|t}, \quad \widehat{\boldsymbol{\beta}}_{t+1} = \omega_\beta + B_\beta \widehat{\boldsymbol{\beta}}_{t|t}$
  2. $\widehat{\boldsymbol{y}}_{t+1} = \Gamma(\boldsymbol{\tau}; \widehat{\gamma}_{t+1})\widehat{\boldsymbol{\beta}}_{t+1}$

Self-Driving Filter

- Initialize $\widehat{\gamma}_1$ and $\widehat{\boldsymbol{\beta}}_1$ at $t = 0$.
- At $t = 1$, $\boldsymbol{y}_1$ enters, how do we update?
- For $1, \dots, T$ do:

  UPDATE:

  1. $\widehat{\gamma}_{t|t} = \widehat{\gamma}_t + A\nabla_t$
  2. $\widehat{\gamma}_{t|t} \Rightarrow \Gamma(\boldsymbol{\tau}; \widehat{\gamma}_{t|t}) \Rightarrow \widehat{\Gamma}_{t|t}$
  3. $\widehat{\boldsymbol{\beta}}_{t|t} = (\widehat{\Gamma}'_{t|t}\widehat{\Gamma}_{t|t})^{-1}\widehat{\Gamma}'_{t|t}\boldsymbol{y}_t$

  PREDICT:

  1. $\widehat{\gamma}_{t+1} = \omega_\gamma + B_\gamma \widehat{\gamma}_{t|t}, \quad \widehat{\boldsymbol{\beta}}_{t+1} = \omega_\beta + B_\beta \widehat{\boldsymbol{\beta}}_{t|t}$
  2. $\widehat{\boldsymbol{y}}_{t+1} = \Gamma(\boldsymbol{\tau}; \widehat{\gamma}_{t+1})\widehat{\boldsymbol{\beta}}_{t+1}$

- For *h*-step-ahead forecast, skip UPDATE step and recursively PREDICT *h* times.

Estimation

- We optimize the static parameters of the filter by minimizing the MSE:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\in\Theta}{\arg\min} \sum_{t=1}^{T} (\boldsymbol{y}_t - \widehat{\boldsymbol{y}}_t)'(\boldsymbol{y}_t - \widehat{\boldsymbol{y}}_t)$$

$$\widehat{\boldsymbol{y}}_t = \widehat{\Gamma}_t \widehat{\boldsymbol{\beta}}_t$$

$$\boldsymbol{\theta} = \{\text{vec}(A), \omega_\gamma', \text{vec}(B_\gamma), \omega_\beta', \text{vec}(B_\beta)\}'$$

- We impose restrictions on A and B.
  - SD-NNS: A and B_γ are diagonal.
  - NNS: A and B_γ equal $\mathbf{0}$.
- We scale gradient updates using an adaptive second-moment scheme (standard in neural network optimization), denoted **S**SD-NNS.

Estimation

- We optimize the static parameters of the filter by minimizing the MSE:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\in\Theta}{\arg\min} \sum_{t=1}^{T} (\boldsymbol{y}_t - \widehat{\boldsymbol{y}}_t)'(\boldsymbol{y}_t - \widehat{\boldsymbol{y}}_t)$$

$$\widehat{\boldsymbol{y}}_t = \widehat{\Gamma}_t \widehat{\boldsymbol{\beta}}_t$$

$$\boldsymbol{\theta} = \{\text{vec}(A), \omega'_\gamma, \text{vec}(B_\gamma), \omega'_\beta, \text{vec}(B_\beta)\}'$$

- We impose restrictions on A and B.
  - SD-NNS: A and B_γ are diagonal.
  - NNS: A and B_γ equal $\mathbf{0}$.
- We scale gradient updates using an adaptive second-moment scheme (standard in neural network optimization), denoted **S**SD-NNS.

