Proceedings of Machine Learning Research 343:677–696, 2026 Probabilistic Graphical Models (PGM)

A Factor Graph Approach to Scalable Multi-Output Gaussian Process Regression

Wouter W. L. Nuijten W.W.L.NUIJTEN@TUE.NL
Eindhoven University of Technology & Lazy Dynamics B.V., the Netherlands

Esther G. van Pelt E.G.V.PELT@TUE.NL
Eindhoven University of Technology, Eindhoven, the Netherlands

Albert Podusenko ALBERT@LAZYDYNAMICS.COM
İsmail Şenöz ISENOZ@LAZYDYNAMICS.COM
Lazy Dynamics B.V., the Netherlands

Wouter M. Kouw W.M.KOUW@TUE.NL
Eindhoven University of Technology, Eindhoven, the Netherlands

Editors: Gustau Camps-Valls, Manuele Leonelli and Gherardo Varando

Abstract

Multi-output Gaussian process regression scales cubically in the number of observations times outputs, and dense kernel-matrix methods need bespoke handling whenever different outputs are observed at different inputs. We express multi-output Gaussian process regression as a Forney-style factor graph in which a nearest-neighbor chain orders a fixed candidate set of C inputs into a one-dimensional sequence. Along this chain, latent Matérn processes evolve through linear-Gaussian transition factors, while the linear model of coregionalization mixes L latent processes into D outputs through a deterministic mixing factor and per-output scalar observation factors. Posterior computation reduces to exact Gaussian message passing on the chain at cost $\mathcal{O}(C(DL^2 + L^3))$ after chain construction, and missing observations omit their local factor without any covariance-matrix restructuring. The formulation therefore scales in the number of data samples and in the rate of missing observations, while remaining best suited to candidate sets in low input dimension. We compare the factor-graph formulation against an exact kernel-matrix baseline, a sparse-variational inducing-point baseline, and a nearest-neighbor baseline on a synthetic input-dimension sweep and on electricity time series forecasting. At low input dimension the factor-graph posterior tracks the exact kernel-matrix posterior closely, and the gap grows gradually as input dimension increases while staying competitive with both approximate baselines. On the electricity time series our factor-graph formulation matches all three baselines in forecast accuracy while scaling linearly in the number of data points, where the exact kernel-matrix method becomes infeasible and the inducing-point baseline remains substantially slower.

Keywords: Factor graphs, Gaussian processes, Message passing, Partial observations

1. Introduction

Probabilistic graphical models provide a unified language for inference: belief propagation, Kalman filtering and smoothing, hidden Markov model inference, and variational message passing all arise as instances of message passing on specific factor graphs (Loeliger et al., 2007; Şenöz et al., 2021). Gaussian processes (GPs) are a notable outlier. Defined by a kernel rather than a factorization, GPs are typically treated as monolithic objects, which do

© 2026 W.W.L. Nuijten, E.G. van Pelt, A. Podusenko, İ. Şenöz & W.M. Kouw.

not scale well and have poor compositionality. In the multi-output case, fitting a standard GP with D correlated outputs and N training points requires operations on a $DN \times DN$ covariance matrix, i.e., $\mathcal{O}((DN)^3)$, which becomes prohibitive as D or N grows (Álvarez et al., 2012). Equally inconvenient, when different outputs are observed at different inputs, as is the norm in sensor networks, environmental monitoring, and clinical time series, dense kernel-matrix implementations typically handle these partial observations by restructuring or slicing the covariance matrix rather than making a local edit to a graph.

But non-monolithic treatments of GPs do exist. A GP with a Matérn covariance function can be expressed as the solution of a linear stochastic differential equation (SDE) (Hartikainen and Särkkä, 2010; Särkkä and Solin, 2019), and the resulting linear-Gaussian state-space model has an exact representation as a Forney-style factor graph (FFG) on which Kalman filtering and smoothing reduce to message passing (Loeliger et al., 2007). This bridge has been widely exploited in temporal settings, where the natural ordering of time supplies the one-dimensional chain that the state-space form requires. We use this bridge as the starting point for multi-output regression: we cast low-input-dimensional multi-output GP (MOGP) regression as message passing on an FFG by combining state-space GPs along a greedy nearest-neighbor chain over a fixed candidate set of size C with the Linear Model of Coregionalization (LMC), expressed as a deterministic mixing factor and per-output scalar observation factors. The chain construction is an approximation, since M-dimensional input geometry is compressed onto a one-dimensional Markov sequence. Here we investigate the fidelity of such a compression.

Casting the model in this form has two consequences for inference. First, conditional on the chain, posterior computation is exact Gaussian message passing: the L latent dynamics remain block diagonal in the prior, while the LMC observation factors couple them into dense Gaussian beliefs over the joint latent state. For fixed D and L, inference is therefore linear in the candidate-chain length C, with $\mathcal{O}(C(DL^2 + L^3))$ cost after chain construction. Second, partial observability is native to the factor-graph implementation: outputs that are unobserved at a given input simply omit their local observation message, with no covariance-matrix restructuring and no change to the model or to the inference call. Dense kernel-matrix implementations, by contrast, must explicitly build or slice variable-size covariance matrices from the observed (output, input) pairs.

Concretely, our contribution is firstly a chain-induced factor graph formulation of low-input-dimensional multi-output GP regression, combining state-space GPs (Hartikainen and Särkkä, 2010), FFG message passing (Loeliger et al., 2007), and the LMC (Álvarez et al., 2012), with $\mathcal{O}(C(DL^2 + L^3))$ per-step inference after chain construction. Approximation occurs only through the nearest-neighbor chain, inference on the resulting model is exact. Secondly, partial observability requires no restructuring of the inference procedure, as opposed to dense kernel-matrix treatments that require covariance-matrix slicing.

2. Problem Statement

Let $x \in \mathcal{X} \subset \mathbb{R}^M$ denote an input and $f : \mathcal{X} \to \mathbb{R}^D$ a vector-valued function with D correlated outputs. Given a training set of N inputs $X \in \mathbb{R}^{N \times M}$ and a noisy observation matrix $Y \in \mathbb{R}^{N \times D}$ with $y_i = f(x_i) + \epsilon_i$ and Gaussian noise $\epsilon_i \sim \mathcal{N}(0, \Sigma_\epsilon)$, where $\Sigma_\epsilon \in \mathbb{R}^{D \times D}$ is a diagonal observation-noise covariance, we wish to infer the posterior predictive

distribution of f at unseen inputs under a zero-mean multi-output Gaussian process prior. A binary mask $O \in \{0,1\}^{N\times D}$ records which entries are observed, with arbitrary missingness allowed across (i,d).

We target exact Gaussian inference under the chain-induced factor-graph model with two requirements. First, inference cost should grow linearly in the number of training points N for fixed output count D and fixed latent rank. Second, the missingness mask O should be absorbed into the model itself, so that arbitrary missingness patterns produce no additional restructuring cost. Kernel-matrix methods satisfy neither requirement, as vectorized inference costs $\mathcal{O}((ND)^3)$ and varying missingness forces a variable-size covariance to be rebuilt from the observed entries. Section 4 presents a factor-graph reformulation that meets both requirements, and Section 5 assesses posterior quality empirically.

3. Background

3.1. Multi-Output Gaussian Processes

Exact multi-output GP inference is costly because covariance matrices couple inputs and outputs. We focus on the Linear Model of Coregionalization (LMC), which mixes L latent GPs $g_l(x)$ through a mixing matrix $W \in \mathbb{R}^{D\times L}$ with induced cross-output covariance

$$f_d(x) = \sum_{l=1}^{L} w_{dl} g_l(x)\,, \qquad \mathrm{Cov}\big[f_d(x), f_{d'}(x')\big] = \sum_{l=1}^{L} w_{dl} w_{d'l} k_l(x,x')\,. \tag{1}$$

At a fixed input, or with shared latent input covariance, the output-side structure is low-rank plus diagonal noise. For $L \ll D$, Woodbury applies inverses of $WSW^\top + \Sigma_\epsilon$ through an $L\times L$ inner system, costing $\mathcal{O}(DL^2 + L^3)$, or $\mathcal{O}(DL^2)$ for fixed L, instead of dense $D\times D$ operations. The special case $L = 1$ is the Intrinsic Coregionalization Model (ICM), which is less expressive than LMC (Alvarez and Lawrence, 2011; Bonilla et al., 2007).

