MATRIX-VARIATE GAUSSIAN PROCESS STATE-SPACE MODEL WITH STRUCTURED LATENT DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Matrix-valued time series arise in many applications with coupled row-wise, column-wise, and temporal dependencies. Probabilistic modeling of their dynamics requires a framework that preserves row–column structure while remaining tractable for high-dimensional data. In this paper, we formulate a matrix-variate Gaussian process state-space model (MV-GPSSM) with structured latent dynamics. It represents the observed sequence as low-dimensional latent matrices through a Tucker2-based latent representation with PARATUCK2-type multiplicative and block term decomposition (BTD) Type-2 additive interaction structures, models their temporal evolution using a matrix-variate GP, and lifts the predictive distribution back to the observation space through learned row and column bases. Within this formulation, we characterize how latent interaction mechanisms yield structured matrix-valued kernels, with multiplicative and additive row–column interactions inducing Kronecker-product and Kronecker-sum covariance forms, respectively. Thus, these covariance structures arise as consequences of the assumed latent interaction mechanisms rather than being prescribed directly, thereby providing a dynamical interpretation of these structured covariance forms within the model family. We further present a computational complexity analysis of inference in the proposed MV-GPSSM. Experiments on battery electric vehicle fleet data and publicly available weather data show that the MV-GPSSM maintains predictive performance comparable to that of the baseline methods in terms of both point accuracy (root mean squared error) and probabilistic quality (negative log-likelihood), while remaining tractable at large matrix sizes where observation-space methods become computationally or memory infeasible. Controlled comparisons clarify the roles of matrix structure, latent representation, and GP-based dynamics. The learned row- and column-wise covariance structures further reveal qualitatively interpretable dependencies in vehicle behavior and weather spatiotemporal patterns.

1 INTRODUCTION

Matrix-valued time series arise in many settings where each time step is naturally represented as a matrix, such as multi-location multivariate measurements, network-structured data, and dynamic transport networks (Chen et al., 2021; Yuan et al., 2023; Tsay, 2024). In such problems, the row and column axes typically carry distinct semantics, while the temporal axis describes the evolution of the matrix itself. A useful predictor should therefore preserve matrix structure, capture uncertainty, and model dependencies across rows, columns, and time.

Matrix-variate Gaussian processes (MVGPs) provide a principled probabilistic framework for this purpose (Stegle et al., 2011; Takano and Jimbo, 2025). By introducing separate row and column kernels, MVGPs can explicitly model dependencies along the two matrix axes while retaining uncertainty-aware prediction. However, direct MVGP prediction in the observation space is computationally expensive. For a sequence of N matrices of size $M \times L$, the kernel matrix scales as $(N-1)ML \times (N-1)ML$, leading to a computational cost of $\mathcal{O}(N^3M^3L^3)$ for kernel inversion. This quickly becomes infeasible as the matrix dimensions increase, limiting the scalability of observation-space MVGPs for large-scale matrix-valued time series.

In many applications, matrix-valued data exhibit substantial redundancy across rows and columns, suggesting that their temporal evolution can be captured by a lower-dimensional structure (Yuan

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

et al., 2023; Tsay, 2024). This motivates probabilistic prediction in a reduced representation that preserves matrix structure while improving computational efficiency.

In this paper, we formulate a matrix-variate Gaussian process state-space model (MV-GPSSM) with structured latent dynamics. The key idea is to represent the observed sequence as a third-order tensor and apply a Tucker2-based decomposition (Tucker, 1966; Kolda and Bader, 2009) along the row and column modes, yielding a sequence of low-dimensional latent matrices. The proposed latent interaction structures build on PARATUCK2-type multiplicative interactions (Harshman and Lundy, 1996; Kolda and Bader, 2009) and block term decomposition (BTD) Type-2 additive interactions (Domanov et al., 2024; De Lathauwer, 2008; Vanderstukken et al., 2021). We then model their temporal evolution using a matrix-variate Gaussian process (MVGP) and lift the predictive distribution back to the observation space through learned row and column bases.

Importantly, this construction enables a characterization of how interaction structures in latent dynamics give rise to structured covariance forms. In particular, multiplicative and additive row–column interactions induce Kronecker-product and Kronecker-sum covariance forms, respectively.

By performing inference in a structured latent matrix space rather than the observation space, the proposed formulation reduces computational and memory costs while preserving matrix structure and uncertainty-aware prediction. This enables GP modeling for large matrix sizes where observation-space methods become computationally impractical.

We address the challenge of modeling high-dimensional matrix-valued time series using scalable and uncertainty-aware GP models. Our contributions are summarized as follows.

- We formulate a matrix-variate Gaussian process state-space model (MV-GPSSM) in a Tucker2-based latent matrix space with latent interaction structures that build on PARATUCK2-type multiplicative interactions and block term decomposition (BTD) Type-2 additive interactions. The model preserves row–column structure while modeling dynamics in this reduced latent space and lifts the predictive distribution back to the observation space through learned row and column bases.
- We characterize how PARATUCK2-type multiplicative and BTD Type-2 additive latent interaction structures yield structured matrix-valued kernels, revealing that multiplicative and additive interaction structures induce Kronecker-product and Kronecker-sum covariance forms, respectively. This characterization provides a dynamical interpretation of these structured covariance forms within the model family.
- We analyze the computational complexity of the latent-space MVGP inference, explicitly accounting for the costs of lifting and maximum a posteriori (MAP) filtering.
- We evaluate the proposed method on battery electric vehicle (BEV) fleet data and publicly available weather data, showing that it maintains predictive quality in terms of point accuracy (RMSE) and probabilistic quality (NLL) against baselines, while remaining tractable at large matrix sizes where observation-space methods become infeasible.
- We clarify the roles of matrix structure, latent representation, and GP-based dynamics through controlled comparisons.
- We analyze the learned row- and column-wise covariance structures, showing qualitatively interpretable dependencies in vehicle behavior and weather spatiotemporal patterns.

2 Related Work

Matrix autoregressive (MAR) models describe matrix-valued time series by bilinear parametric transitions acting on the row and column axes, and have been extended to Bayesian and nonlinear variants (Chen et al., 2021; Celani and Pagnottoni, 2024; Samadi and Billard, 2025; Bucci, 2025). More closely related in their use of latent matrix states, dynamic matrix factor models (Yuan et al., 2023; Yu et al., 2024) combine bilinear low-rank factorizations with prespecified parametric bilinear transitions, while stochasticity is represented by vectorized noise.

On the probabilistic side, matrix-variate process models capture dependencies along the row and column axes of matrix-valued observations. For static relational matrices, matrix-variate t-process models induce separable row- and column-wise covariance structures through a Kronecker-product

construction (Xu et al., 2011). Related matrix-variate GP models have used structured row- and column-wise kernels for direct probabilistic prediction of matrix-valued observations in time-series settings, without introducing latent matrix dynamics (Takano and Jimbo, 2025).

Probabilistic dynamics are the subject of Gaussian process state-space models (GPSSMs), which place nonparametric transitions on latent states, typically with independent GP priors across the coordinates of the transition function (Wang et al., 2008; Frigola et al., 2014; Eleftheriadis et al., 2017; Doerr et al., 2018; Mishra and Rajan, 2026), while output dependencies are introduced by coregionalization constructions on a vectorized state, from the intrinsic coregionalization model (ICM) developed for multi-task GP regression (Bonilla et al., 2007), a rank-one form of which underlies shared-GP transitions (Lin et al., 2024), to the more flexible linear model of coregionalization (LMC) used in output-dependent GPSSMs (Lin et al., 2023). These constructions therefore introduce dependencies among latent coordinates without separating the row and column axes of the state.

Across these lines of work, prior approaches provide matrix-structured parametric dynamics, probabilistic models of matrix-valued observations, or nonparametric dynamics on vectorized states. Our starting point is instead the probabilistic modeling of matrix-valued time series through their latent dynamics: a low-rank latent-matrix representation naturally induces a matrix-valued latent state, whose dynamics are modeled by a matrix-variate GP state-space model. Its row- and column-wise covariance structures arise as consequences of assumed latent interaction mechanisms, drawn from multilinear algebra, namely PARATUCK2 and BTD Type-2 decompositions, rather than being prescribed directly. To our knowledge, prior work has not formulated matrix-valued latent dynamics as a matrix-variate GP state-space model whose covariance structures are derived from these mechanisms.

3 Problem Setting

We consider a matrix-valued discrete-time series $\{X_k\}_{k\geq 1}$, with $X_k \in \mathbb{R}^{M\times L}$. The row and column axes encode two distinct structured attributes of the data, whereas the sequence index k describes the evolution of the matrix as a whole. Each matrix thus aggregates multiple structured attributes into a single matrix-valued state, whose row and column patterns can be highly heterogeneous; representative constructions are illustrated in Appendix A. This heterogeneity motivates models that capture structured dependencies across both modes.

Our objective is to infer the predictive distribution of the next observation matrix along a prediction trajectory:

$$p(\boldsymbol{X}_{t+1} \mid \mathcal{H}_t, \mathcal{D}_{\mathrm{tr}}).$$

Here, $\mathcal{H}_t$ denotes the observation history of $\boldsymbol{X}_k$ available up to time t along the prediction trajectory, and $\mathcal{D}_{\mathrm{tr}}$ denotes the training trajectory of N observations of the same form. Both trajectories are drawn from the same underlying dynamical process.

4 A Structured Model Family for Latent Dynamics

This section introduces a structured model family for latent dynamics by recalling the Tucker decomposition of a matrix-valued sequence, specializing it to Tucker2, and treating the resulting core matrices as latent states in a reduced matrix space. A key question is then how row- and column-wise interactions should enter their evolution; we draw on multiplicative (PARATUCK2) and additive (BTD Type-2) tensor decompositions to construct two such structures.

We consider a third-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{M\times L\times T}$ whose mode-3 slices form a temporal sequence $\{\boldsymbol{X}_t\}_{t=1}^{T}$, with $\boldsymbol{X}_t \in \mathbb{R}^{M\times L}$. A general Tucker decomposition of $\boldsymbol{\mathcal{X}}$ is written as

$$\boldsymbol{\mathcal{X}} \approx \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} \times_3 \boldsymbol{C},$$

where $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ denote factor matrices, $\boldsymbol{\mathcal{G}}$ denotes the core tensor, and $\times_n$ denotes the mode-n matrix product of a tensor with a matrix (Kolda and Bader, 2009; Ballard and Kolda, 2025). When the mode-3 factor $\boldsymbol{C}$ is taken as the identity, the model reduces to Tucker2 (Tucker, 1966), and each slice is represented as

$$\boldsymbol{X}_t \approx \boldsymbol{A}\boldsymbol{G}_t\boldsymbol{B}^{\top}, \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{M\times P}$, $\boldsymbol{B} \in \mathbb{R}^{L\times Q}$, and $\boldsymbol{G}_t \in \mathbb{R}^{P\times Q}$ is the t-th slice of $\boldsymbol{\mathcal{G}}$. The temporal variation in the mode-3 slice sequence $\{\boldsymbol{X}_t\}_{t=1}^{T}$ is represented by the corresponding core slice sequence $\{\boldsymbol{G}_t\}_{t=1}^{T}$.

By viewing the slice index as time, we can model its evolution as

$$\boldsymbol{G}_{t+1} = \boldsymbol{F}(\boldsymbol{G}_t). \tag{2}$$

We refer to this evolution as the latent dynamics, with $\boldsymbol{G}_t$ interpreted as a latent matrix.

4.1 Multiplicative Latent-Dynamics Model

To instantiate Eq. (2), we consider a PARATUCK2 form for the mode-3 slice sequence. In PARATUCK2, each core matrix is represented as $\boldsymbol{G}_t = \mathrm{diag}(\alpha_t)\,\boldsymbol{H}\,\mathrm{diag}(\beta_t)$, where $\boldsymbol{H} \in \mathbb{R}^{P\times Q}$ is a fixed matrix (Harshman and Lundy, 1996; Kolda and Bader, 2009). Here, α_t and β_t characterize the row- and column-mode contributions to the temporal sequence, respectively. Motivated by this interpretation, we formulate a generalized multiplicative model for the latent dynamics; the detailed construction is deferred to Appendix B.1. As a result, the latent dynamics take the form

$$\boldsymbol{G}_{t+1} = \boldsymbol{F}(\boldsymbol{G}_t) = \boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G}_t)\,\boldsymbol{W}\,\boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G}_t)^{\top}, \tag{3}$$

where $\boldsymbol{\Phi}_{\mathrm{R}}(\cdot) \in \mathbb{R}^{P\times S}$ and $\boldsymbol{\Phi}_{\mathrm{C}}(\cdot) \in \mathbb{R}^{Q\times S}$ each collect S vector-valued features for the row and column modes, respectively, and $\boldsymbol{W} \in \mathbb{R}^{S\times S}$ is a coefficient matrix. The number of features S may be finite or taken to infinity.

4.2 Additive Latent-Dynamics Model

As an additive counterpart to the multiplicative model Eq. (3), we consider a block decomposition model for the mode-3 slice sequence. This construction corresponds to the BTD Type-2 decomposition into a sum of rank-$(P/2, Q/2, \cdot)$ terms (Domanov et al., 2024; De Lathauwer, 2008; Vanderstukken et al., 2021); the details are deferred to Appendix B.2. Motivated by this interpretation, we formulate an additive model for the latent dynamics:

$$\boldsymbol{G}_{t+1} = \boldsymbol{F}(\boldsymbol{G}_t) = \boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G}_t)\boldsymbol{W}_{\mathrm{R}} + \boldsymbol{W}_{\mathrm{C}}^{\top}\boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G}_t)^{\top}, \tag{4}$$

where $\boldsymbol{W}_{\mathrm{R}} \in \mathbb{R}^{S\times Q}$ and $\boldsymbol{W}_{\mathrm{C}} \in \mathbb{R}^{S\times P}$ are coefficient matrices. This additive form separates the row-wise and column-wise contributions linearly, in contrast to the bilinear coupling of the multiplicative model of Eq. (3).

The multiplicative and additive forms above provide two structured models for latent dynamics in the Tucker core space. In the next section, we endow these latent dynamics with a matrix-variate GP prior and derive the resulting probabilistic predictor in the observation space.

5 MV-GPSSM with Structured Latent Dynamics

In Section 4, we introduced deterministic latent dynamics in Eq. (2). We now extend them to a probabilistic predictor with structured matrix-valued kernels that preserve row- and column-wise structure while operating on a reduced GP state space of dimension PQ instead of ML. Within this formulation, we characterize the covariance forms induced by the latent interaction mechanisms in Eqs. (3) and (4), and lift the predictive distributions back to the observation space through the Tucker2 structure.

5.1 Probabilistic Latent Dynamics and Structured Kernels

We place a matrix-variate Gaussian process (MVGP) prior on the transition map $\boldsymbol{F}$ and add matrix-variate process noise:

$$\boldsymbol{G}_{t+1} = \boldsymbol{F}(\boldsymbol{G}_t) + \boldsymbol{V}_t, \quad \boldsymbol{V}_t \sim \mathcal{MN}\Big(\boldsymbol{O},\, \boldsymbol{\Sigma}_{\mathrm{C}}^{(v)},\, \boldsymbol{\Sigma}_{\mathrm{R}}^{(v)}\Big). \tag{5}$$

Thus, for any pair of the latent matrices $\boldsymbol{G}_i$ and $\boldsymbol{G}_j$, the key quantity is the transition covariance $\mathrm{Cov}\big(\mathrm{vec}(\boldsymbol{G}_{i+1}), \mathrm{vec}(\boldsymbol{G}_{j+1}) \mid \boldsymbol{G}_i, \boldsymbol{G}_j\big)$, which is induced by the GP prior on $\boldsymbol{F}$ together with the process noise in Eq. (5). We write

$$\boldsymbol{K}^{(f)}(\boldsymbol{G}_i, \boldsymbol{G}_j) = \mathrm{Cov}\big(\mathrm{vec}(\boldsymbol{F}(\boldsymbol{G}_i)), \mathrm{vec}(\boldsymbol{F}(\boldsymbol{G}_j))\big), \qquad \boldsymbol{\Sigma}^{(v)} = \boldsymbol{\Sigma}_{\mathrm{C}}^{(v)} \otimes \boldsymbol{\Sigma}_{\mathrm{R}}^{(v)} \tag{6}$$

for the latent-function kernel and for the process-noise covariance of $\mathrm{vec}(\boldsymbol{V}_t)$, respectively. Since $\{\boldsymbol{V}_t\}$ and $\boldsymbol{F}$ are jointly independent, the process noise contributes to the transition covariance only when $i = j$.

For the multiplicative latent structure Eq. (3), we use

$$\boldsymbol{F}(\boldsymbol{G}) = \boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G})\, \boldsymbol{W}\, \boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G})^{\top}, \quad \boldsymbol{W} \sim \mathcal{MN}(\boldsymbol{O}, \boldsymbol{\Sigma}_{\mathrm{C}}^{(w)}, \boldsymbol{\Sigma}_{\mathrm{R}}^{(w)}).$$

This construction induces row and column kernels

$$\boldsymbol{K}_{\mathrm{R}}(\boldsymbol{G}_i, \boldsymbol{G}_j) = \boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G}_i)\boldsymbol{\Sigma}_{\mathrm{R}}^{(w)}\boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G}_j)^{\top}, \quad \boldsymbol{K}_{\mathrm{C}}(\boldsymbol{G}_i, \boldsymbol{G}_j) = \boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G}_i)\boldsymbol{\Sigma}_{\mathrm{C}}^{(w)}\boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G}_j)^{\top}. \tag{7}$$

Hence the latent-function kernel in Eq. (6) takes the Kronecker-product form

$$\boldsymbol{K}_{\otimes}^{(f)}(\boldsymbol{G}_i, \boldsymbol{G}_j) = \boldsymbol{K}_{\mathrm{C}}(\boldsymbol{G}_i, \boldsymbol{G}_j) \otimes \boldsymbol{K}_{\mathrm{R}}(\boldsymbol{G}_i, \boldsymbol{G}_j) \tag{8}$$

and the transition covariance is given by

$$\mathrm{Cov}\big(\mathrm{vec}(\boldsymbol{G}_{i+1}), \mathrm{vec}(\boldsymbol{G}_{j+1}) \,|\, \boldsymbol{G}_i, \boldsymbol{G}_j\big) = \boldsymbol{K}_{\otimes}^{(f)}(\boldsymbol{G}_i, \boldsymbol{G}_j) + \delta_{ij}\, \boldsymbol{\Sigma}^{(v)}. \tag{9}$$

The Kronecker-product form captures multiplicative interactions between the two modes, since high covariance requires both row-side and column-side similarities to be large.

For the additive latent structure Eq. (4), we use

$$\boldsymbol{F}(\boldsymbol{G}) = \boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G})\, \boldsymbol{W}_{\mathrm{R}} + \boldsymbol{W}_{\mathrm{C}}^{\top}\, \boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G})^{\top},\ \boldsymbol{W}_{\mathrm{R}} \sim \mathcal{MN}(\boldsymbol{O}, \boldsymbol{I}_Q, \boldsymbol{\Sigma}_{\mathrm{R}}^{(w)}),\ \boldsymbol{W}_{\mathrm{C}} \sim \mathcal{MN}(\boldsymbol{O}, \boldsymbol{I}_P, \boldsymbol{\Sigma}_{\mathrm{C}}^{(w)}).$$

Then the latent-function kernel takes the Kronecker-sum form

$$\boldsymbol{K}_{\oplus}^{(f)}(\boldsymbol{G}_i, \boldsymbol{G}_j) = \boldsymbol{K}_{\mathrm{C}}(\boldsymbol{G}_i, \boldsymbol{G}_j) \oplus \boldsymbol{K}_{\mathrm{R}}(\boldsymbol{G}_i, \boldsymbol{G}_j) \tag{10}$$

and the transition covariance is given by

$$\mathrm{Cov}\big(\mathrm{vec}(\boldsymbol{G}_{i+1}), \mathrm{vec}(\boldsymbol{G}_{j+1}) \mid \boldsymbol{G}_i, \boldsymbol{G}_j\big) = \boldsymbol{K}_{\oplus}^{(f)}(\boldsymbol{G}_i, \boldsymbol{G}_j) + \delta_{ij}\, \boldsymbol{\Sigma}^{(v)}, \tag{11}$$

where $\oplus$ denotes the Kronecker sum

$$\boldsymbol{K}_{\mathrm{C}}(\boldsymbol{G}_i, \boldsymbol{G}_j) \oplus \boldsymbol{K}_{\mathrm{R}}(\boldsymbol{G}_i, \boldsymbol{G}_j) = \boldsymbol{K}_{\mathrm{C}}(\boldsymbol{G}_i, \boldsymbol{G}_j) \otimes \boldsymbol{I}_P + \boldsymbol{I}_Q \otimes \boldsymbol{K}_{\mathrm{R}}(\boldsymbol{G}_i, \boldsymbol{G}_j).$$

The Kronecker-sum form allows dependencies to arise from either row-side or column-side similarity, and is therefore appropriate when the two modes contribute additively. Detailed derivations and the infinite-feature RBF kernels are given in Appendix C.