Estimation

- We optimize the static parameters of the filter by minimizing the MSE:

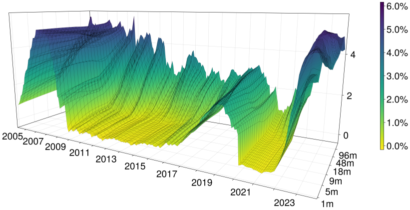
$$
\begin{aligned}
\widehat{\boldsymbol{\theta}} &= \underset{\boldsymbol{\theta}\in\Theta}{\arg\min} \sum_{t=1}^{T} (\boldsymbol{y}_t - \widehat{\boldsymbol{y}}_t)'(\boldsymbol{y}_t - \widehat{\boldsymbol{y}}_t) \\
\widehat{\boldsymbol{y}}_t &= \widehat{\Gamma}_t \widehat{\boldsymbol{\beta}}_t \\
\boldsymbol{\theta} &= \{\text{vec}(A), \omega_\gamma', \text{vec}(B_\gamma), \omega_\beta', \text{vec}(B_\beta)\}'
\end{aligned}
$$

- We impose restrictions on A and B.
  - SD-NNS: A and B_γ are diagonal.
  - NNS: A and B_γ equal $\mathbf{0}$.
- We scale gradient updates using an adaptive second-moment scheme (standard in neural network optimization), denoted **S**SD-NNS.

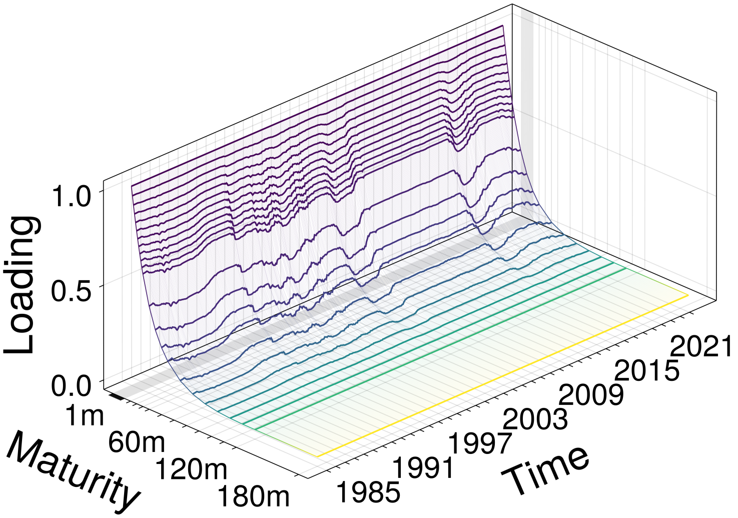
US Yield Curve Forecasting

Outline of Experiment

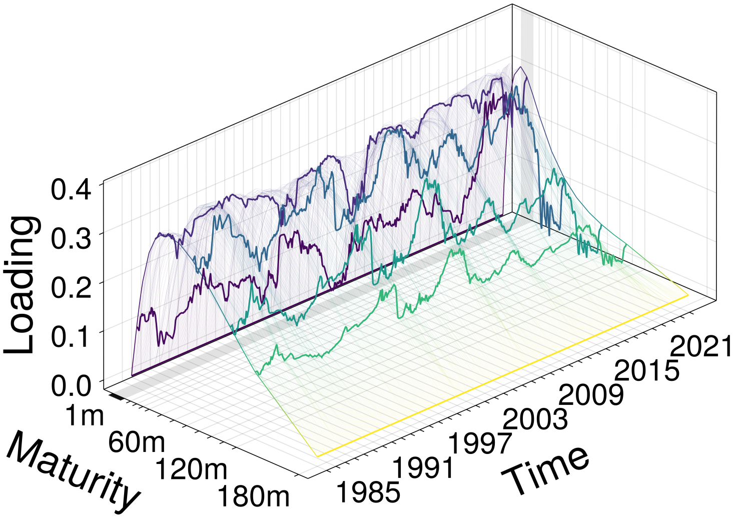
- Yield Curve Forecasting
  - Monthly US yield curve data (Liu and Wu, 2021)
  - Horizons: 1 till 12 months
  - In-sample: 1985-01 - 2004-02
  - Out-of-sample: 2004-03 - 2024-12
- Moving window estimation



SSD-NNS Loadings Estimated till 2024 (1/2)



SSD-NNS, the loadings on the slope factor.