3.2. State-Space Gaussian Processes

The cubic complexity in samples has driven extensive work on scalable approximations. Sparse variational methods based on inducing points reduce this cost while preserving a variational lower bound on the marginal likelihood (Smola et al., 2004; Bruinsma et al., 2020). However, these are approximations. Temporal GPs with Matérn covariance functions admit an exact representation as linear SDEs, converting GP regression into Kalman filtering and smoothing at $\mathcal{O}(N)$ cost (Hartikainen and Särkkä, 2010; Särkkä, 2013; Särkkä and Solin, 2019). For example, a scalar temporal GP with a stationary Matérn 3/2 covariance function $\kappa(t,t')$, length-scale ℓ, and output scale γ^2 can be represented exactly as the solution of the linear SDE

$$\mathrm{d}\begin{bmatrix} g(t) \\ \dot{g}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\lambda^2 & -2\lambda \end{bmatrix} \begin{bmatrix} g(t) \\ \dot{g}(t) \end{bmatrix} \mathrm{d}t + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathrm{d}w(t)\,, \tag{2}$$

for $\lambda = \sqrt{3}/\ell$ and white noise $w(t)$ with spectral density $4\lambda^3\gamma^2$, where γ^2 is the GP output scale (Hartikainen and Särkkä, 2010). We shall refer to the state vector of the GP $g(t)$ and its derivative in time $\dot{g}(t)$ as $z(t)$. The stationary covariance of the state is $P_\infty =$

$\mathrm{diag}(\begin{bmatrix}\gamma^2 & \lambda^2\gamma^2\end{bmatrix})$. This SDE can be discretized to form a discrete-time state-space model over states $z_t = [g_t,\, \dot{g}_t]^\top$. Given the inter-point distance Δ_i between consecutive points in the chain, the discrete-time transition and process noise matrices are

$$A_i = \exp\left(\begin{bmatrix} 0 & 1 \\ -\lambda^2 & -2\lambda \end{bmatrix} \Delta_i\right), \qquad Q_i = P_\infty - A_i\, P_\infty\, A_i^\top. \tag{3}$$

The observation model extracts the function value from the state through the observation vector $h = [1,\, 0]^\top$, so that $g_t = h^\top z_t$. This yields a linear-Gaussian state-space model equivalent to the GP, with $\mathcal{O}(N)$ inference in the number of points for a single scalar output via Bayesian filtering and smoothing (Särkkä, 2013). The multi-output cost is treated in Section 4.2.

3.3. Factor Graphs and Message Passing

Factor graphs provide a unified language for probabilistic inference, subsuming belief propagation, variational inference and expectation propagation as message computation on a graph (Kschischang et al., 2001; Şenöz et al., 2021; Heess et al., 2013). Many classical algorithms, including Kalman filtering and smoothing, can be written as message passing on particular factor graphs (Loeliger et al., 2007; Palmieri et al., 2022). Take, for example, the below probabilistic model over x_1, x_2 and x_3, and assume that we are interested in the marginal distribution for x_3:

$$p(x_1, x_2, x_3) = p(x_1 \,|\, x_2)p(x_2 \,|\, x_3)p(x_3)\,, \qquad p(x_3) = \int p(x_1, x_2, x_3)\mathrm{d}(x_1, x_2)\,. \tag{4}$$

In a Forney-style factor graph (FFG), factors are nodes and variables are edges. Each edge connects at most two factor nodes (Loeliger, 2004). The notation is illustrated in Figure 1: boxes denote factors, edge labels denote variables, and arrows label messages passed along those variables. For a generic node $f(y, x_1, \ldots, x_n)$, the sum-product message to edge y is

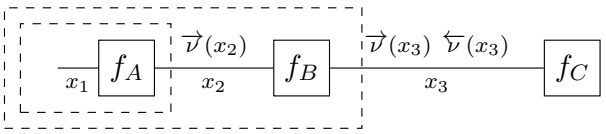


Figure 1: Example FFG. Messages along an edge summarize all factors of other edges.

$$\overrightarrow{\nu}(y) = \int \cdots \int f(y, x_1, \ldots, x_n) \prod_{i=1}^{n} \overrightarrow{\nu}(x_i)\mathrm{d}x_i\,, \tag{5}$$

where $\overrightarrow{\nu}(x_i)$ are incoming messages to the node. Priors can be represented as terminal factors, likelihoods send observation information back into the graph, and posterior marginals are obtained by multiplying incoming messages on an edge (Şenöz et al., 2021). A key property of FFGs is compositionality: model changes, such as adding or removing observation factors, require only local graph modifications, not a new inference algorithm. Reactive message passing exploits this structure by compiling model declarations into message updates and recomputing only affected messages when data or factors change (Bagaev and De Vries, 2023). This compositionality is central to our treatment of partial observations.

4. Message Passing-Based Multi-Output GP Inference

4.1. Model Specification

4.1.1. Nearest-Neighbor Chain Ordering

Extending state-space GPs beyond one-dimensional inputs is an active research area. In spatiotemporal settings, a natural temporal ordering exists, and the spatial structure can be handled separately via inducing points (Särkkä et al., 2013; Tebbutt et al., 2021; Hamelijnck et al., 2021). An alternative approach is the SPDE method of Lindgren et al. (2011), which discretizes the stochastic partial differential equation on a mesh to obtain a sparse precision matrix for Matérn fields in arbitrary dimensions. Recently, Li et al. (2025) proposed a universal method for converting any stationary temporal GP into a state-space model via spectral factorization. Our work differs from these spatiotemporal approaches in that we do not assume separable structure. Instead, we construct a one-dimensional chain via nearest-neighbor ordering. This makes state-space inference applicable to non-temporal candidate sets, but it is an approximation whose fidelity depends on how well the chain preserves the relevant geometry. A closely related family also relies on nearest-neighbor orderings: the Vecchia approximation (Vecchia, 1988) conditions each point on a small set of preceding neighbors to obtain a sparse Cholesky factor of the precision matrix, realized as the hierarchical nearest-neighbor GP of Datta et al. (2016) and generalized by Katzfuss and Guinness (2021). These methods produce an *approximate* sparse precision from variable-size conditioning sets, whereas our Markov-1 chain gives an exact message-passing recursion at the cost of compressing the M-dimensional geometry into a single chain.

For M-dimensional inputs $X \in \mathbb{R}^{C\times M}$, with fixed candidate-set size C, we construct this ordering via a greedy nearest-neighbor heuristic. Let $\pi : \{1,\dots,C\} \to \{1,\dots,C\}$ denote the resulting permutation, so that $x_{\pi(i)}$ is the i-th point visited in the chain: starting from an arbitrary point (e.g., x_1), we repeatedly visit the closest unvisited point in Euclidean distance and record the inter-point distances $\Delta_i = \|x_{\pi(i)} - x_{\pi(i-1)}\|$, which serve as the time steps in the state-space model. This ordering defines a chain-induced kernel approximation. Let $s_{\pi(1)} = 0$ and $s_{\pi(i)} = \sum_{r=2}^{i} \Delta_r$ be the one-dimensional coordinate induced by the chain. The chain distance is the distance between these coordinates and the chain kernel is the kernel defined by the Matérn kernel covariance function evaluated at the chain distance:

$$[K_\pi]_{ij} = \kappa\big(|s_{\pi(i)} - s_{\pi(j)}|\big)\,. \tag{6}$$

A naive greedy approach to constructing the chain will cost $\mathcal{O}(C^2)$, but techniques such as *kd*-trees can speed this up to $\mathcal{O}(C \log C)$ in low input dimension (Cormen et al., 2022).

To characterize the fidelity of the chain-induced kernel approximation, we derive a bound on the distance in Matérn-kernel GP posterior parameters. First, we tackle the case of single-output GP, i.e., for the l-th GP, and then generalize to multi-output, i.e., all L GPs.