5.2 Observation Model and MV-GPSSM Formulation

Building on the Tucker2 representation in Eq. (1), we define the observation model at time t as

$$\boldsymbol{X}_t = \boldsymbol{A}\boldsymbol{G}_t\boldsymbol{B}^{\top} + \boldsymbol{E}_t, \quad \boldsymbol{E}_t \sim \mathcal{MN}\Big(\boldsymbol{O},\, \boldsymbol{\Sigma}_{\mathrm{C}}^{(e)},\, \boldsymbol{\Sigma}_{\mathrm{R}}^{(e)}\Big), \tag{12}$$

where $\boldsymbol{E}_t$ follows a matrix-normal distribution with column covariance $\boldsymbol{\Sigma}_{\mathrm{C}}^{(e)} \in \mathbb{R}^{L\times L}$ and row covariance $\boldsymbol{\Sigma}_{\mathrm{R}}^{(e)} \in \mathbb{R}^{M\times M}$.

Based on the probabilistic latent dynamics in Eq. (5) and the observation model in Eq. (12), we define the proposed MV-GPSSM with structured latent dynamics:

$$\boldsymbol{G}_{t+1} = \boldsymbol{F}(\boldsymbol{G}_t) + \boldsymbol{V}_t, \quad \boldsymbol{X}_t = \boldsymbol{A}\boldsymbol{G}_t\boldsymbol{B}^{\top} + \boldsymbol{E}_t. \tag{13}$$

Here, the transition covariance is specified by either Eq. (9) or Eq. (11), depending on the chosen structured kernel construction.

5.3 Lifted Predictive Distribution in the Observation Space

We derive the exact conditional one-step predictive distribution under the proposed MV-GPSSM by constructing the latent-space GP predictive distribution and lifting it to the observation space.

Let $\{\boldsymbol{G}_k\}_{k=1}^{N}$ denote the latent matrix sequence associated with $\mathcal{D}_{\mathrm{tr}}$, yielding $N-1$ one-step transitions. Because the regression targets are the noisy transitions in Eq. (5), the process noise enters only the diagonal blocks of the training covariance matrix,

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}^{(f)}(\boldsymbol{G}_1, \boldsymbol{G}_1) & \cdots & \boldsymbol{K}^{(f)}(\boldsymbol{G}_1, \boldsymbol{G}_{N-1}) \\ \vdots & \ddots & \vdots \\ \boldsymbol{K}^{(f)}(\boldsymbol{G}_{N-1}, \boldsymbol{G}_1) & \cdots & \boldsymbol{K}^{(f)}(\boldsymbol{G}_{N-1}, \boldsymbol{G}_{N-1}) \end{bmatrix} + \boldsymbol{I}_{N-1} \otimes \boldsymbol{\Sigma}^{(v)}, \tag{14}$$

and the corresponding training output vector $\boldsymbol{y} := [\mathrm{vec}(\boldsymbol{G}_2)^\top \ \cdots \ \mathrm{vec}(\boldsymbol{G}_N)^\top]^\top$.

Given the current latent matrix $\boldsymbol{G}_t$, define the test–training cross-covariance $\boldsymbol{K}_t^{(f)} = [\,\boldsymbol{K}^{(f)}(\boldsymbol{G}_t, \boldsymbol{G}_1), \ldots, \boldsymbol{K}^{(f)}(\boldsymbol{G}_t, \boldsymbol{G}_{N-1})\,]$ and the test covariance $\boldsymbol{K}_{tt} = \boldsymbol{K}^{(f)}(\boldsymbol{G}_t, \boldsymbol{G}_t) + \boldsymbol{\Sigma}^{(v)}$. Standard GP conditioning yields the latent predictive distribution

$$\mathrm{vec}(\boldsymbol{G}_{t+1}) \mid \boldsymbol{G}_t, \mathcal{D}_{\mathrm{tr}} \sim \mathcal{N}\big(\boldsymbol{\mu}_{t+1|t},\, \boldsymbol{Q}_{t+1|t}\big)\,,$$

with $\boldsymbol{\mu}_{t+1|t} = \boldsymbol{K}_t^{(f)} \boldsymbol{K}^{-1} \boldsymbol{y}$ and $\boldsymbol{Q}_{t+1|t} = \boldsymbol{K}_{tt} - \boldsymbol{K}_t^{(f)} \boldsymbol{K}^{-1} \boldsymbol{K}_t^{(f)\top}$. This distribution is then lifted to the observation space via Eq. (12), yielding

$$\mathrm{vec}(\boldsymbol{X}_{t+1}) \mid \boldsymbol{G}_t, \mathcal{D}_{\mathrm{tr}} \sim \mathcal{N}\Big((\boldsymbol{B} \otimes \boldsymbol{A})\boldsymbol{\mu}_{t+1|t},\, (\boldsymbol{B} \otimes \boldsymbol{A})\boldsymbol{Q}_{t+1|t}(\boldsymbol{B} \otimes \boldsymbol{A})^\top + \boldsymbol{\Sigma}_{\mathrm{C}}^{(e)} \otimes \boldsymbol{\Sigma}_{\mathrm{R}}^{(e)}\Big)\,.$$

Finally, the exact predictive distribution is obtained by marginalizing over the latent state,

$$p(\boldsymbol{X}_{t+1} \mid \mathcal{H}_t, \mathcal{D}_{\mathrm{tr}}) = \int p(\boldsymbol{X}_{t+1} \mid \boldsymbol{G}_t, \mathcal{D}_{\mathrm{tr}})\, p(\boldsymbol{G}_t \mid \mathcal{H}_t, \mathcal{D}_{\mathrm{tr}})\, d\boldsymbol{G}_t, \tag{15}$$

which is generally non-Gaussian and therefore intractable (see Appendix D for details).

5.4 Learning and Prediction in MV-GPSSM

As well as the predictive distribution, the marginal likelihood $p(\mathcal{D}_{\mathrm{tr}} \mid \theta)$ with the model parameters θ is also intractable. This is due to the marginalization over the latent matrix trajectory $\boldsymbol{G}_{1:N}$. Therefore, our learning procedure approximately maximizes $p(\mathcal{D}_{\mathrm{tr}} \mid \theta)$ by a hard-EM (expectation-maximization) scheme that alternates between:

- **(E-step)** MAP reconstruction of the latent trajectory $\boldsymbol{G}_{1:N}$,
- **(M-step)** maximization of the complete-data log-likelihood $\mathcal{L}_c(\theta, \boldsymbol{G}_{1:N}; \mathcal{D}_{\mathrm{tr}})$.

The objective $\mathcal{L}_c(\theta, \boldsymbol{G}_{1:N}; \mathcal{D}_{\mathrm{tr}})$ decomposes into prior, transition, and observation terms, where the transition term is the joint MVGP marginal log-likelihood of all $N-1$ transitions with the training covariance matrix in Eq. (14). Here, $\theta = \{\theta_{\mathrm{s}}, \theta_{\mathrm{o}}\}$, where $\theta_{\mathrm{s}} = \{\theta_K, \boldsymbol{\Sigma}_{\mathrm{C}}^{(v)}, \boldsymbol{\Sigma}_{\mathrm{R}}^{(v)}\}$ and $\theta_{\mathrm{o}} = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\Sigma}_{\mathrm{C}}^{(e)}, \boldsymbol{\Sigma}_{\mathrm{R}}^{(e)}\}$ denote the latent dynamics and observation model parameters, respectively, and θ_K denotes the hyperparameters of the MVGP kernel in Eqs. (8) or (10). See Appendix E.1 for the detailed derivation. Under this objective, in contrast to conventional two-stage procedures that estimate θ_{s} and θ_{o} separately (Yu et al., 2024; Yuan et al., 2023), the parameters are optimized jointly, allowing the latent representation to adapt to prediction.

To address the intractability of the predictive distribution in Eq. (15), the prediction procedure proceeds in a similar way. At time t, given $\mathcal{H}_t$, we reconstruct the latent state by sequential MAP filtering,

$$\hat{\boldsymbol{G}}_{t|t} = \underset{\boldsymbol{G}_t}{\mathrm{argmax}} \Big[\log p(\boldsymbol{G}_t \mid \hat{\boldsymbol{G}}_{t-1|t-1}, \hat{\theta}_{\mathrm{s}}, \mathcal{D}_{\mathrm{tr}}) + \log p(\boldsymbol{X}_t \mid \boldsymbol{G}_t, \hat{\theta}_{\mathrm{o}})\Big], \tag{16}$$

where $\hat{\theta}$ denotes the parameters learned from the training data, and only the current latent matrix is updated while the previous estimate $\hat{\boldsymbol{G}}_{t-1|t-1}$ is held fixed. Appendix E.3 derives Eq. (16) as a Dirac assumed-density approximation of the exact filtering recursion. We then evaluate the predictive distribution under the MV-GPSSM by latent-space GP conditioning followed by lifting through the observation model, yielding a tractable approximation to Eq. (15) in the form

$$p(\mathrm{vec}(\boldsymbol{X}_{t+1}) \mid \mathcal{H}_t, \mathcal{D}_{\mathrm{tr}}) \approx \mathcal{N}\Big(\hat{\boldsymbol{m}}_{t+1}, \hat{\boldsymbol{S}}_{t+1}\Big)\,,$$

where the predictive moments are given explicitly in Appendix E.2.

5.5 Computational Complexity Analysis

We analyze the computational complexity of the proposed MV-GPSSM, explicitly accounting for the costs of lifting and MAP filtering. In the Obs. MVGP formulation, the setup for the $\boldsymbol{K}^{-1}$-terms in GP conditioning costs $\mathcal{O}(N^3M^3L^3)$ in time and $\mathcal{O}(N^2M^2L^2)$ in space, and each subsequent full-covariance prediction costs $\mathcal{O}(N^2M^3L^3)$ in time. The proposed MV-GPSSM replaces these Obs. MVGP operations with latent-space operations of dimension PQ, followed by lifting to the observation space.

Proposition 1 (Computational complexity of the MV-GPSSM)**.** *Let $\mathcal{T}$ and $\mathcal{S}$ denote time and space complexity, respectively. For $N-1$ latent transitions, the kernel matrix of the latent-space MVGP has size $(N-1)PQ \times (N-1)PQ$. The inverse-kernel setup costs*

$$\mathcal{T}_{\text{setup}} = \mathcal{O}(N^3P^3Q^3), \qquad \mathcal{S}_{\text{setup}} = \mathcal{O}(N^2P^2Q^2).$$

Given this setup, each subsequent prediction with a full observation-space covariance costs

$$\mathcal{T}_{\text{pred}} = \mathcal{T}_{\text{GP}} + \mathcal{T}_{\text{lift}} + \mathcal{T}_{\text{MAP}}, \text{ where } \mathcal{T}_{\text{GP}} = \mathcal{O}(N^2P^3Q^3),\ \mathcal{T}_{\text{lift}} = \mathcal{O}(MLPQ(PQ+ML)).$$

For the first-order gradient-based MAP update, the MAP filtering costs

$$\mathcal{T}_{\text{MAP}} = \mathcal{O}\big(n_{\text{iter}}\left(N^2P^3Q^3 + MPQ + MQL + M^2L + ML^2\right)\big),$$

where n_{iter} is the number of gradient steps used for one-step MAP filtering.

Proof. See Appendix F. □

6 Experiments

We evaluate predictive accuracy, probabilistic quality, and scalability on two real-world tasks. The baselines comprise an observation-space MVGP (Obs. MVGP; Takano and Jimbo (2025)), matrix autoregression (MAR; Chen et al. (2021)), and vector autoregression (VAR; Lütkepohl (2005)), and latent-space methods with the low-dimensional representation: vectorized GPSSMs modeling the latent coordinates with independent GPs (Vec. Indep-GPSSM) or an LMC (Vec. LMC-GPSSM; Lin et al. (2023)), and a parametric bilinear Matrix SSM (Yu et al., 2024). See Table 3 in Appendix G.1 for the model configurations. These comparisons examine the effects of matrix structure, latent representation, and GP-based dynamics.

We report per-entry root mean squared error (RMSE) and negative log-likelihood (NLL) evaluated with the full predictive covariance, as the mean and standard deviation over five seeds. Unless otherwise stated, we use $P = 10$ and $Q = 18$, a Kronecker-sum kernel with row- and column-wise radial basis function (RBF) kernels, structured process noise, and joint optimization. Appendices G.3.3 and G.3.4 examine these choices.

6.1 SOC Prediction on BEV Fleet Data

We evaluate fleet-level state-of-charge (SOC) prediction for battery electric vehicles (BEVs), motivated by the use of aggregated EV fleets as distributed energy resources in virtual power plants. Their operational flexibility depends on future SOC trajectories, making accurate fleet-wide forecasts important for coordinating charging and discharging while preserving mobility needs and supporting grid balancing under high renewable penetration. We jointly predict SOC trajectories using a vehicle $\times$ time matrix representation that captures temporal context and cross-vehicle heterogeneity.

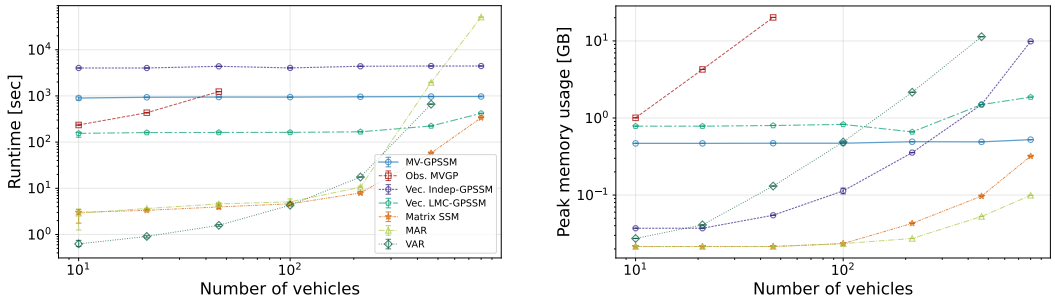
We use proprietary SOC records from October 1, 2024 to April 30, 2025. For each week k, $\boldsymbol{X}_k \in \mathbb{R}^{M\times L}$ with $L = 42$ (4-hour samples per week), and we predict $\boldsymbol{X}_{k+1}$ from past data. Using real-world fleet data, we evaluate M from 10 to 800 vehicles, validating scalability at fleet sizes where Obs. MVGP becomes impractical. See Appendix G.2 for details. Representative examples of the resulting matrix-valued SOC trajectories are shown in Fig. 4.

Predictive performance. Table 1 reports RMSE and NLL across representative fleet sizes. Full results across all fleet sizes are provided in Appendix G.3.1. The proposed MV-GPSSM achieves RMSE and NLL comparable to those of Obs. MVGP for $M \leq 46$, beyond which Obs. MVGP becomes infeasible (Appendix G.3.1).

The comparisons clarify the roles of the three components. For the matrix structure, the lower NLL of MV-GPSSM relative to those of Vec. Indep-GPSSM and Vec. LMC-GPSSM indicates that preserving row- and column-wise covariance structures is more effective than introducing output correlations after vectorization; moreover, the performance of the vectorized models changes little with the flexibility of the output kernel (Appendix G.3.1). For the latent representation, MV-GPSSM remains comparable to Obs. MVGP at the sizes for which the latter is feasible, while scaling to larger fleets. For the GP-based dynamics, replacing parametric bilinear transitions with GP transitions improves both the RMSE and the NLL in the observation space (Obs. MVGP vs. MAR) and in the latent space (MV-GPSSM vs. Matrix SSM). Taken together, the advantage of MV-GPSSM over VAR, which lacks all three components, reflects their combined effect.

Table 1: Representative BEV SOC validation results, reported as mean (standard deviation) over five seeds. OOM denotes an out-of-memory failure; complete results are provided in Appendix G.3.1.

	RMSE per entry			NLL per entry		
Model	$M = 10$	$M = 100$	$M = 800$	$M = 10$	$M = 100$	$M = 800$
MV-GPSSM (ours)	**0.223 (0.074)**	**0.225 (0.008)**	**0.230 (0.001)**	**-0.547 (0.227)**	**-0.437 (0.148)**	**-0.158 (0.012)**
Obs. MVGP	0.231 (0.079)	OOM	OOM	-0.348 (0.812)	OOM	OOM
Vec. Indep-GPSSM	0.325 (0.101)	0.245 (0.021)	0.239 (0.002)	$2.85\ (1.24) \times 10^4$	$2.45\ (0.52) \times 10^4$	$2.32\ (0.69) \times 10^4$
Vec. LMC-GPSSM	0.241 (0.080)	0.233 (0.011)	0.238 (0.002)	$1.11\ (0.99) \times 10^4$	$2.11\ (0.72) \times 10^4$	$4.28\ (3.98) \times 10^3$
Matrix SSM	0.291 (0.098)	0.273 (0.022)	0.256 (0.003)	$5.22\ (0.82) \times 10^2$	$2.27\ (0.16) \times 10^4$	$2.52\ (0.17) \times 10^4$
MAR	0.843 (0.791)	0.463 (0.259)	0.261 (0.002)	0.422 (0.038)	0.065 (0.041)	0.048 (0.001)
VAR	2.732 (2.210)	17.1 (8.6)	OOM	0.954 (0.022)	0.952 (0.005)	OOM



(a) Runtime

(b) Peak memory usage

Figure 1: Scalability for the BEV SOC prediction task with respect to the number of vehicles M. Missing markers denote out-of-memory configurations.

Scalability. Figure 1 shows that Obs. MVGP becomes infeasible beyond $M = 46$ and VAR fails at $M = 800$. Both failures are structural: observation-space GP methods require a kernel matrix of size $(N-1)ML \times (N-1)ML$, i.e., $638{,}400 \times 638{,}400$ at $M = 800$ and $N = 20$. VAR requires estimating an $ML \times ML$ coefficient matrix, i.e., $33{,}600 \times 33{,}600$ at $M = 800$, together with several other dense $ML \times ML$ matrices, such as the Gram and residual covariance matrices and their factorizations. The latent-space SSM baselines rely on a low-rank latent representation, and MAR uses bilinear $M \times M$ and $L \times L$ coefficient matrices; both therefore remain feasible at $M = 800$, but they attain higher RMSE and NLL than the proposed method.

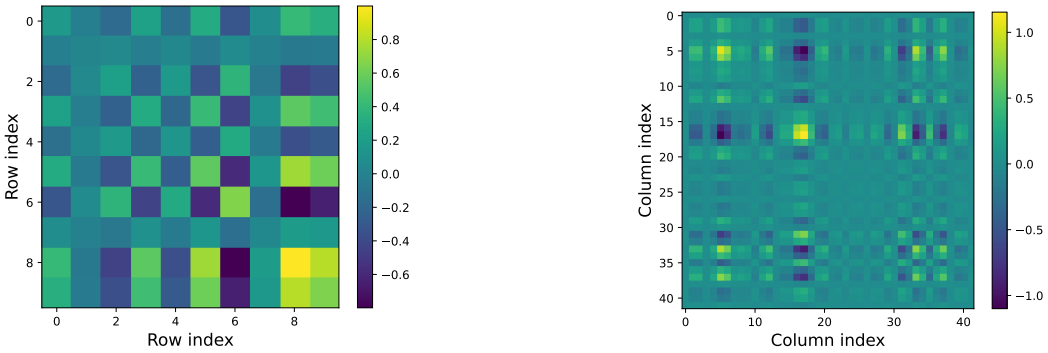
As a result, the proposed method remains tractable up to $M = 800$ while maintaining comparable RMSE and favorable NLL (Table 1). Furthermore, we examine whether the proposed model captures heterogeneous vehicle behaviors through interpretability analysis, with detailed results in Appendix G.3.1.