SSD-NNS, the loadings on the curvature factor.

Relative Root Mean Squared Error

Out-of-Sample Relative RMSE (Model RMSE / RW RMSE) for 1-Month-Ahead Forecast Horizon - Moving Window

Model	m-3	m-6	m-12	m-24	m-60	m-120	Average
RW	1.000	1.000	1.000	1.000	1.000	1.000	1.000
NS	0.900**	0.966	1.060	1.035**	1.065**	0.989	1.010
NNS	0.940	0.946	1.044	1.032	1.063**	1.029	1.018
SSD-NS	0.918***	0.927*	0.960	1.022	1.054*	0.984	0.988*
SSD-NNS	0.911**	0.933**	0.954**	1.018	1.018	0.993	0.980**
SSD Ensemble†	0.907***	0.915***	0.955*	1.018	1.042	0.975	0.979**

Asterisks denote Diebold–Mariano rejection of equal RMSE with the RW benchmark (* 10%; ** 5%; *** 1%).

† : Equally weighted ensemble between SSD-NS and SSD-NNS

Relative Root Mean Squared Error

Out-of-Sample Relative RMSE (Model RMSE / RW RMSE) for 1-Month-Ahead Forecast Horizon - Moving Window

Model	m-3	m-6	m-12	m-24	m-60	m-120	Average
RW	1.000	1.000	1.000	1.000	1.000	1.000	1.000
NS	0.900**	0.966	1.060	1.035**	1.065**	0.989	1.010
NNS	0.940	0.946	1.044	1.032	1.063**	1.029	1.018
SSD-NS	0.918***	0.927*	0.960	1.022	1.054*	0.984	0.988*
SSD-NNS	0.911**	0.933**	0.954**	1.018	1.018	0.993	0.980**
SSD Ensemble†	0.907***	0.915***	0.955*	1.018	1.042	0.975	0.979**

Asterisks denote Diebold–Mariano rejection of equal RMSE with the RW benchmark (* 10%; ** 5%; *** 1%).

† : Equally weighted ensemble between SSD-NS and SSD-NNS

Relative Root Mean Squared Error

Out-of-Sample Relative RMSE (Model RMSE / RW RMSE) for 1-Month-Ahead Forecast Horizon - Moving Window

Model	m-3	m-6	m-12	m-24	m-60	m-120	Average
RW	1.000	1.000	1.000	1.000	1.000	1.000	1.000
NS	0.900**	0.966	1.060	1.035**	1.065**	0.989	1.010
NNS	0.940	0.946	1.044	1.032	1.063**	1.029	1.018
SSD-NS	0.918***	0.927*	0.960	1.022	1.054*	0.984	0.988*
SSD-NNS	0.911**	0.933**	0.954**	1.018	1.018	0.993	0.980**
SSD Ensemble†	0.907***	0.915***	0.955*	1.018	1.042	0.975	0.979**

Asterisks denote Diebold–Mariano rejection of equal RMSE with the RW benchmark (* 10%; ** 5%; *** 1%).

† : Equally weighted ensemble between SSD-NS and SSD-NNS

Relative Root Mean Squared Error

Out-of-Sample Relative RMSE (Model RMSE / RW RMSE) for 1-Month-Ahead Forecast Horizon - Moving Window

Model	m-3	m-6	m-12	m-24	m-60	m-120	Average
RW	1.000	1.000	1.000	1.000	1.000	1.000	1.000
NS	0.900**	0.966	1.060	1.035**	1.065**	0.989	1.010
NNS	0.940	0.946	1.044	1.032	1.063**	1.029	1.018
SSD-NS	0.918***	0.927*	0.960	1.022	1.054*	0.984	0.988*
SSD-NNS	0.911**	0.933**	0.954**	1.018	1.018	0.993	0.980**
SSD Ensemble†	0.907***	0.915***	0.955*	1.018	1.042	0.975	0.979**

Asterisks denote Diebold–Mariano rejection of equal RMSE with the RW benchmark (* 10%; ** 5%; *** 1%).