Lemma 1 *Let $\kappa_l(\cdot)$ be a Matérn-class kernel with length-scale $\ell_l > 0$ and output scale $\gamma_l^2 > 0$, and let α_l be its Lipschitz constant on the range of distances induced by X and the chain π. Define $[K_l]_{ij} = \kappa_l\big(\|x_i - x_j\|\big)$, $[K_{\pi,l}]_{ij} = \kappa_l\big(|s_{\pi(i)} - s_{\pi(j)}|\big)$ and $\varepsilon_\pi = \max_{i,j} \big|\|x_i - x_j\| - |s_{\pi(i)} - s_{\pi(j)}|\big|$. Then,*

$$\|K_l - K_{\pi,l}\|_2 \le C\,\alpha_l\,\varepsilon_\pi\,. \tag{7}$$

The proof is in Appendix E. Here C is the candidate-set size, so the bound scales with the number of points but vanishes as the chain distortion $\varepsilon_\pi \to 0$, that is, as the one-dimensional ordering better preserves the input geometry. Under scalar observation noise of variance σ_n^2, the GP posterior parameters obtained by conditioning on observations y are $\mu = K(K+\sigma_n^2 I)^{-1}y$ and $\Sigma = K - K(K+\sigma_n^2 I)^{-1}K$ (Rasmussen and Williams, 2006), with μ_π and Σ_π the same expressions using the chain-induced kernel of Eq. (6) in place of K. A resolvent perturbation argument converts a bound on $\|K - K_\pi\|_2$ into bounds on these parameters. We defer that step (Lemma 2 in Appendix F) and state the multi-output LMC form directly. Writing $w_l := W_{:,l} \in \mathbb{R}^D$ for the l-th column of the mixing matrix, the Euclidean and chain-induced LMC covariance matrices are defined as

$$K = \sum_{l=1}^{L} (w_l w_l^\top) \otimes K_l \,, \qquad K_\pi = \sum_{l=1}^{L} (w_l w_l^\top) \otimes K_{\pi,l} \,, \tag{8}$$

where $K_l, K_{\pi,l}$ are as in Lemma 1.

Theorem 1 *Let $\eta = \|K - K_\pi\|_2$. Assume scalar observation noise $\sigma_n^2 I$ and that each κ_l is a Matérn-class kernel with parameters (ℓ_l, γ_l^2) and Lipschitz constant α_l. Then the multi-output GP posterior parameters obtained by conditioning on output matrix Y satisfy*

$$\|\mu - \mu_\pi\|_2 \le \left(\frac{\eta}{\sigma_n^2} + \frac{(\|K\|_2 + \eta)\,\eta}{\sigma_n^4} \right) \|Y\|_2 \,, \tag{9}$$

$$\|\Sigma - \Sigma_\pi\|_2 \le \eta + \frac{\eta\,(\|K\|_2 + \|K_\pi\|_2)}{\sigma_n^2} + \frac{\eta\,\|K\|_2\,\|K_\pi\|_2}{\sigma_n^4} \,, \tag{10}$$

with $\eta \le C\,\varepsilon_\pi \sum_{l=1}^{L} \|w_l\|_2^2\,\alpha_l$.

The proof is in Appendix G. Thus, the difference between the per-latent bound and the total multi-output GP bound is the weighted sum of Lipschitz constants. For inputs that are roughly uniform in $[0,1]^M$, the typical nearest-neighbor distance scales as $C^{-1/M}$, so the greedy chain has unnormalized path length $\sum_i \tilde{\Delta}_i = \Theta(C^{1-1/M})$. The chain coordinate range $\max_{i,j} |s_{\pi(i)} - s_{\pi(j)}|$ inherits this scaling, while the Euclidean diameter is bounded by $\sqrt{M}$. Hence the worst-case distortion satisfies $\varepsilon_\pi = O(C^{1-1/M})$ for $M \ge 2$, while at $M{=}1$ the sorted chain is an isometry of the Euclidean ordering and $\varepsilon_\pi = 0$. Plugging back into Theorem 1, $\eta = \mathcal{O}\big(C^{2-1/M} \sum_l \|w_l\|_2^2 \alpha_l\big)$. This is informative for small M, but becomes uninformative as M grows. We stress that both bounds are worst case. ε_π is a maximum over all pairs, and η enters the posterior bounds through σ_n^{-4}. So at practical C and small observation noise they are loose and certify no numerically small posterior error.

Different starts of the chain produce different permutations π and hence different chain coordinate ranges. A classical result on the metric traveling salesman problem bounds this dependence.

Proposition 1 *Define $T(\pi) := \max_i s_{\pi(i)}$ and let π_1, π_2 be two greedy nearest-neighbor chains over the same candidate set X from different starting points. Then*

$$T(\pi_1) \;\le\; \tfrac{1}{2}\big(\lceil \log_2 C \rceil + 1\big)\, T(\pi_2) \,. \tag{11}$$

The proof is in Appendix H. Since $\varepsilon_\pi \le T(\pi)$ for every chain, the bound of Theorem 1 varies by at most an $\mathcal{O}(\log C)$ factor across starting points. Appendix D measures the variation that actually occurs, and finds it far inside this envelope.

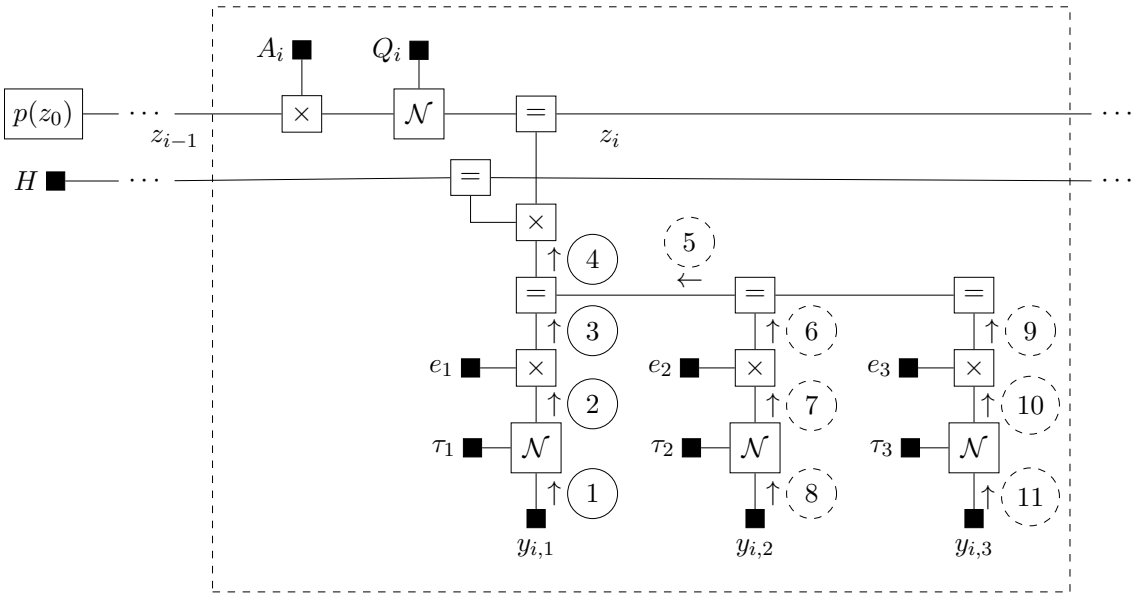


Figure 2: Example Forney-style factor graph of SS-LMC, for $D = 3$. The dashed box indicates a plate repeated over all points in the chain. Messages are carried upwards and joined into a total observation message 4, which updates the chain. If $y_{i,2}$ and $y_{i,3}$ are unobserved, the dashed messages (5–11) are absent and do not contribute.

4.1.2. State-Space GP with LMC

In state-space form, the full state $z_i \in \mathbb{R}^{2L}$ is the concatenation of all latent states. The system matrices are block-diagonal:

$$A_i = \text{blkdiag}\big(A_i^{(1)}, .., A_i^{(L)}\big), \quad Q_i = \text{blkdiag}\big(Q_i^{(1)}, .., Q_i^{(L)}\big), \quad P = \text{blkdiag}\big(P_\infty^{(1)}, .., P_\infty^{(L)}\big). \tag{12}$$

The observation matrix combines the mixing matrix W with the per-latent observation vectors $h^{(l)} = [1,\, 0]^\top$ (each picking out the function value from the l-th latent state):

$$H = \begin{bmatrix} W_{:,1}\, h^{(1)\top} & \cdots & W_{:,L}\, h^{(L)\top} \end{bmatrix} \in \mathbb{R}^{D \times 2L}, \tag{13}$$

so that $H\, z_i$ yields a D-dimensional vector. Each output dimension is observed independently with per-output scalar noise variance τ_d^{-1}. This per-output scalar formulation keeps the model fully linear-Gaussian, enabling exact single-pass inference without variational approximations. The full generative model is expressed as a factor graph for message passing (see Figure 2):

$$p(z_0) = \mathcal{N}(z_0|0, P), \quad p(z_i|z_{i-1}) = \mathcal{N}(z_i \,|\, A_i z_{i-1}, Q_i), \quad p(y_{id} \,|\, z_i) = \mathcal{N}(y_{id} \,|\, e_d^\top H z_i, \tau_d^{-1}), \tag{14}$$

where e_d is a canonical basis vector. Each output at each chain position is an independent scalar observation factor. The observation factor for output d at chain position i is included in the joint only when $O_{i,d} = 1$, and entries with $O_{i,d} = 0$ contribute no factor so that predictions flow through the graph unaffected.