6.2 SLP Prediction on Public Weather Data

To complement the proprietary BEV study, we use public sea level pressure (SLP) observations from MERRA-2 reanalysis data over Western Europe, sampled daily (Gelaro et al., 2017; Shinde et al., 2026). Whereas the BEV experiment emphasizes large-scale fleet prediction, this experiment assesses reproducibility and generality on public geophysical data with structured spatiotemporal dependencies.

We extract SLP data from $M = 10$ cities (shown in Table 6) and construct matrices $\boldsymbol{X}_k \in \mathbb{R}^{M \times L}$ with $L = 42$ daily observations over a 42-day window. See Appendix G.2 for details. The task is to predict $\boldsymbol{X}_{k+1}$ one step ahead based on past observations, with forecasting performed jointly across all cities. As shown in Appendix G.3.2, the proposed method achieves comparable RMSE and the best NLL among all compared methods for up to $M = 50$ cities, while capturing the main temporal variations of SLP.

Interpretability. Figure 2 shows the learned row and column covariance structures lifted through the coefficients $\boldsymbol{A}$ and $\boldsymbol{B}$, yielding $\boldsymbol{A}\boldsymbol{K}_{\mathrm{R}}\boldsymbol{A}^\top$ and $\boldsymbol{B}\boldsymbol{K}_{\mathrm{C}}\boldsymbol{B}^\top$, respectively. The row kernel reflects spatial relationships across cities, while the column kernel captures temporal dependencies over



(a) Row kernel structure $\boldsymbol{A}\boldsymbol{K}_{\mathrm{R}}\boldsymbol{A}^{\top}$

(b) Column kernel structure $\boldsymbol{B}\boldsymbol{K}_{\mathrm{C}}\boldsymbol{B}^{\top}$

Figure 2: Lifted row and column covariance structures for the MERRA-2 SLP matrix. Row indices 0–9 correspond to Edinburgh, Glasgow, Manchester, Dublin, Galway, London, Rennes, Nantes, A Coruña, and Bilbao, respectively.

the 42-day horizon. The row covariance is consistent with large-scale SLP variability over Western Europe, driven by the North Atlantic pressure contrast between the Azores High and the Icelandic Low. The column covariance exhibits nontrivial temporal structure, consistent with the persistence of large-scale circulation patterns over several days to weeks. These results indicate that the learned covariance structures capture spatiotemporal dependencies that are qualitatively consistent with known large-scale circulation patterns, rather than providing a quantitative physical validation.

7 Conclusion

We formulated a matrix-variate Gaussian process state-space model (MV-GPSSM) with structured latent dynamics for probabilistic prediction of matrix-valued time series. The model represents the observed sequence through a Tucker2-based latent matrix representation, models the temporal evolution of the latent matrices with a matrix-variate GP, and lifts the predictive distribution back to the observation space through learned row and column bases. This shifts the dominant cost from the observation space to the reduced latent space while preserving row–column structure and uncertainty-aware prediction. Within this formulation, the latent interaction structures that build on PARATUCK2-type multiplicative and BTD Type-2 additive interactions induce Kronecker-product and Kronecker-sum covariance forms, respectively, providing a dynamical interpretation of these structured covariance forms in the proposed model family.

Experiments on BEV fleet data and publicly available weather data show that the proposed method maintains predictive performance comparable to that of the baseline methods in terms of both point accuracy (RMSE) and probabilistic quality (NLL), while remaining tractable at large matrix sizes where observation-space methods become infeasible, including experiments with up to 800 vehicles. Controlled comparisons clarify the roles of matrix structure, latent representation, and GP-based dynamics; and the learned row- and column-wise covariance structures show qualitatively interpretable dependencies in vehicle behavior and weather spatiotemporal patterns.

Several limitations remain. The latent dimensions (P, Q) are determined a priori, learning and prediction rely on approximate inference via hard-EM and MAP filtering, so that the latent filtering distribution is propagated only through its mode (Appendix G.3.5 shows that the omitted latent uncertainty has a limited influence on the predictive distribution in the evaluated settings), and the cubic scaling with respect to the sequence length N is not addressed. These motivate future work on improved inference and better scalability with respect to sequence length.

AI USE STATEMENT

In this work, we used generative AI tools for English language editing and improving the readability of the manuscript. We have not used generative AI tools to generate research ideas, scientific claims, experimental results, datasets, or analyses. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not raise specific ethical concerns.

REPRODUCIBILITY STATEMENT

Appendices B–F provide the derivations of the structured transition covariances, predictive distribution, learning procedure, and computational complexity.

Appendix G.1 details the experimental setup, including hardware and software environments, model configurations, initialization procedures, optimization settings, and evaluation protocols. Appendix G.2 describes the construction, preprocessing, and chronological splits of both datasets, while Appendix G.3 reports the complete results and ablation studies.

The supplementary material includes the source code required to reproduce the experiments on the publicly available MERRA-2 dataset.

REFERENCES

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, C. K. Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Zhang Shunting, Michael Suo, Phil Tillet, Xu Zhao, Eikan Wang, Keren Zhou, Richard Zou, Xiaodong Wang, Ajit Mathews, William Wen, Gregory Chanan, Peng Wu, and Soumith Chintala. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*:929–947, 2024.

Grey Ballard and Tamara G. Kolda. *Tensor Decompositions for Data Science*. Cambridge University Press, New York, 2025.

Edwin V. Bonilla, Kian Ming A. Chai, and Christopher K. I. Williams. Multi-task Gaussian Process Prediction. *Advances in Neural Information Processing Systems*, 20, 2007.

Andrea Bucci. A Smooth Transition Autoregressive Model for Matrix-Variate Time Series. *Computational Economics*, 65(1):429–458, 2025.

Alessandro Celani and Paolo Pagnottoni. Matrix Autoregressive Models: Generalization and Bayesian Estimation. *Studies in Nonlinear Dynamics & Econometrics*, 28(2):227–248, 2024.

Rong Chen, Han Xiao, and Dan Yang. Autoregressive Models for Matrix-valued Time Series. *Journal of Econometrics*, 222(1):539–560, 2021.

Lieven De Lathauwer. Decompositions of a Higher-Order Tensor in Block Terms—Part II: Definitions and Uniqueness. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1033–1066, 2008.

Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A Multilinear Singular Value Decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000a.

Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. On the Best Rank-1 and Rank-$(R_1, R_2, \ldots, R_N)$ Approximation of Higher-Order Tensors. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2000b.

Andreas Doerr, Christian Daniel, Martin Schiegg, Duy Nguyen-Tuong, Stefan Schaal, Marc Toussaint, and Sebastian Trimpe. Probabilistic Recurrent State-Space Models. *Proceedings of the 35th International Conference on Machine Learning*, PMLR 80:1280–1289, 2018.

Ignat Domanov, Nico Vervliet, Eric Evert, and Lieven De Lathauwer. Decomposition of a Tensor into Multilinear Rank-$(M_r, N_r, \cdot)$ Terms. *SIAM Journal on Matrix Analysis and Applications*, 45 (3):1310–1334, 2024.

Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On Sequential Monte Carlo Sampling Methods for Bayesian Filtering. *Statistics and Computing*, 10(3):197–208, 2000.

Stefanos Eleftheriadis, Tom Nicholson, Marc Deisenroth, and James Hensman. Identification of Gaussian Process State Space Models. *Advances in Neural Information Processing Systems*, 30, 2017.

Roger Frigola, Yutian Chen, and Carl E. Rasmussen. Variational Gaussian Process State-Space Models. *Advances in Neural Information Processing Systems*, 27, 2014.

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration. *Advances in Neural Information Processing Systems*, 31, 2018.

Ronald Gelaro, Will McCarty, Max J. Suárez, Ricardo Todling, Andrea Molod, Lawrence Takacs, Cynthia A. Randles, Anton Darmenov, Michael G. Bosilovich, Rolf Reichle, Krzysztof Wargan, Lawrence Coy, Richard Cullather, Clara Draper, Santha Akella, Virginie Buchard, Austin Conaty, Arlindo M. Da Silva, Wei Gu, Gi-Kong Kim, Randal Koster, Robert Lucchesi, Dagmar Merkova, Jon Eric Nielsen, Gary Partyka, Steven Pawson, William Putman, Michele Rienecker, Siegfried D. Schubert, Meta Sienkiewicz, and Bin Zhao. The Modern-Era Retrospective Analysis for Research and Applications, Version 2 (MERRA-2). *Journal of Climate*, 30(14):5419–5454, 2017.

Richard A. Harshman and Margaret E. Lundy. Uniqueness Proof for a Family of Models Sharing Features of Tucker's Three-mode Factor Analysis and PARAFAC/candecomp. *Psychometrika*, 61 (1):133–154, 1996.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations*, 2015.

Tamara G. Kolda and Brett W. Bader. Tensor Decompositions and Applications. *SIAM Review*, 51 (3):455–500, 2009.

Zhidi Lin, Lei Cheng, Feng Yin, Lexi Xu, and Shuguang Cui. Output-Dependent Gaussian Process State-Space Model. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1–5, 2023.

Zhidi Lin, Juan Maroñas, Ying Li, Feng Yin, and Sergios Theodoridis. Towards Efficient Modeling and Inference in Multi-Dimensional Gaussian Process State-Space Models. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 12881–12885, 2024.

Helmut Lütkepohl. *New Introduction to Multiple Time Series Analysis*. Springer, Berlin, 2005.

Anurodh Mishra and Raj Thilak Rajan. Domain-Aware Gaussian Process State-Space Models. *Signal Processing*, 238:110003, 2026.

Natural earth. `https://www.naturalearthdata.com`, 2026.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

S. Yaser Samadi and Lynne Billard. On a matrix-valued autoregressive model. *Journal of Time Series Analysis*, 46(1):3–32, 2025.

Rajat Shinde, Kumar Ankur, Christopher E. Phillips, Aman Gupta, Simon Pfreundschuh, Sujit Roy, Sheyenne Kirkland, Vishal Gaur, Venkatesh Kolluru, Amy Lin, Prajun Trital, Aditi Sheshadri, Udaysankar Nair, Manil Maskey, and Rahul Ramachandran. WxC-bench: A Novel Dataset for Weather and Climate Downstream Tasks. *Scientific Data*, 13:596, 2026.

Oliver Stegle, Christoph Lippert, Joris M Mooij, Neil Lawrence, and Karsten Borgwardt. Efficient Inference in Matrix-variate Gaussian Models with iid Observation Noise. *Advances in Neural Information Processing Systems*, 24, 2011.

Seiya Takano and Tomohiko Jimbo. Warped Matrix-variate Gaussian Processes with Structured Kernels for Multi-step-ahead Driving Behaviour Prediction. *SICE Journal of Control, Measurement, and System Integration*, 18(1):2455222, 2025.

Ruey S. Tsay. Matrix-Variate Time Series Analysis: A Brief Review and Some New Developments. *International Statistical Review*, 92(2):246–262, 2024.

Ledyard R. Tucker. Some Mathematical Notes on Three-mode Factor Analysis. *Psychometrika*, 31 (3):279–311, 1966.

Jeroen Vanderstukken, Patrick Kürschner, Ignat Domanov, and Lieven De Lathauwer. Systems of Polynomial Equations, Higher-Order Tensor Decompositions, and Multidimensional Harmonic Retrieval: A Unifying Framework. Part II: The Block Term Decomposition. *SIAM Journal on Matrix Analysis and Applications*, 42(2):913–953, 2021.

Jack M. Wang, David J. Fleet, and Aaron Hertzmann. Gaussian Process Dynamical Models for Human Motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):283–298, 2008.

Zenglin Xu, Feng Yan, and Yuan Qi. Sparse Matrix-Variate t Process Blockmodels. *Proceedings of the AAAI Conference on Artificial Intelligence*, 25(1):543–548, 2011.

Ruofan Yu, Rong Chen, Han Xiao, and Yuefeng Han. Dynamic Matrix Factor Models for High Dimensional Time Series. *arXiv preprint arXiv:2407.05624*, 2024.

Chaofeng Yuan, Zhigen Gao, Xuming He, Wei Huang, and Jianhua Guo. Two-way Dynamic Factor Models for High-dimensional Matrix-valued Time Series. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(5):1517–1537, 2023.

A Illustrative Matrix Constructions for Time-Series Data

In this appendix, we provide illustrative examples of matrix-valued representations to aid intuition for the modeling choices in the main text. These examples illustrate how temporal context and row/column structures are encoded within each matrix.

Figure 3 shows an abstract view of matrix-valued time series. Each matrix can be interpreted as a slice of a third-order tensor, where the row and column axes encode two distinct structured attributes and the sequence index captures temporal evolution.

Figure 4 presents a concrete example of the matrix-valued time series used in Section 6.1, constructed from SOC data of multiple BEVs. In this representation, rows correspond to individual vehicles and columns correspond to temporal indices within a prediction window. Each matrix captures the temporal evolution of multiple vehicles, where behaviors are heterogeneous but may exhibit latent similarities in usage patterns that can be exploited for prediction.

In practice, such data exhibit heterogeneous patterns across rows and columns, as well as variability across time indices. This motivates structured probabilistic models for matrix-valued dynamics.

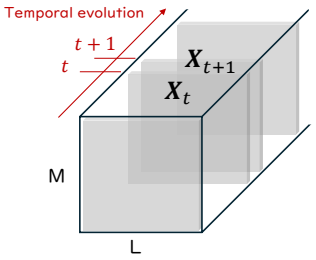


Figure 3: Abstract illustration of a matrix-valued time series as slices of a third-order tensor.

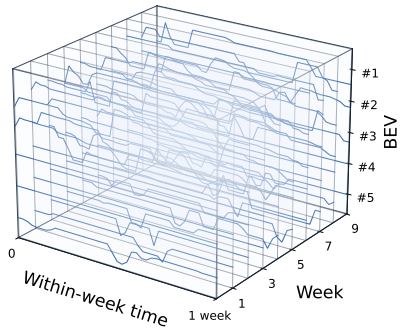


Figure 4: Example of a matrix-valued time series.

B Construction of Latent-Dynamics Models

We first recall the Tucker2 decomposition (Tucker, 1966), shown in Fig. 5, which represents each matrix using shared row and column bases and a low-dimensional core tensor. This yields a natural matrix-valued latent representation.

Building on this representation, we directly model the temporal evolution of the core tensor slices, $\{\boldsymbol{G}_t\}_{t=1}^{T}$. In the following, we construct multiplicative and additive latent-dynamics models that capture interactions between row-side and column-side factors.

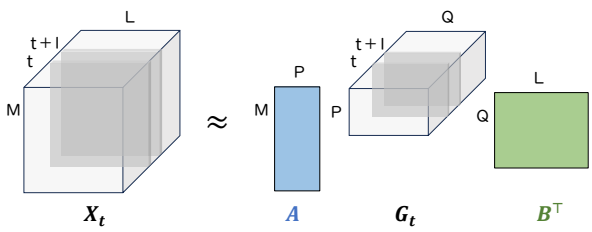


Figure 5: Tucker2 decomposition.

B.1 Multiplicative Latent-Dynamics Model

In this appendix, we construct the multiplicative latent-dynamics model given in Eq. (3). We begin with a generalized factorization, illustrated in Fig. 6:

$$\boldsymbol{G}_t = \boldsymbol{U}_{\mathrm{R},t} \boldsymbol{H} \boldsymbol{U}_{\mathrm{C},t}^{\top}, \tag{17}$$

where $\boldsymbol{U}_{\mathrm{R},t} \in \mathbb{R}^{P \times P}$ and $\boldsymbol{U}_{\mathrm{C},t} \in \mathbb{R}^{Q \times Q}$.

This formulation can be viewed as a generalization of PARATUCK2. In the PARATUCK2 model, the core matrix is expressed as

$$\boldsymbol{G}_t = \mathrm{diag}(\alpha_t)\, \boldsymbol{H}\, \mathrm{diag}(\beta_t),$$

where $\alpha_t \in \mathbb{R}^P$ and $\beta_t \in \mathbb{R}^Q$ are mode-specific scaling vectors (Harshman and Lundy, 1996; Kolda and Bader, 2009). Compared with this form, Eq. (17) replaces the diagonal transformations with full matrices $\boldsymbol{U}_{\mathrm{R},t}$ and $\boldsymbol{U}_{\mathrm{C},t}$, thereby allowing richer interactions between row-side and column-side factors.

Since PARATUCK2 is a structural factorization rather than a dynamical model, we introduce the temporal evolution at time t by letting the latent matrices evolve according to

$$\boldsymbol{U}_{\mathrm{R},t+1} = f_{\mathrm{R}}(\boldsymbol{U}_{\mathrm{R},t}) = h_{\mathrm{R}}(\boldsymbol{G}_t), \qquad \boldsymbol{U}_{\mathrm{C},t+1} = f_{\mathrm{C}}(\boldsymbol{U}_{\mathrm{C},t}) = h_{\mathrm{C}}(\boldsymbol{G}_t), \tag{18}$$

where $h_{\mathrm{R}} := f_{\mathrm{R}} \circ g_{\mathrm{R}}$ and $h_{\mathrm{C}} := f_{\mathrm{C}} \circ g_{\mathrm{C}}$, with $\boldsymbol{U}_{\mathrm{R},t} = g_{\mathrm{R}}(\boldsymbol{G}_t)$ and $\boldsymbol{U}_{\mathrm{C},t} = g_{\mathrm{C}}(\boldsymbol{G}_t)$. Substituting Eq. (18) into Eq. (17) yields

$$\boldsymbol{G}_{t+1} = h_{\mathrm{R}}(\boldsymbol{G}_t)\, \boldsymbol{H}\, h_{\mathrm{C}}(\boldsymbol{G}_t)^{\top}. \tag{19}$$

To model the generally nonlinear mappings $h_{\mathrm{R}}(\cdot)$ and $h_{\mathrm{C}}(\cdot)$ in a tractable form, we parameterize them by feature expansions:

$$h_{\mathrm{R}}(\boldsymbol{G}_t) = \boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G}_t)\boldsymbol{W}_{\mathrm{R}}, \qquad h_{\mathrm{C}}(\boldsymbol{G}_t) = \boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G}_t)\boldsymbol{W}_{\mathrm{C}}, \tag{20}$$

where $\boldsymbol{\Phi}_{\mathrm{R}}(\cdot) \in \mathbb{R}^{P \times S}$ and $\boldsymbol{\Phi}_{\mathrm{C}}(\cdot) \in \mathbb{R}^{Q \times S}$ each collect S vector-valued features for the row and column modes, respectively, and $\boldsymbol{W}_{\mathrm{R}} \in \mathbb{R}^{S \times P}$ and $\boldsymbol{W}_{\mathrm{C}} \in \mathbb{R}^{S \times Q}$ are the corresponding coefficient matrices. The feature number S may be finite or taken to infinity. Substituting Eq. (20) into Eq. (19) gives

$$\boldsymbol{G}_{t+1} = \boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G}_t)\, \boldsymbol{W}\, \boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G}_t)^{\top},$$

where $\boldsymbol{W} := \boldsymbol{W}_{\mathrm{R}} \boldsymbol{H} \boldsymbol{W}_{\mathrm{C}}^{\top} \in \mathbb{R}^{S \times S}$.

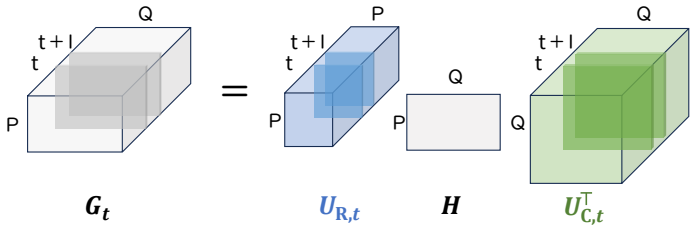


Figure 6: Multiplicative model from a generalized PARATUCK2.