† : Equally weighted ensemble between SSD-NS and SSD-NNS

Relative Root Mean Squared Error

Out-of-Sample Relative RMSE (Model RMSE / RW RMSE) for 12-Month-Ahead Forecast Horizon - Moving Window

Model	m-3	m-6	m-12	m-24	m-60	m-120	Average
RW	1.000	1.000	1.000	1.000	1.000	1.000	1.000
NS	0.866	0.893	0.935	1.027	1.092	1.032	0.952
NNS	0.872	0.883	0.902	0.949	0.980	0.982	0.914**
SSD-NS	0.863	0.888	0.932	1.03	1.136	1.057	0.958
SSD-NNS	0.858	0.880	0.905	0.943	0.956	0.960	0.905**
SSD Ensemble	0.844	0.870	0.907	0.970	1.008	0.953	0.910**

Asterisks denote Diebold–Mariano rejection of equal RMSE with the RW benchmark (* 10%; ** 5%; *** 1%).

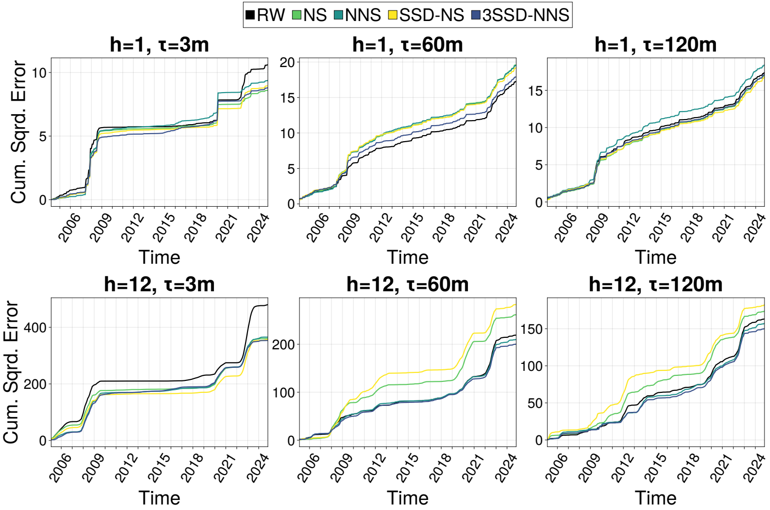
Relative Root Mean Squared Error

Out-of-Sample Relative RMSE (Model RMSE / RW RMSE) for 12-Month-Ahead Forecast Horizon - Moving Window

Model	m-3	m-6	m-12	m-24	m-60	m-120	Average
RW	1.000	1.000	1.000	1.000	1.000	1.000	1.000
NS	0.866	0.893	0.935	1.027	1.092	1.032	0.952
NNS	0.872	0.883	0.902	0.949	0.980	0.982	0.914**
SSD-NS	0.863	0.888	0.932	1.03	1.136	1.057	0.958
SSD-NNS	0.858	0.880	0.905	0.943	0.956	0.960	0.905**
SSD Ensemble	0.844	0.870	0.907	0.970	1.008	0.953	0.910**

Asterisks denote Diebold–Mariano rejection of equal RMSE with the RW benchmark (* 10%; ** 5%; *** 1%).

Cumulative Errors



Multi-horizon model confidence set (aSPA) (Quaedvlieg (2021))