4.2. Inference

Because all noise precisions τ_d are fixed scalars, the model is fully linear-Gaussian. Inference is a single forward-backward pass (exact Kalman smoother), automated by RxInfer.jl. After the candidate chain has been constructed, the per-step inference cost is $\mathcal{O}(C(DL^2+L^3))$ for fixed Matérn state dimension. The block-diagonal structure of A_i and Q_i keeps the latent dynamics sparse, but the LMC observation matrix H couples the L latent processes, so exact inference maintains dense Gaussian beliefs over the joint $2L$-dimensional state. The D per-output scalar observation factors contribute $\mathcal{O}(DL^2)$ work per chain position, and the dense Gaussian state updates contribute the $\mathcal{O}(L^3)$ term. With partial observations, the effective D at each chain position may be smaller, reducing the cost further. We detail this modularity next.

Partial Observations via Modularity In multi-output settings it is common for different outputs to be observed at different inputs. In sensor networks, for instance, sensors at different locations measure different quantities at different times (Liu et al., 2015). The mask O of Section 2 encodes exactly this pattern, with $O_{i,d} = 0$ marking unobserved entries.

The factor graph formulation handles arbitrary masks natively. Each potential observation $y_{i,d}$ corresponds to an independent scalar factor, and the local update at chain position i only fires the factors for which $O_{i,d} = 1$. No covariance-matrix restructuring is needed and no code path changes, so the same model and the same inference call produce posteriors regardless of O. The local message count drops when O has zeros, although fixed overheads may dominate at moderate D. Figure 2 visualizes this for $O_{i,2} = O_{i,3} = 0$, where the dashed messages (5–11) are absent and the subgraph drops from 11 messages to 4.

This contrasts with dense kernel-matrix GP baselines, where partial observations require explicitly constructing or slicing a variable-size $P \times P$ kernel matrix from the P observed output–point pairs. The representational model is the same, but the dense covariance implementation has cost $\mathcal{O}(P^3)$ coupled to the number of observed entries.

5. Experiments

We compare the proposed state-space LMC (SS-LMC) against an exact kernel-matrix LMC (KM-LMC), a sparse-variational inducing-point LMC (SVGP-LMC), and a nearest-neighbor LMC (NNGP-LMC) in two studies: an input-dimension sweep on a synthetic sensor network benchmark that probes the chain approximation (Section 5.1), and a real-data forecasting task on ETTh1 that probes the cost of partial-observation updating (Section 5.2). All methods use identical LMC model structure: the same mixing matrix W, length-scales ℓ_l, output scales γ_l^2, and fixed diagonal noise. The comparison isolates the effect of the inference method and candidate-chain approximation under controlled hyperparameters, reported through held-out posterior-quality metrics and wall-clock time.

RxInfer.jl is used for a reactive message passing implementation (RMP)[1] (Bagaev et al., 2023). All timings compare CPU implementations in the same pipeline on a MacBook Pro with Apple M1 Pro CPU (8 cores) and 32 GB RAM.

1. Experiment code available at https://github.com/biaslab/PGM_2026_SSMOGP

5.1. Accuracy as a Function of Input Dimensionality

We probe the nearest-neighbor chain approximation directly by sweeping the input dimension M on a synthetic sensor network benchmark. Each of the M input coordinates is one weather station at a fixed location, and the D=3 outputs are spatially weighted, saturating combinations of the per-station readings, so increasing M enlarges the input geometry that the one-dimensional chain must compress rather than merely duplicating information. We draw a candidate set of size C=2000, observe a uniformly random 50% as training data and hold out the remaining 50% for evaluation, and fit four methods with identical LMC structure (L=2, $\ell_l \in \{1.0, 2.0\}$, $\gamma_l^2 \in \{2.0, 1.0\}$, fixed diagonal noise): the proposed SS-LMC, an exact kernel-matrix KM-LMC, a sparse-variational SVGP-LMC with 64 inducing points per latent, and an NNGP-LMC conditioning each position on its k=20 nearest locations. For each method we report held-out RMSE and mean negative log-likelihood (MNLL) of the held-out observations, alongside the wall-clock runtime. We sweep $M \in \{2, 4, 8, 16, 32\}$ over 100 seeds and report mean ± standard deviation.

Appendix A reports the nearest-neighbor chain stretch Δ, i.e., the consecutive inter-point distance between chain-ordered candidates, along with the difference in predictive performance between SS-LMC and KM-LMC.

Results. Figure 3 reports held-out accuracy as M grows, now alongside the exact kernel-matrix posterior, the NNGP-LMC and the SVGP-LMC baseline. At low input dimension the chain preserves the local geometry (mean $\Delta \approx 0.10$ at M=2) and SS-LMC closely tracks the exact KM-LMC posterior: held-out RMSE is 0.061 versus 0.058 at M=2 and 0.055 versus 0.046 at M=4, with SVGP-LMC in the same range. As M increases the chain stretches (mean Δ rises to 5.6 and max Δ to 11.2 at M=32) and every method's error grows, since at fixed C the regression problem itself becomes harder. The two approximations degrade gracefully rather than catastrophically: at M=32 SS-LMC reaches RMSE 0.531 and SVGP-LMC 0.495, against 0.417 for exact KM-LMC and 0.524 for NNGP-LMC, and the SS-LMC–KM-LMC RMSE gap stays ≤ 0.12 throughout. SS-LMC's MNLL lies above the exact baseline across the range and stays close to SVGP-LMC's (1.29 against 1.26 at M=32). NNGP-LMC shows the sharpest dependence on input dimension: it is indistinguishable from the exact posterior at M=2 (RMSE 0.0584 for both) and close at M=4, where k=20 neighboring locations capture nearly all the relevant correlation, but by M=32 its MNLL is the worst of the four (1.50). Cost separates the methods far more sharply than accuracy does: one smoothing sweep takes 0.015–0.017 s and is flat in M, as $\mathcal{O}(C(DL^2+L^3))$ predicts, against 1.27–1.36 s for exact KM-LMC, 0.07–0.17 s for SVGP-LMC and 0.41–0.51 s for NNGP-LMC.

5.2. Cost of Partial Observation Updating

We forecast the ETTh1 dataset (Zhou et al., 2021) in a multidimensional-input regime that exercises the chain approximation on real data. We split the seven channels into an M=3 input (the useful-load measurements HUFL, MUFL, LUFL) and D=4 correlated outputs (the useless-load channels HULL, MULL, LULL and oil temperature OT), and order the inputs jointly with the nearest-neighbor chain so that SS-LMC performs $\mathcal{O}(N)$ message passing on three-dimensional inputs. We take a contiguous block of $2N$ rows, train on the first N with a training mask $O_{i,d} \sim \text{Bernoulli}(1-p)$ drawn i.i.d. across (i, d),

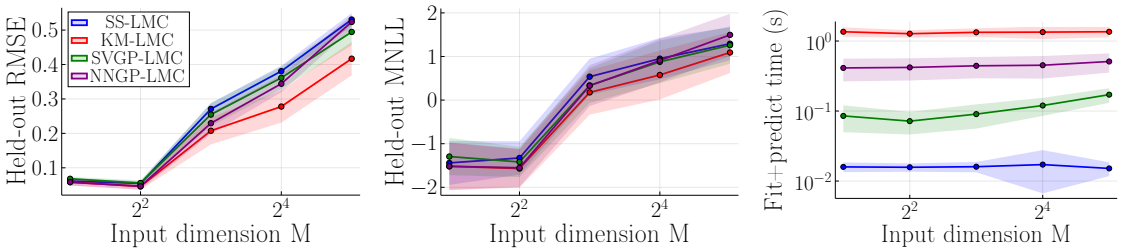


Figure 3: Input-dimension sweep on the synthetic sensor network (C=2000, D=3, L=2, 50/50 train/test, 100 seeds; mean ± std).

and forecast the fully held-out next N rows. We compare four methods with identical LMC structure (L=3, $\ell_l \in \{0.5, 1, 2\}$, γ_l^2=1, fixed diagonal noise R=0.2): the proposed SS-LMC by reactive message passing, an exact KM-LMC by covariance restructuring, a sparse-variational SVGP-LMC with 64 inducing points per latent, and an NNGP-LMC (Datta et al., 2016; Katzfuss and Guinness, 2021) conditioning each position on its k=20 nearest *locations* (kD=80 responses, matched to SVGP-LMC's budget). KM-LMC forms the full structured LMC kernel over the observed (input, output) pairs and refactorizes. Because this dense (DN)-scale factorization is memory-bound, we run it only up to N=2000. SS-LMC simply omits the local observation factor of each missing entry, and SVGP-LMC sums per-datum evidence terms over the observed entries. We report held-out forecast MNLL and RMSE and fit+forecast wall-clock time, sweeping the window length $N \in \{500, 1000, 2000, 4000, 8000\}$ at p=0.3 and the dropout rate $p \in \{0.1, 0.3, 0.5, 0.7\}$ at N=2000, over 10 seeds (mean). The complementary M=1 temporal case, where the natural ordering makes the SS-LMC and KM-LMC posteriors coincide exactly so the comparison is purely computational, is reported in Appendix B.