B.2 Additive Latent-Dynamics Model

In this appendix, we construct the additive latent-dynamics model given in Eq. (4). We begin with the following block decomposition of $\boldsymbol{G}_t \in \mathbb{R}^{P \times Q}$:

$$\boldsymbol{G}_t = [\boldsymbol{H}_{11}\ \boldsymbol{H}_{12}]\, \mathrm{diag}(\tilde{\boldsymbol{G}}_{1,t}, \tilde{\boldsymbol{G}}_{2,t})[\boldsymbol{H}_{21}\ \boldsymbol{H}_{22}]^{\top}.$$

Here, $\tilde{\boldsymbol{G}}_{1,t}, \tilde{\boldsymbol{G}}_{2,t} \in \mathbb{R}^{P/2 \times Q/2}$ are block core matrices, while $\boldsymbol{H}_{11}, \boldsymbol{H}_{12} \in \mathbb{R}^{P \times P/2}$ and $\boldsymbol{H}_{21}, \boldsymbol{H}_{22} \in \mathbb{R}^{Q \times Q/2}$ are the corresponding coefficient matrices. Substituting this decomposition into Eq. (1), we obtain a sum of two Tucker2 decompositions:

$$\boldsymbol{X}_t = \boldsymbol{A}\boldsymbol{G}_t\boldsymbol{B}^{\top} = (\boldsymbol{A}\boldsymbol{H}_{11})\tilde{\boldsymbol{G}}_{1,t}(\boldsymbol{B}\boldsymbol{H}_{21})^{\top} + (\boldsymbol{A}\boldsymbol{H}_{12})\tilde{\boldsymbol{G}}_{2,t}(\boldsymbol{B}\boldsymbol{H}_{22})^{\top}.$$

This construction corresponds to the block term decomposition (BTD) Type-2 decomposition into a sum of rank-$(P/2, Q/2, \cdot)$ terms (Domanov et al., 2024; De Lathauwer, 2008; Vanderstukken et al., 2021). Motivated by this decomposition, we consider the following additive factorization, illustrated in Fig. 7:

$$\boldsymbol{G}_t = \boldsymbol{U}_{\mathrm{R},t}\boldsymbol{H}_{21}^\top + \boldsymbol{H}_{12}\boldsymbol{U}_{\mathrm{C},t}^\top, \tag{21}$$

where $\boldsymbol{U}_{\mathrm{R},t} := \boldsymbol{H}_{11}\tilde{\boldsymbol{G}}_{1,t} \in \mathbb{R}^{P\times Q/2}$ and $\boldsymbol{U}_{\mathrm{C},t} := \boldsymbol{H}_{22}\tilde{\boldsymbol{G}}_{2,t}^\top \in \mathbb{R}^{Q\times P/2}$.

As in the multiplicative case, we introduce the temporal evolution through

$$\boldsymbol{U}_{\mathrm{R},t+1} = f_{\mathrm{R}}(\boldsymbol{U}_{\mathrm{R},t}) = h_{\mathrm{R}}(\boldsymbol{G}_t), \qquad \boldsymbol{U}_{\mathrm{C},t+1} = f_{\mathrm{C}}(\boldsymbol{U}_{\mathrm{C},t}) = h_{\mathrm{C}}(\boldsymbol{G}_t), \tag{22}$$

where $h_{\mathrm{R}} := f_{\mathrm{R}} \circ g_{\mathrm{R}}$ and $h_{\mathrm{C}} := f_{\mathrm{C}} \circ g_{\mathrm{C}}$, with $\boldsymbol{U}_{\mathrm{R},t} = g_{\mathrm{R}}(\boldsymbol{G}_t)$ and $\boldsymbol{U}_{\mathrm{C},t} = g_{\mathrm{C}}(\boldsymbol{G}_t)$. Substituting Eq. (22) into Eq. (21) yields

$$\boldsymbol{G}_{t+1} = h_{\mathrm{R}}(\boldsymbol{G}_t)\boldsymbol{H}_{21}^\top + \boldsymbol{H}_{12}h_{\mathrm{C}}(\boldsymbol{G}_t)^\top. \tag{23}$$

To model the generally nonlinear mappings $h_{\mathrm{R}}(\cdot)$ and $h_{\mathrm{C}}(\cdot)$ in a tractable form, we parameterize them by feature expansions:

$$h_{\mathrm{R}}(\boldsymbol{G}_t) = \boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G}_t)\boldsymbol{W}_1, \qquad h_{\mathrm{C}}(\boldsymbol{G}_t) = \boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G}_t)\boldsymbol{W}_2, \tag{24}$$

where $\boldsymbol{\Phi}_{\mathrm{R}}(\cdot) \in \mathbb{R}^{P\times S}$, $\boldsymbol{\Phi}_{\mathrm{C}}(\cdot) \in \mathbb{R}^{Q\times S}$, $\boldsymbol{W}_1 \in \mathbb{R}^{S\times Q/2}$, and $\boldsymbol{W}_2 \in \mathbb{R}^{S\times P/2}$. Substituting Eq. (24) into Eq. (23) gives

$$\boldsymbol{G}_{t+1} = \boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G}_t)\boldsymbol{W}_{\mathrm{R}} + \boldsymbol{W}_{\mathrm{C}}^\top\boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G}_t)^\top,$$

where $\boldsymbol{W}_{\mathrm{R}} := \boldsymbol{W}_1\boldsymbol{H}_{21}^\top \in \mathbb{R}^{S\times Q}$ and $\boldsymbol{W}_{\mathrm{C}} := \boldsymbol{W}_2\boldsymbol{H}_{12}^\top \in \mathbb{R}^{S\times P}$.

These should be interpreted as model-construction arguments based on structured tensor factorizations, rather than as identifiability results for the corresponding tensor decompositions.

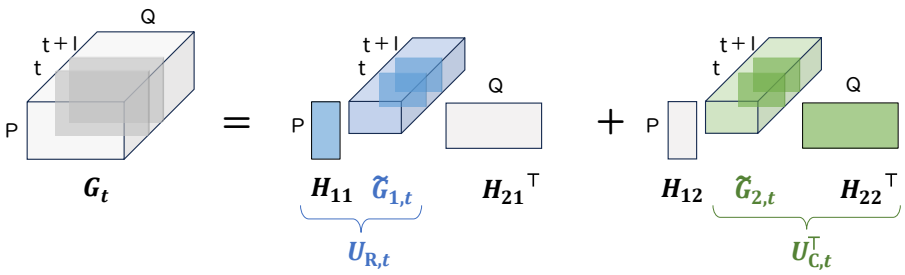


Figure 7: Additive model from the BTD Type-2 decomposition.

C Derivation of Structured Transition Covariances

This appendix derives the transition covariances used in Section 5.1.

C.1 Multiplicative Latent-Dynamics Model

Consider

$$\boldsymbol{F}(\boldsymbol{G}) = \boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G})\,\boldsymbol{W}\,\boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G})^\top, \quad \boldsymbol{W} \sim \mathcal{MN}(\boldsymbol{O}, \boldsymbol{\Sigma}_{\mathrm{C}}^{(w)}, \boldsymbol{\Sigma}_{\mathrm{R}}^{(w)}).$$

By standard properties of the matrix-normal distribution,

$$\boldsymbol{F}(\boldsymbol{G}) \sim \mathcal{MN}\Big(\boldsymbol{O},\, \boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G})\boldsymbol{\Sigma}_{\mathrm{C}}^{(w)}\boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G})^\top,\, \boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G})\boldsymbol{\Sigma}_{\mathrm{R}}^{(w)}\boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G})^\top\Big).$$

Equivalently,

$$\mathrm{vec}\big(\boldsymbol{F}(\boldsymbol{G})\big) \sim \mathcal{N}\Big(\boldsymbol{0},\, \big(\boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G})\boldsymbol{\Sigma}_{\mathrm{C}}^{(w)}\boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G})^\top\big) \otimes \big(\boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G})\boldsymbol{\Sigma}_{\mathrm{R}}^{(w)}\boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G})^\top\big)\Big).$$

Define

$$\boldsymbol{K}_{\mathrm{R}}(\boldsymbol{G}_i, \boldsymbol{G}_j) = \boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G}_i)\boldsymbol{\Sigma}_{\mathrm{R}}^{(w)}\boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G}_j)^\top, \quad \boldsymbol{K}_{\mathrm{C}}(\boldsymbol{G}_i, \boldsymbol{G}_j) = \boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G}_i)\boldsymbol{\Sigma}_{\mathrm{C}}^{(w)}\boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G}_j)^\top.$$

Then the finite-dimensional distributions are jointly Gaussian, with cross-covariance

$$\mathrm{Cov}\big(\mathrm{vec}(\boldsymbol{F}(\boldsymbol{G}_i)), \mathrm{vec}(\boldsymbol{F}(\boldsymbol{G}_j))\big) = \boldsymbol{K}_{\mathrm{C}}(\boldsymbol{G}_i, \boldsymbol{G}_j) \otimes \boldsymbol{K}_{\mathrm{R}}(\boldsymbol{G}_i, \boldsymbol{G}_j). \tag{25}$$

Now consider the noisy state transition

$$\boldsymbol{G}_{k+1} = \boldsymbol{F}(\boldsymbol{G}_k) + \boldsymbol{V}_k, \quad \boldsymbol{V}_k \sim \mathcal{MN}\Big(\boldsymbol{O},\, \boldsymbol{\Sigma}_{\mathrm{C}}^{(v)},\, \boldsymbol{\Sigma}_{\mathrm{R}}^{(v)}\Big), \tag{26}$$

with mutually independent $\{\boldsymbol{V}_k\}$ and independent of $\boldsymbol{F}$. Since

$$\mathrm{vec}(\boldsymbol{G}_{k+1}) = \mathrm{vec}(\boldsymbol{F}(\boldsymbol{G}_k)) + \mathrm{vec}(\boldsymbol{V}_k),$$

we obtain

$$\begin{aligned}\mathrm{Cov}\big(\mathrm{vec}(\boldsymbol{G}_{i+1}), \mathrm{vec}(\boldsymbol{G}_{j+1}) \mid \boldsymbol{G}_i, \boldsymbol{G}_j\big) &= \mathrm{Cov}\big(\mathrm{vec}(\boldsymbol{F}(\boldsymbol{G}_i)), \mathrm{vec}(\boldsymbol{F}(\boldsymbol{G}_j))\big) + \delta_{ij}\ \mathrm{Cov}\big(\mathrm{vec}(\boldsymbol{V}_i)\big) \\ &= \boldsymbol{K}_{\otimes}^{(f)}(\boldsymbol{G}_i, \boldsymbol{G}_j) + \delta_{ij}\, \boldsymbol{\Sigma}^{(v)}.\end{aligned}$$

This is the transition covariance in Eq. (9). Here $\boldsymbol{K}_{\otimes}^{(f)}(\boldsymbol{G}_i, \boldsymbol{G}_j) = \boldsymbol{K}_{\mathrm{C}}(\boldsymbol{G}_i, \boldsymbol{G}_j) \otimes \boldsymbol{K}_{\mathrm{R}}(\boldsymbol{G}_i, \boldsymbol{G}_j)$ is the noise-free latent-function kernel Eq. (25), and the process-noise covariance $\boldsymbol{\Sigma}^{(v)} = \boldsymbol{\Sigma}_{\mathrm{C}}^{(v)} \otimes \boldsymbol{\Sigma}_{\mathrm{R}}^{(v)}$ appears only for $i = j$. Cross-covariances between distinct transitions, in particular those between a test transition and the training transitions, therefore contain no process-noise contribution.

C.2 Additive Latent-Dynamics Model

For the additive structure, consider

$$\boldsymbol{F}(\boldsymbol{G}) = \boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G})\, \boldsymbol{W}_{\mathrm{R}} + \boldsymbol{W}_{\mathrm{C}}^\top \boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G})^\top, \tag{27}$$

with independent priors

$$\boldsymbol{W}_{\mathrm{R}} \sim \mathcal{MN}(\boldsymbol{O}, \boldsymbol{I}_Q, \boldsymbol{\Sigma}_{\mathrm{R}}^{(w)}), \quad \boldsymbol{W}_{\mathrm{C}} \sim \mathcal{MN}(\boldsymbol{O}, \boldsymbol{I}_P, \boldsymbol{\Sigma}_{\mathrm{C}}^{(w)}).$$

Vectorizing Eq. (27) gives

$$\mathrm{vec}\big(\boldsymbol{F}(\boldsymbol{G})\big) = (\boldsymbol{I}_Q \otimes \boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G}))\ \mathrm{vec}(\boldsymbol{W}_{\mathrm{R}}) + (\boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G}) \otimes \boldsymbol{I}_P)\ \mathrm{vec}(\boldsymbol{W}_{\mathrm{C}}^\top).$$

Therefore,

$$\begin{aligned}\mathrm{Cov}\big(\mathrm{vec}(\boldsymbol{F}(\boldsymbol{G}))\big) = (\boldsymbol{I}_Q \otimes \boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G}))\, \mathrm{Cov}\big(\mathrm{vec}(\boldsymbol{W}_{\mathrm{R}})\big)(\boldsymbol{I}_Q \otimes \boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G}))^\top \\ + (\boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G}) \otimes \boldsymbol{I}_P)\, \mathrm{Cov}\big(\mathrm{vec}(\boldsymbol{W}_{\mathrm{C}}^\top)\big)(\boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G}) \otimes \boldsymbol{I}_P)^\top,\end{aligned}$$

because the cross-covariance vanishes by independence of $\boldsymbol{W}_{\mathrm{R}}$ and $\boldsymbol{W}_{\mathrm{C}}$. Using

$$\mathrm{Cov}\big(\mathrm{vec}(\boldsymbol{W}_{\mathrm{R}})\big) = \boldsymbol{I}_Q \otimes \boldsymbol{\Sigma}_{\mathrm{R}}^{(w)}, \quad \mathrm{Cov}\big(\mathrm{vec}(\boldsymbol{W}_{\mathrm{C}}^\top)\big) = \boldsymbol{\Sigma}_{\mathrm{C}}^{(w)} \otimes \boldsymbol{I}_P,$$

we obtain

$$\mathrm{Cov}\big(\mathrm{vec}(\boldsymbol{F}(\boldsymbol{G}))\big) = \big(\boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G})\boldsymbol{\Sigma}_{\mathrm{C}}^{(w)}\boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G})^\top\big) \otimes \boldsymbol{I}_P + \boldsymbol{I}_Q \otimes \big(\boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G})\boldsymbol{\Sigma}_{\mathrm{R}}^{(w)}\boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G})^\top\big).$$

Hence,

$$\mathrm{Cov}\big(\mathrm{vec}(\boldsymbol{F}(\boldsymbol{G}_i)), \mathrm{vec}(\boldsymbol{F}(\boldsymbol{G}_j))\big) = \boldsymbol{K}_{\mathrm{C}}(\boldsymbol{G}_i, \boldsymbol{G}_j) \otimes \boldsymbol{I}_P + \boldsymbol{I}_Q \otimes \boldsymbol{K}_{\mathrm{R}}(\boldsymbol{G}_i, \boldsymbol{G}_j), \tag{28}$$

which is the Kronecker sum

$$\boldsymbol{K}_{\mathrm{C}}(\boldsymbol{G}_i, \boldsymbol{G}_j) \oplus \boldsymbol{K}_{\mathrm{R}}(\boldsymbol{G}_i, \boldsymbol{G}_j) = \boldsymbol{K}_{\mathrm{C}}(\boldsymbol{G}_i, \boldsymbol{G}_j) \otimes \boldsymbol{I}_P + \boldsymbol{I}_Q \otimes \boldsymbol{K}_{\mathrm{R}}(\boldsymbol{G}_i, \boldsymbol{G}_j).$$

Combining this with the noise model Eq. (26) yields

$$\mathrm{Cov}\big(\mathrm{vec}(\boldsymbol{G}_{i+1}), \mathrm{vec}(\boldsymbol{G}_{j+1}) \mid \boldsymbol{G}_i, \boldsymbol{G}_j\big) = \boldsymbol{K}_{\oplus}^{(f)}(\boldsymbol{G}_i, \boldsymbol{G}_j) + \delta_{ij}\, \boldsymbol{\Sigma}^{(v)}.$$

This is the transition covariance in Eq. (11). As in the multiplicative case, $\boldsymbol{K}_{\oplus}^{(f)}(\boldsymbol{G}_i, \boldsymbol{G}_j) = \boldsymbol{K}_{\mathrm{C}}(\boldsymbol{G}_i, \boldsymbol{G}_j) \oplus \boldsymbol{K}_{\mathrm{R}}(\boldsymbol{G}_i, \boldsymbol{G}_j)$ in Eq. (28) is noise-free, and the process-noise covariance enters only the diagonal blocks.

C.3 Row-/Column-wise RBF Kernels

We derive the row- and column-wise RBF kernels used in the latent-space structured kernels, inspired by the derivation of the ordinary RBF kernels for vector inputs in Rasmussen and Williams (2005). Recall that the row- and column-wise kernel matrices are defined in Eq. (7) as

$$\boldsymbol{K}_{\mathrm{R}}(\boldsymbol{G}_i, \boldsymbol{G}_j) = \boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G}_i)\boldsymbol{\Sigma}_{\mathrm{R}}^{(w)}\boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G}_j)^\top, \qquad \boldsymbol{K}_{\mathrm{C}}(\boldsymbol{G}_i, \boldsymbol{G}_j) = \boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G}_i)\boldsymbol{\Sigma}_{\mathrm{C}}^{(w)}\boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G}_j)^\top. \tag{29}$$

Here, $\boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G}) \in \mathbb{R}^{P\times S_{\mathrm{R}}}$ evaluates row-wise features, and $\boldsymbol{\Phi}_{\mathrm{C}}(\boldsymbol{G}) \in \mathbb{R}^{Q\times S_{\mathrm{C}}}$ evaluates column-wise features. The matrices $\boldsymbol{\Sigma}_{\mathrm{R}}^{(w)} \succeq 0$ and $\boldsymbol{\Sigma}_{\mathrm{C}}^{(w)} \succeq 0$ are feature-space covariance matrices. This finite-dimensional form is useful for defining a general class of valid positive semi-definite row- and column-wise kernels.

We first consider the row-wise kernel. Let

$$\boldsymbol{g}_{i,p:} \in \mathbb{R}^Q$$

denote the p-th row of $\boldsymbol{G}_i$. For basis centers $\boldsymbol{u}_s \in \mathbb{R}^Q$, define the row-wise Gaussian features by

$$\phi_{s,p}(\boldsymbol{G}_i) = \gamma_p \exp\left(-\frac{\|\boldsymbol{g}_{i,p:} - \boldsymbol{u}_s\|_2^2}{\zeta_p^2}\right), \qquad s = 1, \dots, S_{\mathrm{R}},$$

where γ_p and ζ_p denote amplitude and width, respectively. Using the normalization

$$[\boldsymbol{\Phi}_{\mathrm{R}}(\boldsymbol{G}_i)]_{ps} = S_{\mathrm{R}}^{-\frac{1}{2}}\phi_{s,p}(\boldsymbol{G}_i),$$

the (p, p')-entry of $\boldsymbol{K}_{\mathrm{R}}(\boldsymbol{G}_i, \boldsymbol{G}_j)$ is

$$[\boldsymbol{K}_{\mathrm{R}}(\boldsymbol{G}_i, \boldsymbol{G}_j)]_{pp'} = \frac{1}{S_{\mathrm{R}}}\sum_{s=1}^{S_{\mathrm{R}}}\sum_{s'=1}^{S_{\mathrm{R}}}\phi_{s,p}(\boldsymbol{G}_i)[\boldsymbol{\Sigma}_{\mathrm{R}}^{(w)}]_{ss'}\phi_{s',p'}(\boldsymbol{G}_j).$$

Thus, any positive semi-definite $\boldsymbol{\Sigma}_{\mathrm{R}}^{(w)}$ defines a valid row-wise kernel by allowing correlations between basis functions. In general, however, this finite feature-covariance construction does not reduce to a simple closed-form pairwise RBF kernel. We therefore specialize next to an infinite-center Gaussian convolution representation, which yields the closed-form RBF kernel used in the experiments.