Model	1	2	3	4	5	6	7	8	9	10	11	12	18	24	30	36	48	60	72	84	96	108	120
RW	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.07	0.09	0.12	0.12	0.10	0.08	0.06	0.07	0.09	0.10	0.10	0.08	0.07
NS	0.41	0.31	0.28	0.24	0.23	0.21	0.18	0.17	0.14	0.12	0.12	0.11	0.12	0.05	0.02	0.02	0.01	0.01	0.01	0.03	0.05	0.05	0.03
NNS	0.05	0.06	0.06	0.05	0.09	0.12	0.16	0.29	0.38	0.39	0.46	0.51	0.33	0.16	0.03	0.01	0.01	0.02	0.05	0.21	0.22	0.11	0.05
SD-NS	0.13	0.15	0.16	0.14	0.12	0.09	0.07	0.05	0.06	0.06	0.07	0.09	0.04	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.03
SSD-NS	0.14	0.22	0.23	0.22	0.21	0.18	0.16	0.13	0.11	0.09	0.08	0.09	0.04	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.03
1SD-NNS	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.07	0.07	0.06	0.06	0.07	0.02	0.00	0.02	0.07	0.21	0.33	0.43	0.52
1SSD-NNS	0.06	0.10	0.09	0.07	0.07	0.06	0.05	0.05	0.06	0.06	0.07	0.09	0.08	0.12	0.24	1.00	1.00	1.00	1.00	0.42	0.39	0.20	0.18
2SSD-NNS	0.09	0.09	0.06	0.05	0.05	0.05	0.05	0.05	0.06	0.07	0.08	0.09	0.22	0.16	0.15	0.08	0.06	0.07	0.30	0.42	0.40	0.55	0.53
3SSD-NNS	0.16	0.31	0.28	0.24	0.23	0.23	0.28	0.29	0.38	0.40	0.48	0.52	1.00	1.00	1.00	0.33	0.15	0.32	0.47	1.00	1.00	1.00	0.53
DNS	0.41	0.37	0.46	0.48	0.44	0.44	0.42	0.43	0.40	0.40	0.48	0.52	0.33	0.14	0.06	0.04	0.02	0.00	0.00	0.00	0.02	0.06	0.05
TVλ-DNS	1.00	1.00	0.46	0.48	0.41	0.33	0.28	0.29	0.38	0.38	0.34	0.29	0.12	0.07	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Static_Ensemble	0.16	0.22	0.23	0.23	0.23	0.21	0.19	0.29	0.38	0.38	0.35	0.38	0.33	0.16	0.09	0.07	0.06	0.07	0.12	0.40	0.40	0.50	0.31
SD_Ensemble	0.08	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.07	0.09	0.12	0.09	0.06	0.05	0.05	0.06	0.09	0.21	0.33	0.43	0.53
SSD_Ensemble	0.41	0.47	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.33	0.14	0.07	0.06	0.05	0.07	0.11	0.30	0.40	0.55	1.00

Maturity (months)

SSD-NNS, the loadings on the curvature factor.

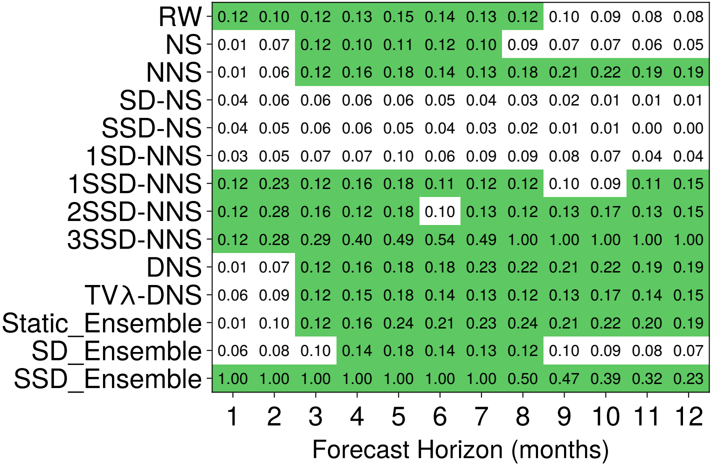
Multi-horizon model confidence set (aSPA) (Quaedvlieg (2021))