Results. Figure 4 reports the four methods across window length and dropout. Because the input is now three-dimensional, the chain is an approximation and SS-LMC no longer coincides with KM-LMC, but its forecast accuracy stays close: at N=2000 the held-out RMSE is 8.17 for SS-LMC against 7.98 for exact KM-LMC, 8.10 for SVGP-LMC and 7.98 for NNGP-LMC, and at N=8000 (beyond KM-LMC's reach) SS-LMC reaches RMSE 5.15 against 4.95 for SVGP-LMC and 5.19 for NNGP-LMC, with SVGP-LMC holding a small MNLL advantage throughout and SS-LMC ahead of KM-LMC and NNGP-LMC on MNLL. The cost, however, separates the methods sharply. KM-LMC's covariance restructuring scales cubically (0.62 → 2.53 → 13.2 s for N=500 → 2000) and is infeasible beyond N=2000, whereas SS-LMC's single forward–backward sweep grows linearly and reaches 0.16 s at N=8000, 7.6× below the inducing-point SVGP-LMC (1.25 s) and 38× below NNGP-LMC (6.16 s). The dropout sweep shows the complementary effect: SS-LMC's cost is essentially invariant to the missingness rate (∼0.036 s across p), since missing entries merely omit local factors, while KM-LMC's cost falls with sparser data (18.5 → 5.7 s as p: 0.1 → 0.7). SVGP-LMC stays near 0.3 s and NNGP-LMC from 1.33 to 0.61 s. Appendix C recasts both views as an accuracy–cost frontier at fixed window length. Handling partial observations therefore requires no covariance-matrix restructuring in SS-LMC, and the chain approximation

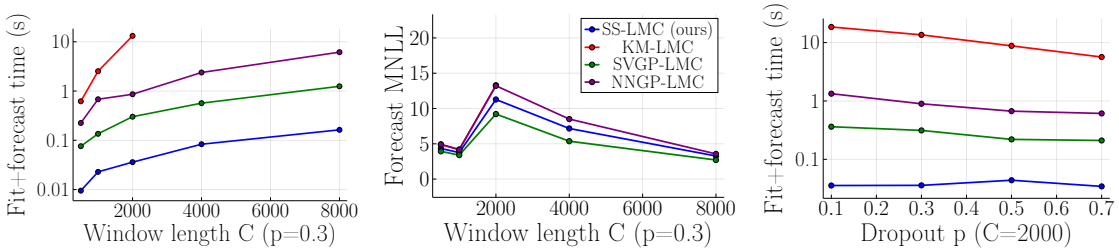


Figure 4: ETTh1 forecasting at M=3, D=4, L=3, 10 seeds. KM-LMC runs only to N=2000. Timings exclude the one-off chain construction (cf. Section 6).

Table 1: Per-step computational cost of the baselines and the proposed SS-LMC. m: SVGP inducing points per latent. k: NNGP conditioning locations, each carrying D outputs. $P = \sum_{i,d} O_{i,d}$: observed output–input pairs.

	Naive MOGP	KM-LMC	SVGP-LMC	NNGP-LMC	SS-LMC (ours)
Per-step inference	$\mathcal{O}((DN)^3)$	$\mathcal{O}(LN^3 + NDL^2)$	$\mathcal{O}(L(Nm^2 + m^3))$	$\mathcal{O}(N(kD)^3)$	$\mathcal{O}(C(DL^2 + L^3))$
Chain construction	—	—	—	$\mathcal{O}(C^2)$ brute force	$\mathcal{O}(C^2)$ brute force
Partial observations	$\mathcal{O}((DN)^3)$	$\mathcal{O}(P^3)$	$\mathcal{O}(L(Pm^2 + m^3))$	$\mathcal{O}(N(kD)^3)$	$\mathcal{O}(C(DL^2 + L^3))$
Total over N steps	$\mathcal{O}(D^3N^4)$	$\mathcal{O}(LN^4{+}N^2DL^2)$	$\mathcal{O}(LNm^2)$	$\mathcal{O}(C^2{+}N^2(kD)^3)$	$\mathcal{O}(C^2{+}NC(DL^2{+}L^3))$

costs little accuracy relative to an exact, a sparse-variational and a nearest-neighbor LMC baseline.

6. Discussion

SS-LMC has $\mathcal{O}(C(DL^2 + L^3))$ per-step inference after chain construction (Table 1). NNGP-LMC (Datta et al., 2016; Katzfuss and Guinness, 2021) shares the nearest-neighbor ingredient but conditions each prediction on k neighbor *locations* instead of propagating one state along a chain, and pays the same $\mathcal{O}(C^2)$ brute-force construction (Finley et al., 2019, §S1). Chain construction is preprocessing that the kernel-matrix and inducing-point baselines avoid, but it depends only on the candidate inputs, and not on Y, the mask O, or the hyperparameters. So it is built once and reused by every later inference call. Writing c_π for its cost, R reuses favor SS-LMC once $R > c_\pi/(t_{\text{base}} - t_{\text{SS}})$. Since the one-off chain cost is well below the per-fit gap to either baseline, the construction is recovered within a single fit, and repeated queries, hyperparameter-learning iterations, and streaming updates amortize it further. The nearest-neighbor chain replaces Euclidean by path distances along a one-dimensional Markov chain, and the input-dimension sweep (Figure 3) shows the resulting trade-off: fidelity is high at low M and degrades gradually as the chain stretches. The method is therefore strongest for low-input-dimensional candidate sets.

Nguyen et al. (2025) also decomposes a Gaussian process into a factor graph formulation, but utilizes variational approximations instead of the state-space formulation. It is a factor graph treatment of the sparse variational Gaussian process, and has the same computational and memory complexity at similar accuracy in regression and classification tasks.

Hyperparameter Learning. Learning the hyperparameters (W, ℓ_l, γ_l^2), which are fixed throughout this work, interacts asymmetrically with the graph. The mixing matrix enters only through H and so touches the C observation blocks alone, whereas ℓ_l and γ_l^2 enter through A_i and Q_i and re-parameterize every transition factor: unlike the missingness mask, a hyperparameter update is not a local graph edit. The ordering is spared, since π and the Δ_i depend only on the input geometry and are computed once and reused across iterations. Expectation maximization is then natural, alternating one $\mathcal{O}(C(DL^2 + L^3))$ smoothing sweep with automatic-differentiation gradient steps on the hyperparameters, taking the free energy returned by that same sweep as the objective, with banded linear algebra for the gradients (Durrande et al., 2019); particle MCMC (Svensson et al., 2016) and recursive Bayesian autoregression (Kouw, 2025) are alternatives. Learned length-scales are moreover fit to chain rather than Euclidean distances, so they absorb part of the chain stretch and inflate α_l, and with it the bound of Theorem 1.

Limitations. Firstly, the greedy nearest-neighbor chain is path-dependent. However, the effect is, so far, empirically negligible: held-out RMSE varies by $<0.1\%$ across 20 starting points (within the $\mathcal{O}(\log C)$ bound of Proposition 1), and even adversarially diverse starts leave it unchanged (Appendix D). Secondly, the ETT study (Section 5.2) exercises an $M{=}3$ input on real data, but a systematic evaluation at higher input dimension on real data is left to future work. Thirdly, the bounds of Lemma 1 and Theorem 1 are illustrative from a theoretical perspective but do not furnish error estimates one would use in practice. Lastly, the method is currently limited to Matérn-class covariances that admit state-space representations.