To obtain the closed-form row-wise RBF kernel used in our experiments, we use an infinite-center Gaussian basis representation. Let the auxiliary center variable be sampled from a density

$$q(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u}; \boldsymbol{0}, \tau_{\mathrm{R}}^2\boldsymbol{I}_Q), \qquad \boldsymbol{u} \in \mathbb{R}^Q.$$

The particular choice of q is used only to define a valid sampling measure for the auxiliary centers. The limiting kernel is defined through the importance-weighted features

$$\varphi_p(\boldsymbol{x}; \boldsymbol{u}) = q(\boldsymbol{u})^{-\frac{1}{2}}\gamma_p\left(\frac{2}{\pi\zeta_p^2}\right)^{\frac{Q}{4}}\exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{u}\|_2^2}{\zeta_p^2}\right),$$

where $\gamma_p > 0$ and $\zeta_p > 0$ are the row-wise amplitude and length-scale hyperparameters. For $\boldsymbol{x} = \boldsymbol{g}_{i,p:}$ and $\boldsymbol{y} = \boldsymbol{g}_{j,p':}$, we obtain

$$\begin{aligned}[\boldsymbol{K}_{\mathrm{R}}(\boldsymbol{G}_i, \boldsymbol{G}_j)]_{pp'} &= \mathbb{E}_{\boldsymbol{u}\sim q}\left[\varphi_p(\boldsymbol{x}; \boldsymbol{u})\varphi_{p'}(\boldsymbol{y}; \boldsymbol{u})\right] \\ &= \int \bar{\varphi}_p(\boldsymbol{x}; \boldsymbol{u})\bar{\varphi}_{p'}(\boldsymbol{y}; \boldsymbol{u})\, d\boldsymbol{u},\end{aligned}$$

where

$$\bar{\varphi}_p(\boldsymbol{x}; \boldsymbol{u}) = \gamma_p\left(\frac{2}{\pi\zeta_p^2}\right)^{\frac{Q}{4}}\exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{u}\|_2^2}{\zeta_p^2}\right).$$

Thus, the factor $q(\boldsymbol{u})^{-1/2}$ converts the expectation with respect to q into a Gaussian convolution integral with respect to $d\boldsymbol{u}$. Evaluating this integral yields

$$[\boldsymbol{K}_{\mathrm{R}}(\boldsymbol{G}_i, \boldsymbol{G}_j)]_{pp'} = \gamma_p\gamma_{p'}\left(\frac{2\zeta_p\zeta_{p'}}{\zeta_p^2 + \zeta_{p'}^2}\right)^{\frac{Q}{2}}\exp\left(-\frac{\|\boldsymbol{g}_{i,p:} - \boldsymbol{g}_{j,p':}\|_2^2}{\zeta_p^2 + \zeta_{p'}^2}\right). \tag{30}$$

The column-wise RBF kernel is obtained in the same way:

$$[\boldsymbol{K}_{\mathrm{C}}(\boldsymbol{G}_i, \boldsymbol{G}_j)]_{qq'} = \gamma_q \gamma_{q'} \left(\frac{2\zeta_q \zeta_{q'}}{\zeta_q^2 + \zeta_{q'}^2} \right)^{\frac{P}{2}} \exp\left(- \frac{\|\boldsymbol{g}_{i,:q} - \boldsymbol{g}_{j,:q'}\|_2^2}{\zeta_q^2 + \zeta_{q'}^2} \right) \tag{31}$$

with the column-wise amplitude and length-scale hyperparameters $\gamma_q > 0$ and $\zeta_q > 0$.

Equations (30) and (31) are the row- and column-wise RBF kernels used in the experiments. They correspond to the special case

$$\boldsymbol{\Sigma}_{\mathrm{R}}^{(w)} = \boldsymbol{I}_{S_{\mathrm{R}}}, \qquad \boldsymbol{\Sigma}_{\mathrm{C}}^{(w)} = \boldsymbol{I}_{S_{\mathrm{C}}},$$

of the more general feature-covariance formulation in Eq. (29).

D Derivation of the Predictive Distribution

This appendix derives the exact latent-space predictor used in Section 5.3 and its lifting to the observation space.

D.1 Latent-space Gaussian Conditioning

Let $\{\boldsymbol{G}_k\}_{k=1}^N$ denote the latent matrix sequence associated with the training data $\mathcal{D}_{\mathrm{tr}}$, and let

$$\boldsymbol{K}^{(f)}(\boldsymbol{G}_i, \boldsymbol{G}_j) \in \mathbb{R}^{PQ \times PQ}$$

denote the latent-function kernel defined in Section 5.1, i.e., either $\boldsymbol{K}_{\otimes}^{(f)}(\boldsymbol{G}_i, \boldsymbol{G}_j)$ in Eq. (8) or $\boldsymbol{K}_{\oplus}^{(f)}(\boldsymbol{G}_i, \boldsymbol{G}_j)$ in Eq. (10), and let $\boldsymbol{\Sigma}^{(v)}$ denote the process-noise covariance in Eq. (6). Define the training output vector

$$\boldsymbol{y} := \begin{bmatrix} \mathrm{vec}(\boldsymbol{G}_2)^\top & \cdots & \mathrm{vec}(\boldsymbol{G}_N)^\top \end{bmatrix}^\top \in \mathbb{R}^{(N-1)PQ}, \tag{32}$$

and the block covariance matrix of the training transitions

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}^{(f)}(\boldsymbol{G}_1, \boldsymbol{G}_1) & \cdots & \boldsymbol{K}^{(f)}(\boldsymbol{G}_1, \boldsymbol{G}_{N-1}) \\ \vdots & \ddots & \vdots \\ \boldsymbol{K}^{(f)}(\boldsymbol{G}_{N-1}, \boldsymbol{G}_1) & \cdots & \boldsymbol{K}^{(f)}(\boldsymbol{G}_{N-1}, \boldsymbol{G}_{N-1}) \end{bmatrix} + \boldsymbol{I}_{N-1} \otimes \boldsymbol{\Sigma}^{(v)} \in \mathbb{R}^{(N-1)PQ \times (N-1)PQ}. \tag{33}$$

At time t, let $\boldsymbol{G}_t$ denote the current latent matrix inferred from the observation history $\mathcal{H}_t$, and define $\boldsymbol{K}_t^{(f)} = \begin{bmatrix} \boldsymbol{K}^{(f)}(\boldsymbol{G}_t, \boldsymbol{G}_1) & \cdots & \boldsymbol{K}^{(f)}(\boldsymbol{G}_t, \boldsymbol{G}_{N-1}) \end{bmatrix} \in \mathbb{R}^{PQ \times (N-1)PQ}$, $\boldsymbol{K}_{tt} = \boldsymbol{K}^{(f)}(\boldsymbol{G}_t, \boldsymbol{G}_t) + \boldsymbol{\Sigma}^{(v)}$.

Then the joint distribution of the training outputs and the next latent matrix is Gaussian:

$$\begin{bmatrix} \boldsymbol{y} \\ \mathrm{vec}(\boldsymbol{G}_{t+1}) \end{bmatrix} \,\middle|\, \boldsymbol{G}_t, \mathcal{D}_{\mathrm{tr}} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{K} & \boldsymbol{K}_t^{(f)\top} \\ \boldsymbol{K}_t^{(f)} & \boldsymbol{K}_{tt} \end{bmatrix} \right).$$

Applying the standard conditioning formula for block Gaussian distributions yields

$$\mathrm{vec}(\boldsymbol{G}_{t+1}) \mid \boldsymbol{G}_t, \mathcal{D}_{\mathrm{tr}} \sim \mathcal{N}\big(\boldsymbol{\mu}_{t+1|t},\, \boldsymbol{Q}_{t+1|t}\big)\,, \tag{34}$$

where

$$\boldsymbol{\mu}_{t+1|t} = \boldsymbol{K}_t^{(f)} \boldsymbol{K}^{-1} \boldsymbol{y}, \qquad \boldsymbol{Q}_{t+1|t} = \boldsymbol{K}_{tt} - \boldsymbol{K}_t^{(f)} \boldsymbol{K}^{-1} \boldsymbol{K}_t^{(f)\top}. \tag{35}$$

D.2 Lifting to the Observation Space

Under the observation model

$$\boldsymbol{X}_{t+1} = \boldsymbol{A}\boldsymbol{G}_{t+1}\boldsymbol{B}^\top + \boldsymbol{E}_{t+1}, \qquad \boldsymbol{E}_{t+1} \sim \mathcal{MN}\Big(\boldsymbol{O},\, \boldsymbol{\Sigma}_{\mathrm{C}}^{(e)},\, \boldsymbol{\Sigma}_{\mathrm{R}}^{(e)}\Big)\,,$$

we have

$$\mathrm{vec}(\boldsymbol{X}_{t+1}) = (\boldsymbol{B} \otimes \boldsymbol{A})\,\mathrm{vec}(\boldsymbol{G}_{t+1}) + \mathrm{vec}(\boldsymbol{E}_{t+1}). \tag{36}$$

Since $\mathrm{vec}(\boldsymbol{W}_{t+1})$ is Gaussian with covariance $\boldsymbol{\Sigma}_{\mathrm{C}}^{(e)} \otimes \boldsymbol{\Sigma}_{\mathrm{R}}^{(e)}$ and is independent of $\boldsymbol{G}_{t+1}$, substituting Eq. (34) into Eq. (36) gives

$$\mathrm{vec}(\boldsymbol{X}_{t+1}) \mid \boldsymbol{G}_t, \mathcal{D}_{\mathrm{tr}} \sim \mathcal{N}\Big((\boldsymbol{B} \otimes \boldsymbol{A})\boldsymbol{\mu}_{t+1|t},\, (\boldsymbol{B} \otimes \boldsymbol{A})\boldsymbol{Q}_{t+1|t}(\boldsymbol{B} \otimes \boldsymbol{A})^\top + \boldsymbol{\Sigma}_{\mathrm{C}}^{(e)} \otimes \boldsymbol{\Sigma}_{\mathrm{R}}^{(e)}\Big)\,.$$

D.3 Exact Marginal Predictive Law

The exact one-step predictive law at time t is obtained by integrating over the filtering distribution of the current latent matrix:

$$p(\boldsymbol{G}_{t+1} \mid \mathcal{H}_t, \mathcal{D}_{\mathrm{tr}}) = \int p(\boldsymbol{G}_{t+1} \mid \boldsymbol{G}_t, \mathcal{D}_{\mathrm{tr}})\, p(\boldsymbol{G}_t \mid \mathcal{H}_t, \mathcal{D}_{\mathrm{tr}})\, d\boldsymbol{G}_t, \tag{37}$$

and

$$p(\boldsymbol{X}_{t+1} \mid \mathcal{H}_t, \mathcal{D}_{\mathrm{tr}}) = \int p(\boldsymbol{X}_{t+1} \mid \boldsymbol{G}_t, \mathcal{D}_{\mathrm{tr}})\, p(\boldsymbol{G}_t \mid \mathcal{H}_t, \mathcal{D}_{\mathrm{tr}})\, d\boldsymbol{G}_t. \tag{38}$$

In general, both the kernel blocks $\boldsymbol{K}^{(f)}(\boldsymbol{G}_t, \boldsymbol{G}_n)$ and the conditional moments $\boldsymbol{\mu}_{t+1|t}$, $\boldsymbol{Q}_{t+1|t}$ depend on the current latent matrix $\boldsymbol{G}_t$. Therefore, even though the conditional law $p(\boldsymbol{X}_{t+1} \mid \boldsymbol{G}_t, \mathcal{D}_{\mathrm{tr}})$ is Gaussian, the marginalized law Eq. (38) is generally non-Gaussian. Equivalently, it can be viewed as a continuous Gaussian mixture induced by the filtering distribution $p(\boldsymbol{G}_t \mid \mathcal{H}_t, \mathcal{D}_{\mathrm{tr}})$.

E Details for Learning and Prediction

This appendix provides additional details for the approximate learning and prediction: the hard-EM learning procedure and the practical prediction scheme described in Section 5.4.

E.1 Complete-data Log-likelihood under the Proposed Model

Given the training time series $\mathcal{D}_{\mathrm{tr}} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$, the complete-data density factorizes as

$$p(\boldsymbol{X}_{1:N}, \boldsymbol{G}_{1:N} \mid \theta) = p(\boldsymbol{G}_1)\, p(\boldsymbol{G}_{2:N} \mid \boldsymbol{G}_{1:N-1}, \theta_{\mathrm{s}}) \prod_{k=1}^{N} p(\boldsymbol{X}_k \mid \boldsymbol{G}_k, \theta_{\mathrm{o}}),$$

where $\theta = \{\theta_{\mathrm{s}}, \theta_{\mathrm{o}}\}$, $\theta_{\mathrm{s}} = \{\theta_K, \boldsymbol{\Sigma}_{\mathrm{C}}^{(v)}, \boldsymbol{\Sigma}_{\mathrm{R}}^{(v)}\}$, $\theta_{\mathrm{o}} = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\Sigma}_{\mathrm{C}}^{(e)}, \boldsymbol{\Sigma}_{\mathrm{R}}^{(e)}\}$, and θ_K denotes the hyperparameters of the MVGP kernel in Eq. (14). The prior on $\boldsymbol{G}_1$ is a fixed standard normal, $p(\mathrm{vec}\,(\boldsymbol{G}_1)) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{PQ})$. The transition factor $p(\boldsymbol{G}_{2:N} \mid \boldsymbol{G}_{1:N-1}, \theta_{\mathrm{s}})$ is the joint density of all $N-1$ transitions under the MVGP prior. The complete-data log-likelihood is

$$\mathcal{L}_c(\theta, \boldsymbol{G}_{1:N}; \mathcal{D}_{\mathrm{tr}}) = \log p(\boldsymbol{G}_1) + \log p(\boldsymbol{G}_{2:N} \mid \boldsymbol{G}_{1:N-1}, \theta_{\mathrm{s}}) + \sum_{k=1}^{N} \log p(\boldsymbol{X}_k \mid \boldsymbol{G}_k, \theta_{\mathrm{o}}). \tag{39}$$

Under the observation model Eq. (12),

$$p(\boldsymbol{X}_k \mid \boldsymbol{G}_k, \theta_{\mathrm{o}}) = \mathcal{MN}\Big(\boldsymbol{A}\boldsymbol{G}_k\boldsymbol{B}^\top,\, \boldsymbol{\Sigma}_{\mathrm{C}}^{(e)},\, \boldsymbol{\Sigma}_{\mathrm{R}}^{(e)}\Big),$$

so that

$$\log p(\boldsymbol{X}_k \mid \boldsymbol{G}_k, \theta_{\mathrm{o}}) = -\frac{1}{2}\Big(ML\log(2\pi) + L\log|\boldsymbol{\Sigma}_{\mathrm{R}}^{(e)}| + M\log|\boldsymbol{\Sigma}_{\mathrm{C}}^{(e)}| + \mathrm{tr}\Big[(\boldsymbol{\Sigma}_{\mathrm{R}}^{(e)})^{-1}\big(\boldsymbol{X}_k - \boldsymbol{A}\boldsymbol{G}_k\boldsymbol{B}^\top\big)(\boldsymbol{\Sigma}_{\mathrm{C}}^{(e)})^{-1}\big(\boldsymbol{X}_k - \boldsymbol{A}\boldsymbol{G}_k\boldsymbol{B}^\top\big)^\top\Big]\Big).$$

Recall the training output vector $\boldsymbol{y}$ of Eq. (32) and the block covariance matrix $\boldsymbol{K}$ of Eq. (33), whose blocks are the latent-function kernels $\boldsymbol{K}^{(f)}(\boldsymbol{G}_i, \boldsymbol{G}_j)$ given by Eq. (8) or Eq. (10) and whose diagonal blocks additionally contain the process-noise covariance $\boldsymbol{\Sigma}^{(v)}$. The transition term is then the joint MVGP marginal log-likelihood of all $N-1$ transitions,

$$\log p(\boldsymbol{G}_{2:N} \mid \boldsymbol{G}_{1:N-1}, \theta_{\mathrm{s}}) = -\tfrac{1}{2}\Big((N-1)PQ\,\log(2\pi) + \log|\boldsymbol{K}| + \boldsymbol{y}^\top\boldsymbol{K}^{-1}\boldsymbol{y}\Big),$$

which is evaluated with the full block matrix $\boldsymbol{K}$, including the off-diagonal blocks $\boldsymbol{K}^{(f)}(\boldsymbol{G}_i, \boldsymbol{G}_j)$ with $i \neq j$ that couple different transitions.

E.2 Prediction via Sequential MAP Filtering

After learning, let $\hat{\boldsymbol{G}}^{\mathrm{tr}}_{1:N}$ denote the latent matrix sequence estimated by maximizing the complete-data log-likelihood Eq. (39) with the training data:

$$\hat{\boldsymbol{G}}^{\mathrm{tr}}_{1:N} = \operatorname*{argmax}_{\boldsymbol{G}_{1:N}} \mathcal{L}_c(\theta, \boldsymbol{G}_{1:N}; \mathcal{D}_{\mathrm{tr}}).$$

The training output vector and kernel matrix used by the latent GP predictor are

$$\hat{\boldsymbol{y}} := \left[\mathrm{vec}(\hat{\boldsymbol{G}}^{\mathrm{tr}}_2)^\top \quad \cdots \quad \mathrm{vec}(\hat{\boldsymbol{G}}^{\mathrm{tr}}_N)^\top\right]^\top,$$

and

$$\hat{\boldsymbol{K}} := \begin{bmatrix} \boldsymbol{K}^{(f)}(\hat{\boldsymbol{G}}^{\mathrm{tr}}_1, \hat{\boldsymbol{G}}^{\mathrm{tr}}_1) & \cdots & \boldsymbol{K}^{(f)}(\hat{\boldsymbol{G}}^{\mathrm{tr}}_1, \hat{\boldsymbol{G}}^{\mathrm{tr}}_{N-1}) \\ \vdots & \ddots & \vdots \\ \boldsymbol{K}^{(f)}(\hat{\boldsymbol{G}}^{\mathrm{tr}}_{N-1}, \hat{\boldsymbol{G}}^{\mathrm{tr}}_1) & \cdots & \boldsymbol{K}^{(f)}(\hat{\boldsymbol{G}}^{\mathrm{tr}}_{N-1}, \hat{\boldsymbol{G}}^{\mathrm{tr}}_{N-1}) \end{bmatrix} + \boldsymbol{I}_{N-1} \otimes \boldsymbol{\Sigma}^{(v)},$$

where each latent-function block is given by either Eq. (8) or Eq. (10), and the process-noise covariance $\boldsymbol{\Sigma}^{(v)}$ enters only the diagonal blocks, as in Eq. (33). The inverse-kernel setup for $\hat{\boldsymbol{K}}$ is computed once after training and reused during prediction.

During prediction, we use a sequential MAP-filtering approximation rather than reoptimizing the entire latent path. Given the previous assimilated latent state $\hat{\boldsymbol{G}}_{t-1|t-1}$, the current observation $\boldsymbol{X}_t$, and the learned parameters $\hat{\theta}$, we update only the current latent matrix by

$$\hat{\boldsymbol{G}}_{t|t} = \operatorname*{argmax}_{\boldsymbol{G}_t} \left[\log p(\boldsymbol{G}_t \mid \hat{\boldsymbol{G}}_{t-1|t-1}, \hat{\theta}_{\mathrm{s}}, \mathcal{D}_{\mathrm{tr}}) + \log p(\boldsymbol{X}_t \mid \boldsymbol{G}_t, \hat{\theta}_{\mathrm{o}})\right]. \tag{40}$$

In our implementation, this one-step MAP problem is solved by Adam (Kingma and Ba, 2015) over the PQ entries of $\boldsymbol{G}_t$, while all model parameters are fixed. For the initial time step, the latent state is initialized from the Tucker2-based observation model with the orthogonal parameters as $\boldsymbol{G}_t = \boldsymbol{A}^\top \boldsymbol{X}_t \boldsymbol{B}$. Equation (40) is the marginal MAP problem Eq. (16), and Appendix E.3 derives it from the exact filtering recursion.