Model	1	2	3	4	5	6	7	8	9	10	11	12	18	24	30	36	48	60	72	84	96	108	120
RW	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.07	0.09	0.12	0.12	0.10	0.08	0.06	0.07	0.09	0.10	0.10	0.08	0.07
NS	0.41	0.31	0.28	0.24	0.23	0.21	0.18	0.17	0.14	0.12	0.12	0.11	0.12	0.05	0.02	0.02	0.01	0.01	0.01	0.03	0.05	0.05	0.03
NNS	0.05	0.06	0.06	0.05	0.09	0.12	0.16	0.29	0.38	0.39	0.46	0.51	0.33	0.16	0.03	0.01	0.01	0.02	0.05	0.21	0.22	0.11	0.05
SD-NS	0.13	0.15	0.16	0.14	0.12	0.09	0.07	0.05	0.06	0.06	0.07	0.09	0.04	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.03
SSD-NS	0.14	0.22	0.23	0.22	0.21	0.18	0.16	0.13	0.11	0.09	0.08	0.09	0.04	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.03
1SD-NNS	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.07	0.07	0.06	0.06	0.07	0.02	0.00	0.02	0.07	0.21	0.33	0.43	0.52
1SSD-NNS	0.06	0.10	0.09	0.07	0.07	0.06	0.05	0.05	0.06	0.06	0.07	0.09	0.08	0.12	0.24	1.00	1.00	1.00	1.00	0.42	0.39	0.20	0.18
2SSD-NNS	0.09	0.09	0.06	0.05	0.05	0.05	0.05	0.05	0.06	0.07	0.08	0.09	0.22	0.16	0.15	0.08	0.06	0.07	0.30	0.42	0.40	0.55	0.53
3SSD-NNS	0.16	0.31	0.28	0.24	0.23	0.23	0.28	0.29	0.38	0.40	0.48	0.52	1.00	1.00	1.00	0.33	0.15	0.32	0.47	1.00	1.00	1.00	0.53
DNS	0.41	0.37	0.46	0.48	0.44	0.44	0.42	0.43	0.40	0.40	0.48	0.52	0.33	0.14	0.06	0.04	0.02	0.00	0.00	0.00	0.02	0.06	0.05
TVλ-DNS	1.00	1.00	0.46	0.48	0.41	0.33	0.28	0.29	0.38	0.38	0.34	0.29	0.12	0.07	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Static_Ensemble	0.16	0.22	0.23	0.23	0.23	0.21	0.19	0.29	0.38	0.38	0.35	0.38	0.33	0.16	0.09	0.07	0.06	0.07	0.12	0.40	0.40	0.50	0.31
SD_Ensemble	0.08	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.07	0.09	0.12	0.09	0.06	0.05	0.05	0.06	0.09	0.21	0.33	0.43	0.53
SSD_Ensemble	0.41	0.47	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.33	0.14	0.07	0.06	0.05	0.07	0.11	0.30	0.40	0.55	1.00

Maturity (months)

SSD-NNS, the loadings on the curvature factor.

Multi-maturity model confidence set (aSPA)



SSD-NNS, the loadings on the curvature factor.

Multi-maturity model confidence set (aSPA)

	1	2	3	4	5	6	7	8	9	10	11	12
RW	0.12	0.10	0.12	0.13	0.15	0.14	0.13	0.12	0.10	0.09	0.08	0.08
NS	0.01	0.07	0.12	0.10	0.11	0.12	0.10	0.09	0.07	0.07	0.06	0.05
NNS	0.01	0.06	0.12	0.16	0.18	0.14	0.13	0.18	0.21	0.22	0.19	0.19
SD-NS	0.04	0.06	0.06	0.06	0.06	0.05	0.04	0.03	0.02	0.01	0.01	0.01
SSD-NS	0.04	0.05	0.06	0.06	0.05	0.04	0.03	0.02	0.01	0.01	0.00	0.00
1SD-NNS	0.03	0.05	0.07	0.07	0.10	0.06	0.09	0.09	0.08	0.07	0.04	0.04
1SSD-NNS	0.12	0.23	0.12	0.16	0.18	0.11	0.12	0.12	0.10	0.09	0.11	0.15
2SSD-NNS	0.12	0.28	0.16	0.12	0.18	0.10	0.13	0.12	0.13	0.17	0.13	0.15
3SSD-NNS	0.12	0.28	0.29	0.40	0.49	0.54	0.49	1.00	1.00	1.00	1.00	1.00
DNS	0.01	0.07	0.12	0.16	0.18	0.18	0.23	0.22	0.21	0.22	0.19	0.19
TVλ-DNS	0.06	0.09	0.12	0.15	0.18	0.14	0.13	0.12	0.13	0.17	0.14	0.15
Static_Ensemble	0.01	0.10	0.12	0.16	0.24	0.21	0.23	0.24	0.21	0.22	0.20	0.19
SD_Ensemble	0.06	0.08	0.10	0.14	0.18	0.14	0.13	0.12	0.10	0.09	0.08	0.07
SSD_Ensemble	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.47	0.39	0.32	0.23