7. Conclusion

We presented low-input-dimensional multi-output GP inference as message passing on a Forney-style factor graph, combining state-space GPs, LMC, and reactive message passing. On a synthetic sensor-network benchmark it tracks the exact kernel-matrix posterior at low input dimension and degrades gracefully as the chain stretches with input dimension. On real ETTh1 forecasting under random per-(row, output) dropout with a three-dimensional input, SS-LMC matches both baselines in forecast accuracy to within a small margin, while its inference cost grows only linearly in the window length and is invariant to the dropout rate. The factor graph handles partial observations without covariance-matrix restructuring.

Acknowledgments

This publication is part of the ROBUST project with project number KICH3.LTP.20.006, which is (partly) financed by the Dutch Research Council (NWO), GN Hearing, and the Dutch Ministry of Economic Affairs and Climate Policy (EZK) under the program LTP KIC 2020-2023. This project is also partly financed by Holland High Tech with PPS funding for the AUTO-AR project RVO TKI2112P09.

References

Mauricio A Alvarez and Neil D Lawrence. Computationally efficient convolved multiple output Gaussian processes. *Journal of Machine Learning Research*, 12:1459–1500, 2011.

Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012.

Dmitry Bagaev and Bert De Vries. Reactive message passing for scalable Bayesian inference. *Scientific Programming*, 2023:1–26, 2023.

Dmitry Bagaev, Albert Podusenko, and Bert de Vries. RxInfer: A Julia package for reactive real-time Bayesian inference. *Journal of Open Source Software*, 8(84):5161, 2023.

Edwin V Bonilla, Kian Ming A. Chai, and Christopher K. I. Williams. Multi-task Gaussian process prediction. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.

Wessel Bruinsma, Eric Perim, William Tebbutt, Scott Hosking, Arno Solin, and Richard Turner. Scalable exact inference in multi-output Gaussian processes. In *International Conference on Machine Learning*, pages 1190–1201. PMLR, 2020.

Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT press, 2022.

Abhirup Datta, Sudipto Banerjee, Andrew O. Finley, and Alan E. Gelfand. Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets. *Journal of the American Statistical Association*, 111(514):800–812, 2016.

Nicolas Durrande, Vincent Adam, Lucas Bordeaux, Stefanos Eleftheriadis, and James Hensman. Banded matrix operators for Gaussian Markov models in the automatic differentiation era. In *International Conference on Artificial Intelligence and Statistics*, pages 2780–2789. PMLR, 2019.

Andrew O. Finley, Abhirup Datta, Bruce D. Cook, Douglas C. Morton, Hans E. Andersen, and Sudipto Banerjee. Efficient algorithms for Bayesian nearest neighbor Gaussian processes. *Journal of Computational and Graphical Statistics*, 28(2):401–414, 2019.

Oliver Hamelijnck, William Wilkinson, Niki Loppi, Arno Solin, and Theodoros Damoulas. Spatio-temporal variational Gaussian processes. In *Advances in Neural Information Processing Systems*, volume 34, pages 23621–23633. Curran Associates, Inc., 2021.

Jouni Hartikainen and Simo Särkkä. Kalman filtering and smoothing solutions to temporal Gaussian process regression models. In *IEEE International Workshop on Machine Learning for Signal Processing*, pages 379–384, 2010.

Nicolas Heess, Daniel Tarlow, and John Winn. Learning to pass expectation propagation messages. In *Advances in Neural Information Processing Systems*, volume 26, pages 3219–3227. Curran Associates, Inc., 2013.

Roger A Horn and Charles R Johnson. *Matrix Analysis*. Cambridge University Press, 2012.

Matthias Katzfuss and Joseph Guinness. A general framework for Vecchia approximations of Gaussian processes. *Statistical Science*, 36(1):124–141, 2021.

Wouter M. Kouw. Bayesian autoregression to optimize temporal Matérn kernel Gaussian process hyperparameters. In *International Conference on Probabilistic Numerics*, pages 122–130. PMLR, 2025.

Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

Weihan Li, Yule Wang, Chengrui Li, and Anqi Wu. Learning time-varying multi-region brain communications via scalable Markovian Gaussian processes. In *International Conference on Machine Learning*, volume 267, pages 36021–36041. PMLR, 2025.

Finn Lindgren, Håvard Rue, and Johan Lindström. An explicit link between Gaussian fields and Gaussian Markov random fields: The stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73 (4):423–498, 2011.

Qinyuan Liu, Zidong Wang, Xiao He, and D. H. Zhou. Event-based recursive distributed filtering over wireless sensor networks. *IEEE Transactions on Automatic Control*, 60(9): 2470–2475, 2015.

Hans-Andrea Loeliger. An introduction to factor graphs. *IEEE Signal Processing Magazine*, 21(1):28–41, 2004.

Hans-Andrea Loeliger, Justin Dauwels, Junli Hu, Sascha Korl, Li Ping, and Frank R. Kschischang. The factor graph approach to model-based signal processing. *Proceedings of the IEEE*, 95(6):1295–1322, 2007.

Hoang M.H. Nguyen, İsmail Şenöz, and Bert De Vries. A factor graph approach to variational sparse Gaussian processes. *IEEE Open Journal of Signal Processing*, 6:815–837, 2025.

Francesco A. N. Palmieri, Krishna R. Pattipati, Giovanni Di Gennaro, Giovanni Fioretti, Francesco Verolla, and Amedeo Buonanno. A unifying view of estimation and control using belief propagation with application to path planning. *IEEE Access*, 10:15193–15216, 2022.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Daniel J. Rosenkrantz, Richard E. Stearns, and Philip M. Lewis, II. An analysis of several heuristics for the traveling salesman problem. *SIAM Journal on Computing*, 6(3):563–581, 1977.

Simo Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

Simo Särkkä and Arno Solin. *Applied Stochastic Differential Equations*. Cambridge University Press, 2019.

Simo Särkkä, Arno Solin, and Jouni Hartikainen. Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing: A look at Gaussian process regression through Kalman filtering. *IEEE Signal Processing Magazine*, 30(4):51–61, 2013.

İsmail Şenöz, Thijs van de Laar, Dmitry Bagaev, and Bert de Vries. Variational message passing and local constraint manipulation in factor graphs. *Entropy*, 23(7):807, 2021.

Alex J. Smola, S. V. N. Vishwanathan, and Eleazar Eskin. Laplace propagation. In *Advances in Neural Information Processing Systems*, pages 441–448, 2004.

Andreas Svensson, Arno Solin, Simo Särkkä, and Thomas Schön. Computationally efficient Bayesian learning of Gaussian process state space models. In *International Conference on Artificial Intelligence and Statistics*, pages 213–221. PMLR, 2016.

Will Tebbutt, Arno Solin, and Richard E. Turner. Combining pseudo-point and state space approximations for sum-separable Gaussian processes. In *Conference on Uncertainty in Artificial Intelligence*, pages 1607–1617. PMLR, 2021.

A. V. Vecchia. Estimation and model identification for continuous spatial processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2):297–312, 1988.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI Conference on Artificial Intelligence*, volume 35, pages 11106–11115, 2021.

Appendix A. Chain Stretch and the Gap to the Exact Posterior

Figure 5 reports the geometry the chain has to compress and what it costs against the exact posterior. The stretch Δ grows steadily with M (mean $0.10 \to 5.6$, max $4.5 \to 11.2$ for $M{:}\,2 \to 32$), and the accuracy gap follows it. The difference between SS-LMC and KM-LMC within each seed rises monotonically in RMSE, from 0.003 ± 0.002 at $M{=}2$ to 0.114 ± 0.035 at $M{=}32$. The MNLL gap rises likewise, from 0.08 ± 0.08 to a maximum of 0.37 ± 0.29 at $M{=}16$, then falls back to 0.20 ± 0.23 at $M{=}32$, where the regression problem is hard enough at fixed C that the exact posterior loses part of its advantage too.

Appendix B. ETT Temporal ($M{=}1$) Case

As a complement to the multidimensional-input study of Section 5.2, we also forecast ETTh1 in the purely temporal setting: all seven channels are modeled jointly as $D{=}7$ correlated outputs over normalized time, so $M{=}1$ and the candidate chain reduces to the natural temporal ordering. Here SS-LMC and KM-LMC realize the same model, so their posteriors coincide to numerical precision at every window length and the comparison is purely computational. Figure 6 shows that SS-LMC's cost grows linearly in the window length while KM-LMC's covariance restructuring grows cubically, with SS-LMC reaching a $4.4\times$ speed-up at $C{=}4000$. SS-LMC's cost is also invariant to the dropout rate, whereas KM-LMC's falls with sparser data but stays tied to dense factorization. These timings come from a separate run and are comparable within the figure, not against Figure 4.