The first term in Eq. (40) is evaluated by the learned latent GP transition predictor. Specifically, for the previous state $\hat{\boldsymbol{G}}_{t-1|t-1}$, define

$$\begin{aligned} \hat{\boldsymbol{K}}^{(f)}_{t-1} &= \left[\boldsymbol{K}^{(f)}(\hat{\boldsymbol{G}}_{t-1|t-1}, \hat{\boldsymbol{G}}^{\mathrm{tr}}_1) \quad \cdots \quad \boldsymbol{K}^{(f)}(\hat{\boldsymbol{G}}_{t-1|t-1}, \hat{\boldsymbol{G}}^{\mathrm{tr}}_{N-1})\right], \\ \hat{\boldsymbol{K}}_{t-1,t-1} &= \boldsymbol{K}^{(f)}(\hat{\boldsymbol{G}}_{t-1|t-1}, \hat{\boldsymbol{G}}_{t-1|t-1}) + \boldsymbol{\Sigma}^{(v)}. \end{aligned}$$

Then the one-step latent transition is approximated by

$$\mathrm{vec}(\boldsymbol{G}_t) \mid \hat{\boldsymbol{G}}_{t-1|t-1}, \mathcal{D}_{\mathrm{tr}} \sim \mathcal{N}\left(\hat{\boldsymbol{\mu}}_{t|t-1}, \hat{\boldsymbol{Q}}_{t|t-1}\right), \tag{41}$$

where

$$\hat{\boldsymbol{\mu}}_{t|t-1} = \hat{\boldsymbol{K}}^{(f)}_{t-1} \hat{\boldsymbol{K}}^{-1} \hat{\boldsymbol{y}}, \qquad \hat{\boldsymbol{Q}}_{t|t-1} = \hat{\boldsymbol{K}}_{t-1,t-1} - \hat{\boldsymbol{K}}^{(f)}_{t-1} \hat{\boldsymbol{K}}^{-1} \hat{\boldsymbol{K}}^{(f)\top}_{t-1}. \tag{42}$$

As in Eq. (35), the process noise enters Eq. (42) only through the diagonal blocks of $\hat{\boldsymbol{K}}$ and through the self-covariance $\hat{\boldsymbol{K}}_{t-1,t-1}$, whereas the cross-covariance $\hat{\boldsymbol{K}}^{(f)}_{t-1}$ is noise-free.

The predictive distribution for the next latent state is then computed by defining

$$\hat{\boldsymbol{K}}^{(f)}_t = \left[\boldsymbol{K}^{(f)}(\hat{\boldsymbol{G}}_{t|t}, \hat{\boldsymbol{G}}^{\mathrm{tr}}_1) \ \cdots \ \boldsymbol{K}^{(f)}(\hat{\boldsymbol{G}}_{t|t}, \hat{\boldsymbol{G}}^{\mathrm{tr}}_{N-1})\right], \quad \hat{\boldsymbol{K}}_{tt} = \boldsymbol{K}^{(f)}(\hat{\boldsymbol{G}}_{t|t}, \hat{\boldsymbol{G}}_{t|t}) + \boldsymbol{\Sigma}^{(v)}, \tag{43}$$

and evaluating

$$\hat{\boldsymbol{\mu}}_{t+1|t} = \hat{\boldsymbol{K}}^{(f)}_t \hat{\boldsymbol{K}}^{-1} \hat{\boldsymbol{y}}, \qquad \hat{\boldsymbol{Q}}_{t+1|t} = \hat{\boldsymbol{K}}_{tt} - \hat{\boldsymbol{K}}^{(f)}_t \hat{\boldsymbol{K}}^{-1} \hat{\boldsymbol{K}}^{(f)\top}_t. \tag{44}$$

Under the observation model, the practical predictive distribution is

$$p(\mathrm{vec}(\boldsymbol{X}_{t+1}) \mid \mathcal{H}_t, \mathcal{D}_{\mathrm{tr}}) \approx \mathcal{N}\left(\hat{\boldsymbol{m}}_{t+1}, \hat{\boldsymbol{S}}_{t+1}\right), \tag{45}$$

with

$$\hat{\boldsymbol{m}}_{t+1} = (\boldsymbol{B} \otimes \boldsymbol{A})\, \hat{\boldsymbol{\mu}}_{t+1|t}, \qquad \hat{\boldsymbol{S}}_{t+1} = (\boldsymbol{B} \otimes \boldsymbol{A})\, \hat{\boldsymbol{Q}}_{t+1|t}\, (\boldsymbol{B} \otimes \boldsymbol{A})^\top + \boldsymbol{\Sigma}^{(e)}_{\mathrm{C}} \otimes \boldsymbol{\Sigma}^{(e)}_{\mathrm{R}}. \tag{46}$$

Thus, prediction is implemented by sequentially assimilating the current observation into a one-step MAP estimate of the latent matrix and then applying the learned latent GP predictor to this MAP state.

E.3 Interpretation as Dirac Assumed-Density Filtering

We now clarify in which sense Eq. (40) approximates the exact latent filtering distribution $p(\boldsymbol{G}_t \mid \mathcal{H}_t, \mathcal{D}_{\mathrm{tr}})$ that appears in Eq. (37). Under the Markov structure of Eq. (13), the exact filtering recursion consists of a prediction step,

$$p(\boldsymbol{G}_t \mid \mathcal{H}_{t-1}, \mathcal{D}_{\mathrm{tr}}) = \int p(\boldsymbol{G}_t \mid \boldsymbol{G}_{t-1}, \mathcal{D}_{\mathrm{tr}})\, p(\boldsymbol{G}_{t-1} \mid \mathcal{H}_{t-1}, \mathcal{D}_{\mathrm{tr}})\, d\boldsymbol{G}_{t-1},$$

followed by an observation update,

$$p(\boldsymbol{G}_t \mid \mathcal{H}_t, \mathcal{D}_{\mathrm{tr}}) \propto p(\boldsymbol{X}_t \mid \boldsymbol{G}_t, \theta_{\mathrm{o}})\, p(\boldsymbol{G}_t \mid \mathcal{H}_{t-1}, \mathcal{D}_{\mathrm{tr}}).$$

Both steps are intractable here, because the transition density is a GP predictive law whose moments depend nonlinearly on the conditioning state. Our scheme replaces the previous filtering distribution by the Dirac measure $\delta(\boldsymbol{G}_{t-1} - \hat{\boldsymbol{G}}_{t-1|t-1})$. The prediction step then collapses to the Gaussian one-step transition Eq. (41), and taking the mode of this approximate filtering posterior yields Eq. (40). Prediction is therefore a sequential MAP filtering induced by a Dirac assumed-density filter, rather than a joint MAP estimate of the entire latent path $\boldsymbol{G}_{1:t}$; the trajectory-level MAP estimate is used only for the training sequence, in the E-step with Eq. (39).

Although Eq. (40) conditions explicitly only on $\hat{\boldsymbol{G}}_{t-1|t-1}$ and $\boldsymbol{X}_t$, the earlier observations are not discarded: $\hat{\boldsymbol{G}}_{t-1|t-1}$ is itself obtained by assimilating $\boldsymbol{X}_{t-1}$ into $\hat{\boldsymbol{G}}_{t-2|t-2}$. The observation history is therefore summarized recursively by a single latent-state estimate, and only the posterior mode is propagated forward. Then the spread and the possible multimodality of $p(\boldsymbol{G}_{t-1} \mid \mathcal{H}_{t-1}, \mathcal{D}_{\mathrm{tr}})$ are not carried forward.

This approximation should not be confused with a deterministic predictor. What is approximated is the latent filtering distribution, which is represented by the point estimates $\hat{\boldsymbol{G}}_{t-1|t-1}$ and $\hat{\boldsymbol{G}}_{t|t}$. What is retained is the model uncertainty evaluated at these estimates: the conditional predictive covariance $\hat{\boldsymbol{Q}}_{t+1|t}$ of the learned MVGP transition in Eq. (44), including its row- and column-wise structure and the process noise $\boldsymbol{\Sigma}^{(v)}$, together with the observation-noise covariance in Eq. (46). The predictive law in Eq. (45) is thus a genuine matrix-variate Gaussian and not a point forecast, whereas the contribution of the filtering spread to the exact mixture Eq. (38) is neglected. Finally, because only $\boldsymbol{G}_t \in \mathbb{R}^{P \times Q}$ is optimized at each step, the cost of one MAP update does not depend on t; see Appendix F. The practical effect of neglecting the filtering spread is examined empirically in Appendix G.3.5, where the MAP filter is replaced by a particle filter during prediction.

E.4 Initialization and Constrained Updates

In practice, the factor matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ and the initial latent cores $\boldsymbol{G}_{1:N}$ are initialized from a Tucker2 approximation of the training tensor obtained by stacking $\mathcal{D}_{\mathrm{tr}}$ along the temporal mode. Several algorithms, such as higher-order singular value decomposition (HOSVD) (De Lathauwer et al., 2000a) and higher-order orthogonal iteration (HOOI) (De Lathauwer et al., 2000b), can be used for constructing Tucker models. This only provides an initialization: during hard-EM, $\boldsymbol{A}$, $\boldsymbol{B}$, the latent cores, and the remaining parameters are all updated jointly by optimizing Eq. (39). Thus, the factor matrices are learned as data-adaptive components of the probabilistic state-space model rather than fixed preprocessing operators.

To make the Tucker2 projection well defined in the form used for initialization, we impose the column-orthogonality constraints

$$\boldsymbol{A}^\top \boldsymbol{A} = \boldsymbol{I}_P, \qquad \boldsymbol{B}^\top \boldsymbol{B} = \boldsymbol{I}_Q.$$

Under these constraints, for fixed $\boldsymbol{A}$ and $\boldsymbol{B}$, the least-squares Tucker2 core associated with $\boldsymbol{X}_k$ is given by

$$\boldsymbol{G}_k = \boldsymbol{A}^\top \boldsymbol{X}_k \boldsymbol{B}.$$

We use this projection to initialize the latent matrices in each E-step, while the subsequent MAP update refines $\boldsymbol{G}_{1:N}$ under the complete-data objective.

F Details for Complexity Analysis

We provide the derivation of Proposition 1. The result follows from the latent-space GP conditioning formula in Appendix D.1, the lifted predictive distribution in Appendix D.2, and the prediction via sequential MAP filtering in Appendix E.2.

Proof of Proposition 1. From Eq. (32) and Eq. (33), the latent-space training kernel matrix satisfies

$$\boldsymbol{K} \in \mathbb{R}^{(N-1)PQ\times(N-1)PQ}.$$

The inverse-kernel setup for evaluating the $\boldsymbol{K}^{-1}$-terms in Eq. (35) and Eq. (44) is implemented by matrix factorization and linear solves rather than by explicitly forming $\boldsymbol{K}^{-1}$. For a dense implementation, it needs

$$\mathcal{T}_{\text{setup}} = \mathcal{O}(N^3P^3Q^3), \qquad \mathcal{S}_{\text{setup}} = \mathcal{O}(N^2P^2Q^2).$$

After this setup, no new factorization of the training kernel is required. Since

$$\boldsymbol{K}_t^{(f)} \in \mathbb{R}^{PQ\times(N-1)PQ},$$

the predictive mean $\boldsymbol{K}_t^{(f)}\boldsymbol{K}^{-1}\boldsymbol{y}$ costs $\mathcal{O}(NP^2Q^2)$. For a full latent predictive covariance, the dominant operation is the linear solve for

$$\boldsymbol{K}_t^{(f)}\boldsymbol{K}^{-1}{\boldsymbol{K}_t^{(f)}}^{\top},$$

which costs

$$\mathcal{T}_{\text{GP}} = \mathcal{O}(N^2P^3Q^3).$$

We next consider lifting. From Eq. (46), the predictive mean can be lifted without explicitly forming $\boldsymbol{B}\otimes\boldsymbol{A}$, costing only $\mathcal{O}(MPQ + MQL)$. The dominant lifting cost arises from the full observation-space covariance in Eq. (46),

$$(\boldsymbol{B}\otimes\boldsymbol{A})\hat{\boldsymbol{Q}}_{t+1|t}(\boldsymbol{B}\otimes\boldsymbol{A})^{\top}.$$

Because

$$\boldsymbol{B}\otimes\boldsymbol{A} \in \mathbb{R}^{ML\times PQ},$$

forming the two products costs

$$\mathcal{O}\big(ML(PQ)^2\big) + \mathcal{O}\big(M^2L^2PQ\big).$$

Thus,

$$\mathcal{T}_{\text{lift}} = \mathcal{O}\big(ML(PQ)^2 + M^2L^2PQ\big) = \mathcal{O}(MLPQ(PQ+ML)).$$

The full observation-space covariance requires $\mathcal{O}(M^2L^2)$ memory if explicitly materialized.

It remains to specify the MAP-filtering term $\mathcal{T}_{\text{MAP}}$. Appendix E.2 uses sequential MAP filtering: at prediction time t, the previous state $\hat{\boldsymbol{G}}_{t-1|t-1}$ is fixed, and only $\boldsymbol{G}_t \in \mathbb{R}^{P\times Q}$ is optimized through Eq. (40). Thus, the number of optimization variables is PQ. This number is independent of t, so that $\mathcal{T}_{\text{MAP}}$ does not grow with the prediction horizon, in contrast to a trajectory-level MAP update over $\boldsymbol{G}_{1:t}$, which would involve tPQ optimization variables.

Each first-order optimization step evaluates the state term and observation term in Eq. (40). The state term uses the transition moments in Eq. (42); the dominant operation is the linear solve for

$$\hat{\boldsymbol{K}}_{t-1}^{(f)}\hat{\boldsymbol{K}}^{-1}{\hat{\boldsymbol{K}}_{t-1}^{(f)}}^{\top},$$

which costs $\mathcal{O}(N^2P^3Q^3)$. The observation term evaluates $\log p(\boldsymbol{X}_t \mid \boldsymbol{G}_t, \hat{\theta}_{\text{o}})$. With full row and column observation-noise covariances, and cached covariance factorizations, this costs

$$\mathcal{O}(MPQ + MQL + M^2L + ML^2),$$

where $MPQ + MQL$ comes from forming $\boldsymbol{A}\boldsymbol{G}_t\boldsymbol{B}^{\top}$, and $M^2L + ML^2$ from the matrix-variate normal quadratic form. The optimizer update itself costs only $\mathcal{O}(PQ)$ per step for standard first-order methods such as Adam (Kingma and Ba, 2015). Therefore, for n_{iter} first-order optimization steps,

$$\mathcal{T}_{\text{MAP}} = \mathcal{O}\big(n_{\text{iter}}\left(N^2P^3Q^3 + MPQ + MQL + M^2L + ML^2\right)\big),$$

up to lower-order terms.

Combining these terms gives

$$\mathcal{T}_{\text{pred}} = \mathcal{T}_{\text{GP}} + \mathcal{T}_{\text{lift}} + \mathcal{T}_{\text{MAP}}.$$

This proves the stated setup and per-prediction complexities. □

Note that the lifting cost reported here corresponds to forming the full observation-space covariance. If only the predictive mean or selected marginal variances are required, the corresponding cost can be reduced.

G Experimental Supplements

G.1 Common Experimental Setup

All experiments used the common hardware and software environment summarized in Table 2. This common environment was used for all compared methods in both the BEV and weather experiments to ensure a fair comparison of predictive performance and computational cost.

All methods were evaluated in a one-step-ahead rolling prediction setting. At each prediction time, the model used the available history to predict the next matrix. The reported RMSE and NLL were computed per matrix entry over all evaluated prediction times and all matrix entries. RMSE was defined as

$$\mathrm{RMSE} = \left(\frac{1}{N_{\mathrm{pred}} ML} \sum_{t=1}^{N_{\mathrm{pred}}} \|\boldsymbol{X}_t - \hat{\boldsymbol{X}}_t\|_F^2 \right)^{\frac{1}{2}},$$

where $\hat{\boldsymbol{X}}_t$ denotes the predictive mean that can be obtained as $\mathrm{vec}^{-1}(\hat{m}_t)$ from Eq. (46). NLL was defined as

$$\mathrm{NLL} = -\frac{1}{N_{\mathrm{pred}} ML} \sum_{t=1}^{N_{\mathrm{pred}}} \log p(\mathrm{vec}(\boldsymbol{X}_t) \mid \mathcal{H}_{t-1}, \mathcal{D}_{\mathrm{tr}}),$$

where N_{pred} denotes the length of the predicted sequence.

The model configurations, implementation details, and dataset protocols are summarized in Tables 3, 4, 5, and 6. Noise covariance parameters were treated as learnable parameters and estimated by maximum likelihood. For matrix-variate normal noise, a gauge constraint was imposed to remove the scale non-identifiability between the two Kronecker covariance factors. See Appendix G.3.3. Out-of-memory results indicate that the method could not complete the corresponding run under the common hardware setting. Ill-conditioned results indicate failure of the required linear algebra operation under the numerical stabilization used in the implementation.

Table 2: Hardware and software environment.

Item	Description
CPU	Intel Core i9-14900K.
System memory	128 GB RAM.
GPU	NVIDIA GeForce RTX 4090 with 24 GB memory.
Operating environment	Linux under WSL2.
Python	Python 3.11.13.
Learning framework	PyTorch 2.8.0+cu129 (Ansel et al., 2024).
GP library	GPyTorch 1.14.2 (Gardner et al., 2018).
CUDA/cuDNN	CUDA 12.9 and cuDNN 9.10.2.

Table 3: Model configurations.

Model name	Description
MV-GPSSM	MV-GPSSM with a Kronecker-sum latent kernel, row- and column-wise RBF kernels, matrix-variate normal process noise, and isotropic observation noise unless otherwise stated. The latent size was fixed to $P = 10$ and $Q = 18$. These design choices are discussed in Appendix G.3.3 and G.3.4.
Obs. MVGP	Used a Kronecker-sum structure with row- and column-wise RBF kernels (Takano and Jimbo, 2025). Inference was exact dense GP inference.
Vec. ICM-GP	Used an RBF kernel on vectorized observations combined with an ICM-type multi-task kernel $\boldsymbol{B} \otimes \boldsymbol{K}$ over the vectorized entries (Bonilla et al., 2007). The model takes $\mathrm{vec}(\boldsymbol{X}_t)$ as input and jointly predicts all entries of $\mathrm{vec}(\boldsymbol{X}_{t+1})$ without any dimensionality reduction. Inference was exact dense GP inference.
Vec. Indep-GPSSM	Vectorized the latent matrix and placed an independent GP prior on each coordinate of the transition function, so that the outputs are conditionally independent given the latent inputs. The latent vector size was fixed to $PQ = 180$.
Vec. ICM-GPSSM	Latent state-space extension of the multi-task GP with an intrinsic coregionalization model (Bonilla et al., 2007). The latent matrix was vectorized, and a shared input kernel was combined with a learnable task covariance, yielding a $\boldsymbol{B} \otimes \boldsymbol{K}$ structure. This construction subsumes the rank-one coupling obtained by pushing a single shared GP through per-coordinate transformations (Lin et al., 2024), which corresponds to a rank-one $\boldsymbol{B}$. The latent vector size was fixed to $PQ = 180$.
Vec. LMC-GPSSM	Latent state-space extension of the output-dependent GPSSM (Lin et al., 2023), which uses a linear model of coregionalization with $Q_{\mathrm{LMC}} = 4$ mixing kernels on the vectorized latent state and is therefore more flexible than the ICM variant. The latent vector size was fixed to $PQ = 180$.
Matrix SSM	Latent state-space formulation of matrix autoregression following the dynamic matrix factor model (Yu et al., 2024), with a low-rank latent matrix and a bilinear transition. The latent matrix size was fixed to $P = 10$ and $Q = 18$.
MAR	Used MAR(1) (Chen et al., 2021). The parameters were estimated by maximum likelihood using the structured-covariance MAR estimator of Chen et al. (2021), corresponding to their Section 3.3.
VAR	Used VAR(1) (Lütkepohl, 2005) on vectorized observations. The maximum-likelihood estimate was computed in closed form.

Table 4: Implementation details.