Forecast Horizon (months)

SSD-NNS, the loadings on the curvature factor.

In the paper...

More In-Sample Results

- NS vs NNS
  - NNS provides a better interpretable factors.
- Curvature loading
  - The location of the maximum shifts from long maturities (72 months in 2006) to short maturities (12 months in 2024).
- Factor analysis
  - SSD-NNS factors align more closely with empirical proxies and deviate when needed.
- Extreme scenarios
  - SSD-NNS captures yield curve shapes more accurately than NS.

Simulation Study

- We simulate yields from a empirically calibrated dynamic three-factor Nelson–Siegel model.
- The shape of the yield curve varies over time through λ_t:
    - AR(1) dynamics
    - Structural breaks
    - Empirically estimated (via SSD-NS)
- Sample sizes: $T = 250,\ 500,\ 2000$
- SSD-NNS is competitive with SSD-NS, and sometimes outperforms it when the loading structure varies over time.

Concluding Remarks

1. Flexibility of neural networks improve capturing the yield curve.
2. Self-driving dynamics track the moving curvature.
3. Best forecasting model across all horizons and maturities.

- Neural Factor Model can be extended to other multivariate time series applications.

Thank you!

References

Bai, J. and Ng, S. (2013). Principal components estimation and identification of static factors. *Journal of Econometrics*, 176(1):18–29.

Bianchi, D., Büchner, M., and Tamoni, A. (2021). Bond Risk Premiums with Machine Learning. *Review of Financial Studies*, 34(2):1046–1089.

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

Creal, D., Koopman, S. J., and Lucas, A. (2013). Generalized autoregressive score models with applications. *Journal of Applied Econometrics*, 28(5):777–795.

Creal, D., Koopman, S. J., Lucas, A., and Zamojski, M. (2024). Observation-driven filtering of time-varying parameters using moment conditions. *Journal of Econometrics*, 238(2):105635.

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

Diebold, F. X., Rudebusch, G. D., and Aruoba, S. B. (2006). The macroeconomy and the yield curve: a dynamic latent factor approach. *Journal of Econometrics*, 131(1-2):309–338.

Hevia, C., Gonzalez-Rozada, M., Sola, M., and Spagnolo, F. (2014). Estimating and Forecasting the Yield Curve Using A Markov Switching Dynamic. *Journal of Applied Econometrics*, 30(June 2014):987–1009.

Koopman, S. J., Mallee, M. I., and Van Der Wel, M. (2010). Analyzing the term structure of interest rates using the dynamic Nelson-Siegel model with time-varying parameters. *Journal of Business and Economic Statistics*, 28(3):329–343.

Liu, Y. and Wu, J. C. (2021). Reconstructing the yield curve. *Journal of Financial Economics*, 142(3):1395–1425.

Nelson, C. R. and Siegel, A. F. (1987). Parsimonious modeling of yield curves. *Journal of Business*, pages 473–489.

Quaedvlieg, R. and Schotman, P. C. (2016). Score-Driven Nelson Siegel: Hedging Long-Term Liabilities. *SSRN Electronic Journal*.

Stock, J. and Watson, M. (2002). Forecasting Using Principal Components From a Large Number of Predictors. *Journal of the American Statistical Association*, 97(460):1167–1179.