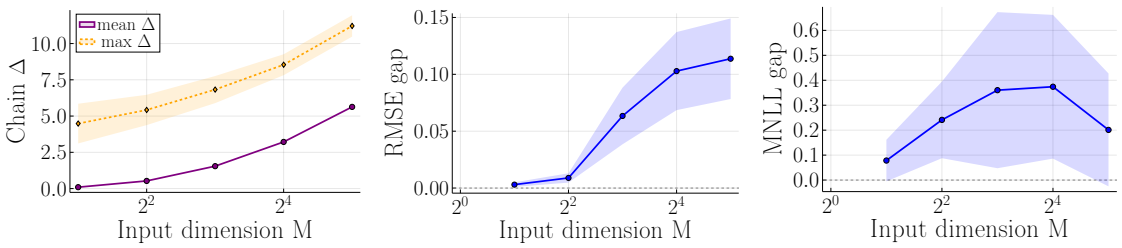


Figure 5: Chain stretch and approximation gap on the sensor network sweep (C=2000, D=3, L=2, 100 seeds). Left: nearest-neighbor chain stretch Δ (mean, max) vs. input dim M. Middle and right: held-out RMSE and MNLL gap vs. input dim M, as SS-LMC minus exact KM-LMC. Difference is taken within each seed and then averaged (mean $\pm$ std).

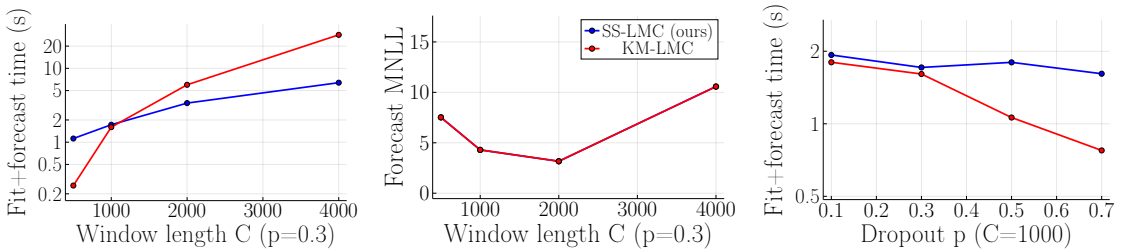


Figure 6: ETTh1 temporal forecasting (M=1 natural ordering, D=7, L=4, 3 seeds): train on the first half of a length-C window, forecast the held-out second half under per-(timestamp, output) dropout. At M=1 the SS-LMC and KM-LMC posteriors are identical, so the comparison is computational. Left: fit+forecast wall-clock vs. C at dropout p=0.3 (log scale). Middle: held-out MNLL vs. C. Right: time vs. p at C=1000 (log scale).

Appendix C. Accuracy–Cost Frontier

Held-out score against fit+forecast wall-clock at fixed window length separates the two accuracy measures of Section 5.2. Both windows shown are ones where exact KM-LMC still fits in memory, so every panel carries a complete four-method frontier. On RMSE (Figure 7, top row) the mean ordering shifts with the window: at C=1000 SVGP-LMC is on average both cheaper and more accurate than KM-LMC and NNGP-LMC, whereas at C=2000 the four means form a monotone frontier. The accuracy differences are small against the seed spread, however: at C=2000 the entire 0.18 RMSE span from SS-LMC to KM-LMC is half the $\pm$0.34 standard deviation across seeds, bought at a 367$\times$ cost ratio, and NNGP-LMC reaches KM-LMC's mean RMSE at 15$\times$ less time. On MNLL (bottom row) SVGP-LMC and SS-LMC hold the frontier at both windows, with KM-LMC and NNGP-LMC both slower and less well calibrated on average.

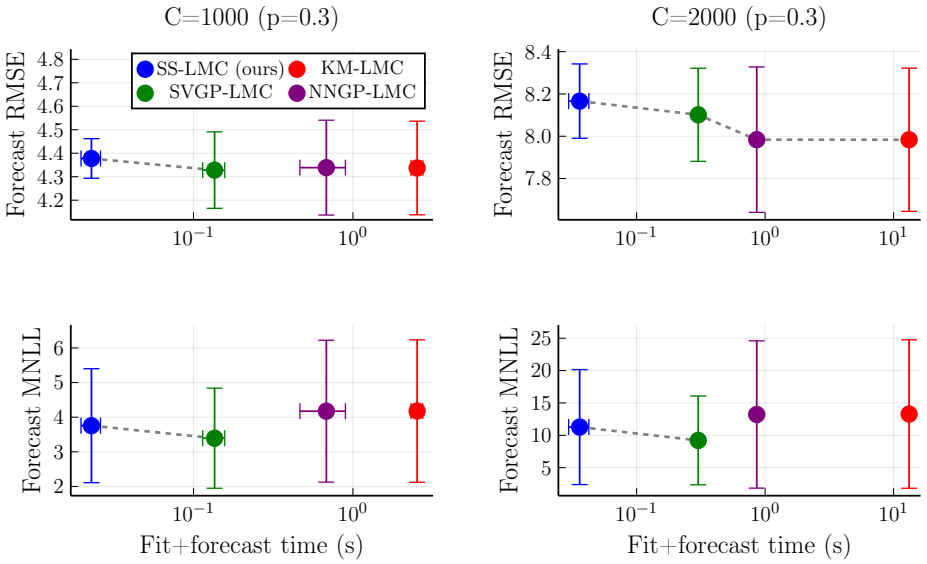


Figure 7: Held-out score against fit+forecast wall-clock on ETTh1 (M=3, D=4, L=3, dropout p=0.3, 10 seeds). Lower-left is better. Top row: RMSE; bottom row: MNLL. Columns are C=1000 and C=2000. Dashed line is Pareto front. Bars are ± 1 standard deviation over seeds.

Appendix D. Sensitivity to the Chain Starting Point

Proposition 1 bounds the worst-case variation across starting points. In this experiment, we measure the variation over 20 starts. Within a seed, the dropout mask and train/test split are fixed on physical points before ordering. W is fixed, so every start sees the same observations. Per start we take the median over 10 seeds, then the standard deviation across starts. Since that is itself an estimated dispersion, the error bars are a bootstrap 95% confidence interval for it (relative standard error $\approx$16% at 20 starts), so only well-separated points differ.

Sensitivity is governed by input dimension, not candidate-set size. Against M (Figure 8, left) the spread is flat at 2.5–4.0 $\times 10^{-4}$ for $M \leq 8$ and steps up roughly threefold at $M \geq 16$ mirroring the $\varepsilon_\pi = \mathcal{O}(C^{1-1/M})$ compression of M-dimensional geometry into one chain coordinate. Against C (right) the intervals overlap, so the spread is flat in C. The relative effect stays negligible: on ETTh1 the coefficient of variation of held-out RMSE across starts is $\leq 0.09\%$ at every C. MNLL, not plotted, is somewhat more sensitive, growing four- to sevenfold over the same range of M. Three checks support reading this as real but harmless path dependence: the permutation-invariant KM-LMC control varies 10–50$\times$ less. The even, random and farthest-point start schemes give the same chain diversity and spread, so a bad chain cannot be engineered from a bad start. ε_π does not predict per-start error, its bootstrap correlation intervals remain close to zero. The observed variation therefore sits far inside the envelope of Proposition 1, which holds but is conservative.

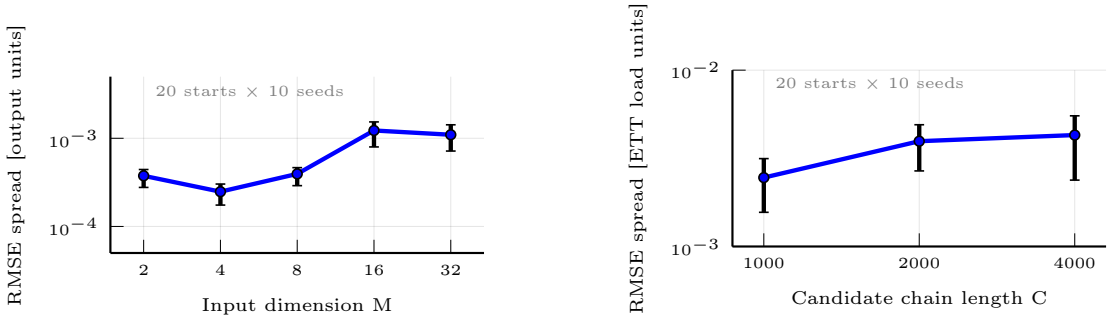


Figure 8: Spread of SS-LMC's held-out RMSE across 20 chain starting points (median over 10 seeds per start, then standard deviation across starts). Bars are a bootstrap 95% confidence interval for that standard deviation. Left: vs. input dimension M on the synthetic sensor network (C=2000, D=3, L=2). Right: vs. candidate-set size C on ETTh1 (M=3, D=4, L=3, p=0.3).