Item	Description
Kernel hyperparameter initialization	Kernel length-scales and amplitudes were initialized using the PyTorch/GPyTorch defaults.
Initialization of $\boldsymbol{A}$ and $\boldsymbol{B}$	Initialized by a Tucker2 decomposition computed with HOOI (De Lathauwer et al., 2000b). See Appendix E.4.
Initialization of $\boldsymbol{G}_{1:N}$	Given the factors $\boldsymbol{A}$ and $\boldsymbol{B}$, the initial latent matrices were computed by the Tucker projection. See Appendix E.4.
Noise-variance initialization	The observation-noise variance was initialized from the mean squared residual of the initial Tucker2 reconstruction. Other noise parameters used the default initialization.
Training objective	All probabilistic models were trained by maximizing the likelihood or the corresponding complete-data likelihood.
Hard-EM iterations	The proposed MV-GPSSM used 100 hard-EM iterations.
E-step optimization	Adam (Kingma and Ba, 2015) with 40 iterations, learning rate 10^{-3}, and weight decay 0.
M-step optimization	Adam (Kingma and Ba, 2015) with 40 iterations, learning rate 10^{-3}, and weight decay 0 with the orthogonality constraints on $\boldsymbol{A}$ and $\boldsymbol{B}$.
Parameter optimization	GP baseline hyperparameters were optimized using the same optimization framework as the proposed model. MAR and VAR parameters were estimated by the maximum-likelihood procedures described in Table 3.
NLL covariance evaluation	The reported NLL was evaluated as a joint multivariate normal density using the full predictive covariance matrix, computed by a Cholesky factorization with a small diagonal jitter.
Runtime measurement	Runtime was reported as end-to-end wall-clock total time, including both training and rolling prediction. CUDA synchronization was performed when measuring GPU runs.
Peak GPU memory	Peak GPU memory was measured using PyTorch's native CUDA memory tracking.
Out-of-memory criterion	A run was reported as OOM when it failed with `torch.cuda.OutOfMemoryError: CUDA out of memory`.

G.2 DATASETS

Table 5 and Table 6 summarize the dataset used in Section 6.1 and 6.2, respectively. The proprietary BEV experiment is not reproducible from the supplementary package because the BEV dataset cannot be redistributed under the privacy policy for BEV users. To support reproducibility, we provide the public weather experiment, implementation details, configuration files, and scripts for reproducing the reported public-data results.

Table 5: Dataset for the BEV SOC prediction task.

Item	Description
Source	Proprietary SOC records from a fleet of 800 passenger BEVs (October 1, 2024 to April 30, 2025).
Matrix	$\boldsymbol{X}_k \in \mathbb{R}^{M\times L}$ denotes the SOC matrix, where $M \in \{10, 21, 46, 100, 215, 464, 800\}$ is the number of vehicles and $L = 42$ corresponds to 4-hour samples over one week. Each entry satisfies $[\boldsymbol{X}_k]_{ij} \in [0, 1]$.
Subset selection	For each M, vehicles were selected by random subsampling from the BEV fleet.
Data split	30 weekly samples; first $N = 20$ weeks for training and remaining 10 weeks for validation (chronological split).
Preprocessing	Training-set mean subtraction in each model.

Table 6: Dataset for the weather prediction task.

Item	Description
Source	MERRA-2 reanalysis data over Western Europe (longitude -15° to 0°, latitude 42° to 58°), January 1, 2019 to May 2, 2019 (Gelaro et al., 2017; Shinde et al., 2026). The data are defined on a 33×25 latitude–longitude grid and include sea level pressure (SLP).
Matrix	$\boldsymbol{X}_k \in \mathbb{R}^{M\times L}$ denotes the SLP matrix, where rows correspond to cities and columns correspond to the most recent $L = 42$ daily observations. We use $M \in \{10, 20, 30, 40, 50\}$ cities selected from the region.
Subset selection	For the main $M = 10$ experiment in Section 6.2, we used Edinburgh, Glasgow, Manchester, Dublin, Galway, London, Rennes, Nantes, A Coruña, and Bilbao. For the experiments in Appendix G.3.2 with $M = 10, \ldots, 50$, cities were selected randomly from the city list visualized in Fig. 10.
Data split	The 122 daily grids from January 1, 2019 to May 2, 2019 were converted into $L = 42$-day windows with a stride of one day, which yields 80 consecutive one-step-ahead pairs $(\boldsymbol{X}_k, \boldsymbol{X}_{k+1})$. The first 40 pairs were used for training and the remaining 40 for validation.
Preprocessing	Min–max normalization was applied so that each entry satisfies $[\boldsymbol{X}_k]_{ij} \in [0, 1]$, followed by training-set mean subtraction in each model.

G.3 ADDITIONAL EXPERIMENTAL RESULTS

G.3.1 FULL RESULTS AND INTERPRETABILITY ON BEV FLEET DATA

This section provides full results and interpretability analysis for the BEV fleet data, complementing Section 6.1.

Predictive performance. Figure 8 shows representative predictions for all models. The proposed method captures both highly varying and nearly constant SOC trajectories, reflecting heterogeneous vehicle-level behaviors. The predictive band adapts to each vehicle's variability, widening for the highly varying Vehicle 1 and narrowing for the near-constant Vehicle 2, while the predictive mean represents the central tendency rather than every fluctuation. Tables 7 and 8 show the complete RMSE and NLL validation results for all models and fleet sizes. Compared with the latent-space GP baselines, MV-GPSSM attains lower NLL at every fleet size. Moreover, Vec. ICM-GPSSM and the more flexible Vec. LMC-GPSSM perform nearly identically in both RMSE and NLL, suggesting that output-kernel flexibility has little effect on the performance of these vectorized models.

Table 7: Full validation RMSE per entry for the BEV SOC prediction task. Each cell reports the mean and standard deviation across five seeds. OOM denotes an out-of-memory failure.

Model	$M = 10$	$M = 21$	$M = 46$	$M = 100$	$M = 215$	$M = 464$	$M = 800$
MV-GPSSM (ours)	**0.223 (0.074)**	0.230 (0.038)	**0.211 (0.011)**	**0.225 (0.008)**	**0.231 (0.006)**	**0.230 (0.008)**	**0.230 (0.001)**
Obs. MVGP	0.231 (0.079)	**0.217 (0.046)**	0.229 (0.019)	OOM	OOM	OOM	OOM
Vec. ICM-GP	0.252 (0.084)	0.244 (0.053)	0.260 (0.043)	OOM	OOM	OOM	OOM
Vec. Indep-GPSSM	0.325 (0.101)	0.276 (0.082)	0.263 (0.032)	0.245 (0.021)	0.243 (0.021)	0.238 (0.010)	0.239 (0.002)
Vec. ICM-GPSSM	0.241 (0.077)	0.228 (0.045)	0.219 (0.013)	0.233 (0.011)	0.238 (0.005)	0.239 (0.008)	0.238 (0.001)
Vec. LMC-GPSSM	0.241 (0.080)	0.226 (0.042)	0.217 (0.013)	0.233 (0.011)	0.239 (0.008)	0.239 (0.008)	0.238 (0.002)
Matrix SSM	0.291 (0.098)	0.295 (0.057)	0.280 (0.026)	0.273 (0.022)	0.280 (0.028)	0.266 (0.010)	0.256 (0.003)
MAR	0.843 (0.791)	0.439 (0.276)	0.487 (0.092)	0.463 (0.259)	0.266 (0.017)	0.263 (0.009)	0.261 (0.002)
VAR	2.732 (2.210)	12.3 (15.2)	57.7 (48.5)	17.1 (8.6)	149 (118)	661 (700)	OOM

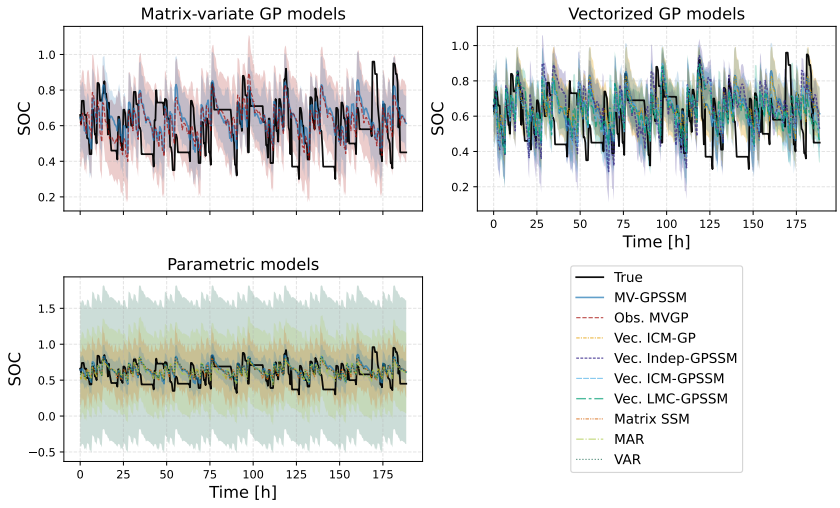
Table 8: Full validation NLL per entry for the BEV SOC prediction task. Each cell reports the mean and standard deviation across five seeds. OOM denotes an out-of-memory failure.

(a) Small-to-medium fleet sizes.

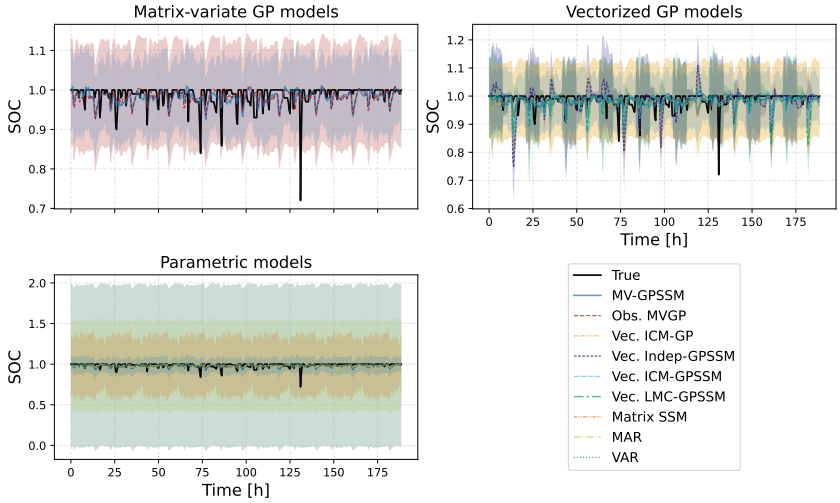
Model	$M = 10$	$M = 21$	$M = 46$	$M = 100$
MV-GPSSM (ours)	**-0.547 (0.227)**	0.087 (0.253)	-0.297 (0.205)	**-0.437 (0.148)**
Obs. MVGP	-0.348 (0.812)	**-0.457 (0.344)**	**-0.505 (0.158)**	OOM
Vec. ICM-GP	0.395 (0.935)	0.240 (0.459)	0.240 (0.289)	OOM
Vec. Indep-GPSSM	$2.85\ (1.24) \times 10^4$	$2.34\ (1.11) \times 10^4$	$2.78\ (0.75) \times 10^4$	$2.45\ (0.52) \times 10^4$
Vec. ICM-GPSSM	$1.11\ (0.99) \times 10^4$	$1.60\ (0.60) \times 10^4$	$1.83\ (0.20) \times 10^4$	$1.97\ (0.99) \times 10^4$
Vec. LMC-GPSSM	$1.11\ (0.99) \times 10^4$	$1.60\ (0.60) \times 10^4$	$1.83\ (0.19) \times 10^4$	$2.11\ (0.72) \times 10^4$
Matrix SSM	$5.22\ (0.82) \times 10^2$	$1.46\ (0.69) \times 10^4$	$1.74\ (0.17) \times 10^4$	$2.27\ (0.16) \times 10^4$
MAR	0.422 (0.038)	0.274 (0.040)	0.150 (0.032)	0.065 (0.041)
VAR	0.954 (0.022)	0.950 (0.013)	0.952 (0.006)	0.952 (0.005)

(b) Large-scale fleet sizes.

Model	$M = 215$	$M = 464$	$M = 800$
MV-GPSSM (ours)	**-0.301 (0.258)**	**-0.205 (0.301)**	**-0.158 (0.012)**
Obs. MVGP	OOM	OOM	OOM
Vec. ICM-GP	OOM	OOM	OOM
Vec. Indep-GPSSM	$2.52\ (0.89) \times 10^4$	$2.34\ (0.99) \times 10^4$	$2.32\ (0.69) \times 10^4$
Vec. ICM-GPSSM	$1.66\ (1.01) \times 10^4$	$6.07\ (8.00) \times 10^3$	$4.53\ (4.28) \times 10^3$
Vec. LMC-GPSSM	$1.19\ (1.29) \times 10^4$	$6.65\ (7.75) \times 10^3$	$4.28\ (3.98) \times 10^3$
Matrix SSM	$2.44\ (0.12) \times 10^4$	$2.49\ (0.15) \times 10^4$	$2.52\ (0.17) \times 10^4$
MAR	0.030 (0.030)	0.044 (0.021)	0.048 (0.001)
VAR	0.952 (0.002)	0.951 (0.001)	OOM



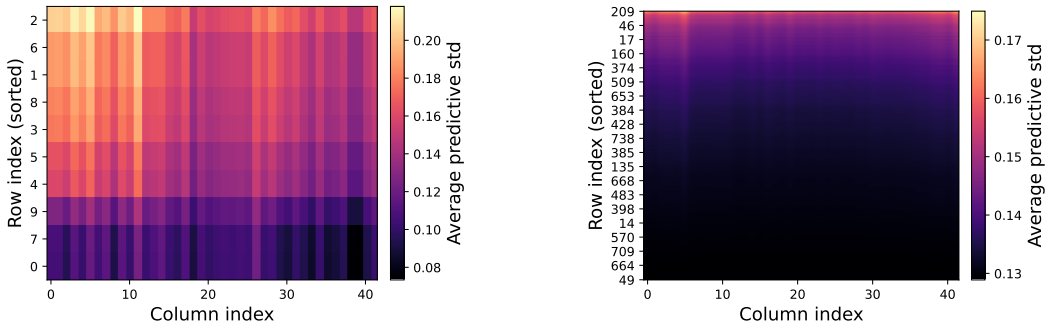
(a) Vehicle 1 with large SOC variation.



(b) Vehicle 2 with small SOC variation.

Figure 8: Representative prediction examples with mean and marginal $\pm 1\sigma$ ranges for two vehicles exhibiting (a) large and (b) small SOC variations in the 10-vehicle BEV task. In each panel, MV-GPSSM is repeated in all three subplots as a common reference, and the remaining methods are grouped as follows: the top-left subplot shows Obs. MVGP, the top-right subplot shows Vec. ICM-GP, Vec. Indep-GPSSM, Vec. ICM-GPSSM, and Vec. LMC-GPSSM, and the bottom-left subplot shows Matrix SSM, MAR, and VAR.

Interpretability. Figure 9 visualizes the time-averaged predictive standard deviation for both the 10-vehicle and 800-vehicle settings. In both cases, the uncertainty varies across vehicles and time indices, indicating that the structured matrix model captures vehicle-wise heterogeneity together with temporal variation in prediction difficulty. Notably, this structured pattern persists even at larger scales, demonstrating that the learned uncertainty remains informative as the number of vehicles increases.



(a) 10 vehicles

(b) 800 vehicles

Figure 9: Time-averaged predictive standard deviation produced by the proposed MV-GPSSM for the BEV SOC prediction task. Rows correspond to vehicles (sorted by average predictive standard deviation), and columns correspond to the 42 time indices within the weekly horizon. The variation along both axes indicates that the structured matrix model captures both vehicle-wise heterogeneity and time-wise structure.

G.3.2 Predictive Performance and Scalability on Public Weather Data

This section provides additional results on predictive performance and scalability for the public weather data, complementing Section 6.2.

We use public MERRA-2 reanalysis data over Western Europe (Gelaro et al., 2017; Shinde et al., 2026), covering longitudes from -15° to 0° and latitudes from 42° to 58°, sampled daily from January 1 to May 2, 2019. The data are defined on a 33×25 latitude–longitude grid and include sea level pressure (SLP). We select up to $M = 50$ representative cities from this region (Fig. 10)—extending beyond the $M = 10$ used in the main text—and construct matrix-valued time series $\boldsymbol{X}_k$ using rolling 42-day windows ($L = 42$) for one-step-ahead prediction.

Predictive performance. Table 9 reports RMSE and NLL for all methods as the number of cities increases. The proposed MV-GPSSM achieves the best NLL at every tested size and also attains comparable RMSE. In contrast, MAR yields relatively high NLL despite reasonable RMSE at small scales, and VAR performs poorly in both metrics. The added latent-space baselines, namely Vec. Indep-GPSSM, Vec. ICM-GPSSM, Vec. LMC-GPSSM, and Matrix SSM, attain moderate RMSE but markedly worse NLL, indicating that their joint predictive covariances are overconfident even when the point predictions are reasonable. As in the BEV experiment, Vec. ICM-GPSSM and Vec. LMC-GPSSM perform nearly identically. Obs. MVGP and Vec. ICM-GP become infeasible as the problem size grows. These results demonstrate that the proposed latent-space modeling preserves predictive accuracy while improving probabilistic quality. Figure 11 shows representative one-step predictions for two cities (Edinburgh and Bilbao). The proposed method captures the main temporal variations of normalized SLP while providing predictive uncertainty, whereas the baselines tend to produce flatter trajectories.

Table 9: Validation metrics for each model and number of cities in the weather prediction task. Each cell reports mean RMSE (left) and NLL (right) per entry. OOM indicates an out-of-memory failure, and Ill-cond. indicates an ill-conditioned failure.

Number of cities (rows)	10	20	30	40	50
MV-GPSSM (ours)	0.083 / **-1.187**	**0.080 / -1.386**	**0.082 / -1.598**	**0.080 / -1.691**	**0.083 / -1.735**
Obs. MVGP	0.102 / -0.930	0.089 / -0.960	0.089 / -0.978	0.089 / -0.989	OOM
Vec. ICM-GP	0.135 / 0.720	0.133 / 0.753	OOM	OOM	OOM
Vec. Indep-GPSSM	$0.113 / 1.41 \times 10^3$	$0.113 / 2.54 \times 10^3$	$0.114 / 3.12 \times 10^3$	$0.112 / 3.34 \times 10^3$	$0.114 / 3.64 \times 10^3$
Vec. ICM-GPSSM	$0.121 / 1.41 \times 10^3$	$0.125 / 2.51 \times 10^3$	$0.129 / 3.01 \times 10^3$	$0.130 / 3.36 \times 10^3$	$0.132 / 3.63 \times 10^3$
Vec. LMC-GPSSM	$0.123 / 1.47 \times 10^3$	$0.125 / 2.54 \times 10^3$	$0.127 / 3.11 \times 10^3$	$0.128 / 3.33 \times 10^3$	$0.126 / 3.63 \times 10^3$
Matrix SSM	$0.149 / 2.04 \times 10^3$	$0.153 / 2.34 \times 10^3$	$0.151 / 2.30 \times 10^3$	$0.150 / 2.16 \times 10^3$	$0.151 / 2.13 \times 10^3$
MAR	**0.080** / 0.361	0.096 / 0.201	0.122 / 0.110	Ill-cond.	Ill-cond.
VAR	0.197 / 0.929	5.874 / 0.929	10.562 / 0.929	9.087 / 0.929	8.650 / 0.929

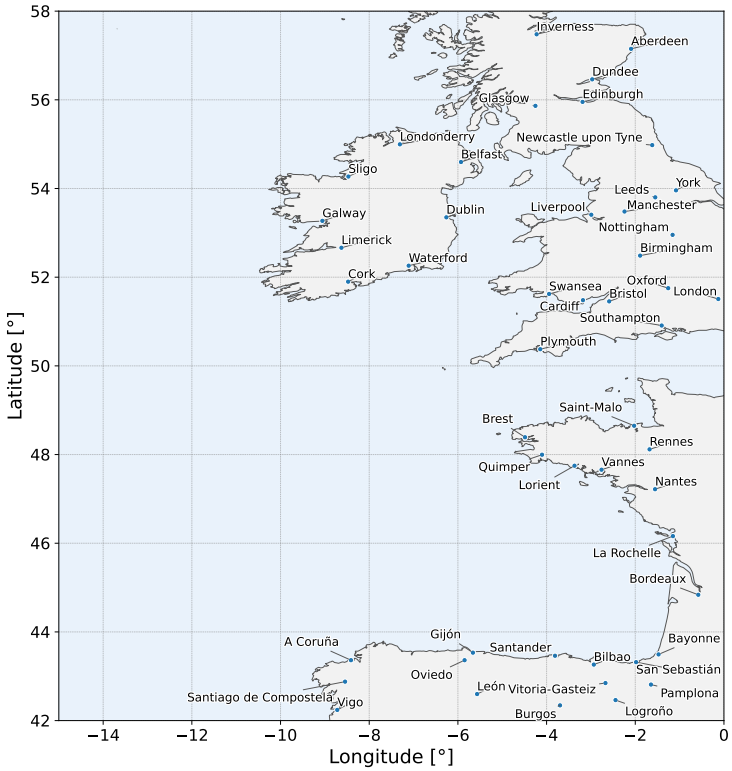


Figure 10: Extracted 50 cities from MERRA-2 reanalysis data over Western Europe. The basemap was made with Natural Earth (2026).

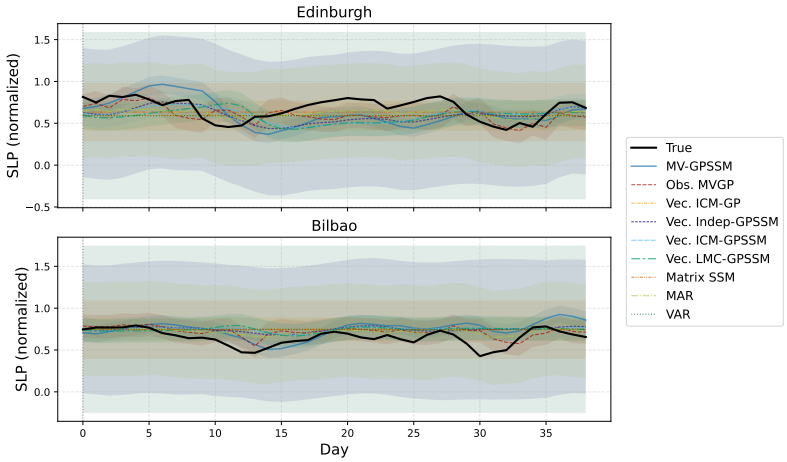
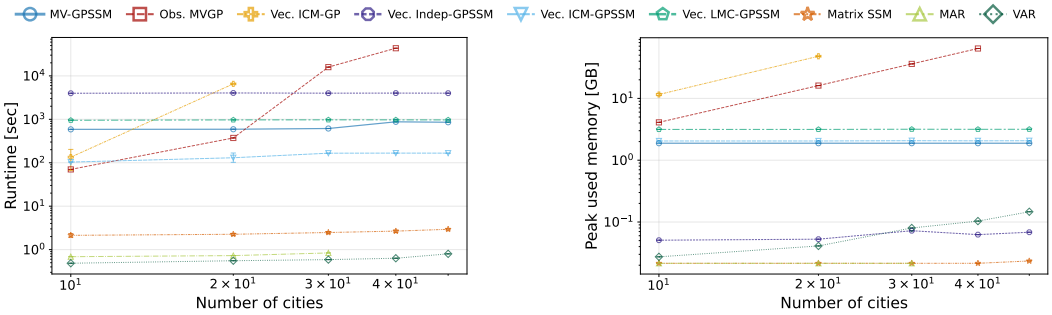


Figure 11: Representative prediction examples with mean and marginal $\pm 1\sigma$ ranges for two cities in the 10-city weather task.

Scalability. Figure 12 shows runtime and peak memory usage as functions of the number of cities. The proposed method remains nearly constant in both runtime and memory usage, consistent with the theoretical complexity analysis based on the latent matrix representation. In contrast, observation-space GPs such as Obs. MVGP and Vec. ICM-GP scale poorly, with rapidly increasing runtime and memory usage, becoming infeasible at larger sizes.

Overall, the weather experiment confirms that the proposed MV-GPSSM maintains strong predictive performance, delivers superior probabilistic quality, and scales efficiently to larger matrix sizes, demonstrating that the approach generalizes beyond the BEV application.



(a) Runtime

(b) Peak memory usage

Figure 12: Scalability for the SLP prediction task with respect to the number of cities M.

G.3.3 Design Choices in the Experiments

This subsection provides additional ablation studies for the experimental design choices used in Section 6.

We evaluate key design choices of the proposed model on the BEV dataset with $M = 100$ vehicles, including kernel structure, noise modeling, and training strategy. Unless otherwise specified, all variants use the same configuration as the main experiments: a Kronecker-sum kernel structure, matrix-variate normal process noise with isotropic observation noise, joint optimization of θ_{s} and θ_{o}, and latent dimensions $P = 10$ and $Q = 18$ (see Appendix G.3.4 for the latent-dimension selection procedure). Each ablation changes only the component under examination with a single seed.

Kronecker-sum and Kronecker-product kernels. Table 10 compares the two structured kernel variants introduced in Section 5.1. The Kronecker-sum variant yields better validation performance than the Kronecker-product variant and is therefore used in the main results. Since the two forms are derived from the interaction structure of the latent dynamics rather than being posited, this selection amounts to selecting the additive interaction of Eq. (4) over the multiplicative one of Eq. (3).

Table 10: Ablation study comparing Kronecker-sum and Kronecker-product kernel variants (BEV, $M = 100$, single seed).

Variant	RMSE per entry	NLL per entry
Kronecker-sum	**0.222**	**-0.425**
Kronecker-product	0.234	-0.120

Structured and isotropic noise. We compared where to introduce structured covariance in the noise terms of the proposed MV-GPSSM. The process noise $\boldsymbol{V}_t$ and observation noise $\boldsymbol{E}_t$ are defined in Eq. (5) and Eq. (12), respectively. In this ablation, Iso. denotes isotropic noise, i.e.,

$$\mathrm{vec}(\boldsymbol{V}_t) \sim \mathcal{N}(\mathbf{0}, \sigma_v^2 \boldsymbol{I}_{PQ}), \qquad \mathrm{vec}(\boldsymbol{E}_t) \sim \mathcal{N}(\mathbf{0}, \sigma_e^2 \boldsymbol{I}_{ML}),$$

which assigns the same variance to all entries. MVN denotes matrix-variate normal noise with diagonal row and column covariance factors,

$$\mathrm{Cov}(\mathrm{vec}(\boldsymbol{E})) = \boldsymbol{\Sigma}_C \otimes \boldsymbol{\Sigma}_R,$$

where $\mathbf{\Sigma}_R$ and $\mathbf{\Sigma}_C$ are diagonal. Thus, MVN noise captures row- and column-wise heterogeneity without introducing a full covariance matrix over all entries. We used diagonal covariance factors and did not assign MVN noise to both $\boldsymbol{V}_t$ and $\boldsymbol{E}_t$, so that the ablation isolates where structured uncertainty is most useful while avoiding overlapping structured noise sources. For MVN noise, we fixed the scale ambiguity between the two Kronecker factors by normalizing the average row variance.

Table 11 reports the results. The fully isotropic variant provided a baseline without structured row- or column-wise noise covariance. When the matrix-variate normal covariance was assigned only to the observation noise, both RMSE and NLL degraded. In contrast, assigning the matrix-variate normal covariance to the process noise yielded the best RMSE and NLL. These results indicate that, in this setting, structured uncertainty is more effectively represented in the process noise $\boldsymbol{V}_t$ than in the observation noise $\boldsymbol{E}_t$.

Table 11: Ablation study on noise covariance models (BEV, $M = 100$, single seed).

Noise model	RMSE per entry	NLL per entry
Iso. process, Iso. obs. noise	0.231	-0.413
Iso. process, MVN obs. noise	0.253	20.8
MVN process, Iso. obs. noise	**0.222**	**-0.425**

Joint optimization and two-stage procedure. Table 12 compares joint optimization and a two-stage procedure in which the observation-model parameters θ_{o} are fixed to their HOOI-based initialization (Appendix E.4) before only $G_{1:N}$ and θ_s are updated. Jointly optimizing the factor matrices substantially improves NLL with comparable RMSE compared with the two-stage variant, indicating more effective learning of the latent representation.

Table 12: Ablation study comparing joint optimization and two-stage procedures (BEV, $M = 100$, single seed).

Variant	RMSE per entry	NLL per entry
Joint optimization	**0.222**	**-0.425**
Two-stage procedure	0.224	-0.028

G.3.4 Latent Size Selection

This subsection provides additional details on the selection of the latent dimensions P and Q used in the experiments of Section 6.

The latent dimensions P and Q determine the size of the reduced matrix state $\boldsymbol{G}_t \in \mathbb{R}^{P\times Q}$. Their choice is task-dependent: larger latent dimensions reduce the representation error of the Tucker2 approximation, but increase the computational cost of latent-space GP inference, whose dominant terms scale polynomially in PQ. Thus, P and Q should be selected by balancing representation accuracy and computational tractability.

As a practical diagnostic for the representation–computation trade-off, we examined the relative reconstruction error of the Tucker2 approximation defined in Eq. (1). For each candidate pair (P, Q), we compute the Tucker2 approximation as Eq. (1) and evaluate relative reconstruction error

$$\epsilon_{\mathrm{rec}}(P,Q) = \left(\frac{\sum_{k=1}^{N} \left\| \boldsymbol{X}_k - \boldsymbol{A}\boldsymbol{G}_k\boldsymbol{B}^\top \right\|_F^2}{\sum_{k=1}^{N} \left\| \boldsymbol{X}_k \right\|_F^2} \right)^{\frac{1}{2}}.$$

This quantity measures how accurately the matrix sequence can be represented with a latent matrix of size $P \times Q$. We use it only as a diagnostic for the representation–computation trade-off, not as an automatic rank-selection criterion.

Figure 13 shows the relative reconstruction-error landscape $\epsilon_{\mathrm{rec}}(P, Q)$ for different Tucker2 latent dimensions on the BEV dataset with $M = 800$. The landscape reveals a clear trade-off between latent dimensionality and representation accuracy: increasing either P or Q reduces the reconstruction

error, while larger latent dimensions increase the computational cost of latent-space GP inference through PQ.

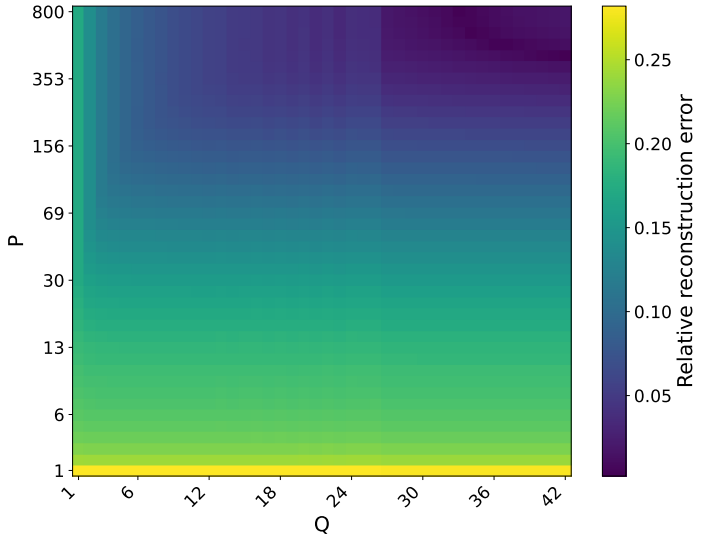


Figure 13: Relative reconstruction-error landscape for different Tucker2 latent dimensions (P, Q) on the BEV dataset with $M = 800$.

In the BEV experiments, the number of vehicles varied from $M = 10$ to $M = 800$. We therefore fixed $P = 10$, the smallest row dimension used in the scalability study, so that the same latent row dimension could be used across all vehicle-count settings. We set $Q = 18$ by considering both the reconstruction-error landscape and the resulting computation time. Automatic selection of P and Q is beyond the scope of this paper.

G.3.5 Diagnostic of Latent-State Uncertainty

Prediction in the proposed model uses the sequential MAP filtering of Appendix E.2, which propagates only the mode of the latent filtering distribution, as formalized in Appendix E.3. This subsection reports a diagnostic experiment that quantifies how much this approximation affects the predictive distribution.

During prediction, we replace the sequential MAP filtering in Eq. (40) with a particle filter (PF). For each particle, the transition distribution in Eq. (41) is computed using the particle state $G_{t-1}^{(i)}$ instead of the previous MAP estimate $\hat{G}_{t-1|t-1}$, and combined with the observation likelihood to form the locally optimal proposal $p(G_t \mid G_{t-1}^{(i)}, X_t)$. Particle weights are updated using the predictive observation likelihood, followed by systematic resampling (Doucet et al., 2000). The Gaussian predictive distributions obtained from Eqs. (43)–(46) for all particles are combined into a Gaussian mixture, whose mean and covariance are used to evaluate RMSE and NLL. We use $M = 10$ and $N_p \in \{10, 100\}$.

As summarized in Table 13, propagating the latent filtering posterior by PF yields almost identical RMSE on both datasets and an NLL that is not improved relative to MAP filtering, while the state-estimation time increases by roughly one order of magnitude for $N_p = 100$. Increasing the number of particles from 10 to 100 does not change this trend.

To further examine this observation, we quantify how much latent uncertainty remains after the observation update by comparing the filtering posterior covariance $\boldsymbol{\Sigma}_{\mathrm{post},t}$, estimated from the particle representation, with the one-step predictive covariance $\hat{\boldsymbol{Q}}_{t|t-1}$ of Eq. (42) through

$$r_{\mathrm{trace}} := \frac{\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{post},t})}{\mathrm{tr}\left(\hat{\boldsymbol{Q}}_{t|t-1}\right)}, \qquad r_{\max} := \max_{\boldsymbol{v}\neq\boldsymbol{0}} \frac{\boldsymbol{v}^\top \boldsymbol{\Sigma}_{\mathrm{post},t} \boldsymbol{v}}{\boldsymbol{v}^\top \hat{\boldsymbol{Q}}_{t|t-1} \boldsymbol{v}} = \lambda_{\max}\left(\hat{\boldsymbol{Q}}_{t|t-1}^{-1/2} \boldsymbol{\Sigma}_{\mathrm{post},t} \hat{\boldsymbol{Q}}_{t|t-1}^{-1/2}\right) \tag{47}$$

Table 13: Predictive performance and computational cost of prediction-time MAP filtering and particle filtering (PF) at $M = 10$. Values are the mean and standard deviation over seeds (five model seeds, and three particle-sampling seeds for PF). The state-estimation time is the wall-clock time of the state-estimation step during prediction, and N_p denotes the number of particles.

Task	Method	RMSE per entry	NLL per entry	State est. time [s]
BEV	MAP	0.223 (0.074)	-0.547 (0.227)	6.9 (1.3)
	PF ($N_p = 10$)	0.223 (0.068)	-0.516 (0.262)	9.0 (1.8)
	PF ($N_p = 100$)	0.223 (0.068)	-0.521 (0.252)	90.0 (15.6)
Weather	MAP	0.083 (0.005)	-1.187 (0.297)	121.5 (2.8)
	PF ($N_p = 10$)	0.083 (0.003)	-1.131 (0.111)	188.1 (13.4)
	PF ($N_p = 100$)	0.083 (0.003)	-1.130 (0.111)	1726.4 (134.4)

where $\lambda_{\max}(\cdot)$ denotes the largest eigenvalue. The first quantity in Eq. (47) compares the total posterior variance with the total one-step predictive variance, whereas the second gives the largest remaining variance ratio over all directions. Values substantially below one indicate that only a small fraction of the one-step predictive uncertainty remains after the observation update, so that the Dirac approximation discards little uncertainty.

Table 14: Ratios Eq. (47) of the latent filtering posterior covariance to the one-step predictive covariance, estimated from the particle filter at $M = 10$. Values are the mean and standard deviation over seeds.

Task	Method	r_{trace}	$r_{\max}$
BEV	PF ($N_p = 10$)	$6.08\ (3.62) \times 10^{-3}$	0.042 (0.035)
	PF ($N_p = 100$)	$5.97\ (3.78) \times 10^{-3}$	0.083 (0.068)
Weather	PF ($N_p = 10$)	$3.52\ (1.88) \times 10^{-2}$	0.356 (0.190)
	PF ($N_p = 100$)	$3.77\ (1.87) \times 10^{-2}$	0.570 (0.230)

Table 14 shows that the filtering posterior is highly concentrated overall, retaining less than $1\ \%$ of the total one-step predictive variance on the BEV data and less than $4\ \%$ on the weather data. Some uncertainty nevertheless remains along the least informative directions, where $r_{\max}$ reaches 0.57 on the weather data. Propagating this residual uncertainty by PF still changes the RMSE and the NLL only negligibly, which indicates that the latent uncertainty omitted by the MAP approximation has a limited influence on the predictive distribution in the evaluated settings. A plausible reason is that each latent matrix $\boldsymbol{G}_t \in \mathbb{R}^{P \times Q}$ is constrained by ML observed entries through the linear observation model Eq. (12), so that the observation update is highly informative relative to the transition prior.

These results do not imply that MAP filtering is theoretically equivalent to full posterior propagation, and the conclusion is restricted to the datasets and horizons considered here. They do indicate, however, that the point approximation of the current latent state is empirically adequate at a substantially lower computational cost, while the predictive uncertainty of the learned transition and observation models is retained as described in Appendix E.3. Propagating the full latent posterior by variational or particle-based inference, and learning the model under such an inference scheme, remain directions for future work.

H Model Documentation

This appendix provides a concise, structured description of the proposed model and its intended scope, following the spirit of model documentation frameworks such as Model Cards, while reflecting the nature of this work as a methodological contribution.

H.1 Model Overview

The proposed model is a matrix-variate Gaussian process state-space model (MV-GPSSM) with structured latent dynamics. It represents a matrix-valued time series through a sequence of low-

dimensional latent matrices obtained via a Tucker2-style factorization, and models their temporal evolution using a matrix-variate Gaussian process with structured row and column kernels. The primary purpose of the model is to study how probabilistic matrix-valued dynamics can be modeled efficiently by transferring inference from the observation space to a reduced latent matrix space.

H.2 Intended Use

The model is intended for research and experimental evaluation of probabilistic prediction methods for matrix-valued time series. Typical use cases include:

- Benchmarking scalable probabilistic models for structured spatiotemporal data.
- Analyzing the trade-off between predictive uncertainty, structural interpretability, and computational efficiency.
- Studying latent matrix representations and structured GP dynamics.

In this work, the model is evaluated on battery electric vehicle state-of-charge (SOC) prediction, and sea level pressure (SLP) prediction on public weather data to illustrate these aspects.

H.3 Out-of-Scope Use

The proposed model is not intended to be deployed directly in safety-critical or decision-making systems without further validation and adaptation. In particular:

- The model is not presented as a production-ready forecasting system.
- No claims are made regarding fairness, robustness to distribution shift, or safety guarantees.
- The model should not be used as-is for real-time control, policy-making, or automated decision processes involving human subjects.

H.4 Model Availability

This paper does not release trained model parameters or pretrained predictors. Instead, we release the implementation of the proposed modeling framework, including training and inference code, to support reproducibility and independent evaluation. Users are expected to train models on their own datasets and to assess suitability for their specific applications.

H.5 Data and Evaluation Context

The experimental results reported in the paper are obtained on:

- Proprietary fleet-scale battery electric vehicle data, used solely for research evaluation.
- Publicly available weather data.

Performance characteristics reported in the paper are specific to these datasets, kernel choices, and experimental protocols, and should not be interpreted as universal guarantees.

H.6 Technical Limitations and Known Assumptions

The proposed approach relies on several assumptions and design choices:

- The existence of a low-dimensional latent matrix representation that captures essential dynamics.
- GP priors with kernel structures that encode row-wise and column-wise similarities.
- Approximate learning and prediction based on MAP estimation in a hard-EM framework.

If these assumptions are violated, predictive performance and uncertainty quantification may degrade.

H.7 SUMMARY

In summary, this work introduces a methodological framework for scalable probabilistic modeling of matrix-valued time series. The provided code is intended to facilitate research reproducibility and further methodological development, rather than direct deployment of trained models. Researchers and practitioners are encouraged to carefully evaluate the model assumptions and limitations in any downstream application.

I SOCIETAL IMPACT AND LIMITATIONS

This work presents a methodological contribution for probabilistic prediction of matrix-valued time series. Although the paper does not itself propose a decision-making or control policy, predictions from the model could be used in downstream operational settings such as fleet energy management. In such settings, distribution shift, model misspecification, or under/overconfident uncertainty estimates may lead to unreliable decisions if the model is used without task-specific validation and human oversight. The experiments use previously collected proprietary BEV data and public weather data, and no new human-subject data are collected. Any practical deployment on mobility-related or user-related data would require additional assessment of privacy, robustness, and subgroup reliability.