Appendix E. Proof of Lemma 1

Proof Matérn-class kernels are differentiable away from the origin with bounded derivative on any compact range of distances, hence Lipschitz with a finite constant α_l on the range induced by X and the chain π. For Matérn-3/2 in particular, the maximum of $|\kappa_l'(d)|$ is attained at $d = \ell_l/\sqrt{3}$, giving $\alpha_l = \sqrt{3}\,\gamma_l^2/(e\,\ell_l)$. Other smoothness orders give analogous closed forms. Lipschitz continuity then upper bounds the difference in kernel entries

$$\left|\,[K_l - K_{\pi,l}]_{ij}\,\right| = \left|\kappa_l(\|x_i - x_j\|) - \kappa_l(|s_{\pi(i)} - s_{\pi(j)}|)\right| \le \alpha_l \left|\|x_i - x_j\| - |s_{\pi(i)} - s_{\pi(j)}|\right| . \quad (15)$$

This implies $\|K_l - K_{\pi,l}\|_{\max} \le \alpha_l \max_{ij} \left|\|x_i - x_j\| - |s_{\pi(i)} - s_{\pi(j)}|\right| = \alpha_l\,\varepsilon_\pi$. For any $C \times C$ matrix A, $\|A\|_2 \le \sqrt{\|A\|_1\|A\|_\infty} \le C\|A\|_{\max}$ (Horn and Johnson, 2012). Hence $\|K_l - K_{\pi,l}\|_2 \le C\,\alpha_l\,\varepsilon_\pi$. ∎

Appendix F. Lemma 2 and Its Proof

Lemma 2 *Let $K, K_\pi \succeq 0$ and $\eta = \|K - K_\pi\|_2$. Under scalar observation noise $\sigma_n^2 I$, the GP posterior parameters obtained from K and from K_π satisfy*

$$\|\mu - \mu_\pi\|_2 \le \left(\frac{\eta}{\sigma_n^2} + \frac{(\|K\|_2 + \eta)\eta}{\sigma_n^4}\right)\|y\|_2 \quad (16)$$

$$\|\Sigma - \Sigma_\pi\|_2 \le \eta + \frac{\eta(\|K\|_2 + \|K_\pi\|_2)}{\sigma_n^2} + \frac{\eta\|K\|_2\|K_\pi\|_2}{\sigma_n^4}. \quad (17)$$

Proof Let $B = K + \sigma_n^2 I$, $B_\pi = K_\pi + \sigma_n^2 I$. Since $K, K_\pi \succeq 0$, we have $B, B_\pi \succeq \sigma_n^2 I$, so $\|B^{-1}\|_2, \|B_\pi^{-1}\|_2 \le \sigma_n^{-2}$. The resolvent identity

$$B^{-1} - B_\pi^{-1} = B^{-1}(B_\pi - B)B_\pi^{-1} = B^{-1}(K_\pi - K)B_\pi^{-1} \quad (18)$$

gives $\|B^{-1} - B_\pi^{-1}\|_2 \le \eta\sigma_n^{-4}$. Together with submultiplicativity gives

$$\|B^{-1} - B_\pi^{-1}\|_2 \le \|B^{-1}\|_2 \, \|K - K_\pi\|_2 \, \|B_\pi^{-1}\|_2 \le \eta \, \sigma_n^{-4}. \tag{19}$$

Adding and subtracting $K_\pi B^{-1} y$ inside $\mu - \mu_\pi = KB^{-1}y - K_\pi B_\pi^{-1} y$ splits the difference into a prior-difference term and an inverse-difference term:

$$\mu - \mu_\pi = (K - K_\pi)\, B^{-1}\, y \;+\; K_\pi\, (B^{-1} - B_\pi^{-1})\, y. \tag{20}$$

Taking norms, applying submultiplicativity together with $\|B^{-1}\|_2 \le \sigma_n^{-2}$ and the resolvent bound above, and using $\|K_\pi\|_2 \le \|K\|_2 + \eta$,

$$\|\mu - \mu_\pi\|_2 \le \frac{\eta}{\sigma_n^2}\, \|y\|_2 + \frac{(\|K\|_2 + \eta)\, \eta}{\sigma_n^4}\, \|y\|_2. \tag{21}$$

Note that $\Sigma - \Sigma_\pi = (K - K_\pi) - \big(KB^{-1}K - K_\pi B_\pi^{-1} K_\pi\big)$. If the second piece is expanded by inserting two telescoping terms, $\pm K_\pi B^{-1} K$ and $\pm K_\pi B_\pi^{-1} K$, then we have:

$$KB^{-1}K - K_\pi B_\pi^{-1} K_\pi = (K - K_\pi) B^{-1}\, K + K_\pi\, (B^{-1} - B_\pi^{-1})\, K + K_\pi\, B_\pi^{-1}\, (K - K_\pi). \tag{22}$$

Taking norms termwise and using the same norm bounds,

$$\|\Sigma - \Sigma_\pi\|_2 \le \eta + \frac{\eta\, (\|K\|_2 + \|K_\pi\|_2)}{\sigma_n^2} + \frac{\eta\, \|K\|_2\, \|K_\pi\|_2}{\sigma_n^4}. \tag{23}$$

■

Appendix G. Proof of Theorem 1

Proof Applying Lemma 1 to each Matérn-class latent kernel gives $\|K_l - K_{\pi,l}\|_2 \le C\, \alpha_l\, \varepsilon_\pi$. By subadditivity of the spectral norm, $\|A \otimes B\|_2 = \|A\|_2\, \|B\|_2$ and $\|w_l w_l^\top\|_2 = \|w_l\|_2^2$ (Horn and Johnson, 2012). Thus,

$$\eta \;=\; \|K - K_\pi\|_2 \;\le\; \sum_{l=1}^{L} \|w_l w_l^\top\|_2\, \|K_l - K_{\pi,l}\|_2 \;\le\; C\, \varepsilon_\pi \sum_{l=1}^{L} \|w_l\|_2^2\, \alpha_l\,. \tag{24}$$

Since $K, K_\pi \succeq 0$, applying Lemma 2 with this η yields the stated bounds on $\|\mu - \mu_\pi\|_2$ and $\|\Sigma - \Sigma_\pi\|_2$. ■

Appendix H. Proof of Proposition 1

Proof Let $T(\pi) := \max_i s_{\pi(i)} = \sum_{i=2}^{C} \Delta_i$ denote the chain coordinate range from starting point $\pi(1)$. Since $s_{\pi(1)} = 0$ and the chain visits each point of X exactly once, $T(\pi)$ equals the total length of the greedy nearest-neighbor Hamiltonian path on X initiated at $\pi(1)$. Let $T^\star$ denote the length of the shortest Hamiltonian path on X under the Euclidean metric.

Rosenkrantz et al. (1977) prove that, for any metric instance on C points, the greedy nearest-neighbor heuristic produces a Hamiltonian path of length at most

$$T(\pi) \;\le\; \frac{1}{2}\big(\lceil \log_2 C \rceil + 1\big)\, T^{\star} \tag{25}$$

regardless of the starting point. Applying this upper bound to π_1 and the trivial lower bound $T^{\star} \le T(\pi_2)$ to π_2 gives the claimed inequality

$$T(\pi_1) \;\le\; \frac{1}{2}\big(\lceil \log_2 C \rceil + 1\big)\, T(\pi_2)\,. \tag{26}$$

For any i, j, $\|x_i - x_j\| \le \operatorname{diam}(X) \le T(\pi)$, where the last inequality holds because any Hamiltonian path on X is at least as long as the Euclidean diameter. Furthermore, $|s_{\pi(i)} - s_{\pi(j)}| \le T(\pi)$ and thus, $\varepsilon_{\pi} \le T(\pi)$. Hence the upper bound of Theorem 1 inherits the same $\mathcal{O}(\log C)$ variation across choices of starting point. ■