Supplementary Material

A PRELIMINARIES

In this section, we list some linear algebra properties related to Kronecker products, which will be used in proofs.

We denote the Kronecker product $\otimes$. Let A be of dimension $m \times r$ and B be of dimension $r \times n$; then Harville (1997),

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1r}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mr}B \end{bmatrix}. \tag{A.1}$$

For matrices A, B and X, it holds that

$$\text{vec}(AXB) = (B^\top \otimes A)\text{vec}(X). \tag{A.2}$$

We can particularize this formula for an $r \times 1$ vector x as

$$Ax = \text{vec}(Ax) = (x^\top \otimes I_d)\text{vec}(A). \tag{A.3}$$

Kronecker product has the following mixed product property Harville (1997)

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD), \tag{A.4}$$

and the inversion property Harville (1997)

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}. \tag{A.5}$$

B PROOF OF PROPOSITION 1

We adapt the proof in Akyildiz and Míguez (2019). We first note that for a Gaussian prior $\tilde{p}(c|y_{1:k-1}) = \mathcal{N}(c; c_{k-1}, L_{k-1})$ and likelihood of the form $p(y_k|y_{1:k-1}, c) = \mathcal{N}(y_k; H_k c, G_k)$, we can write the posterior analytically $\tilde{p}(c|y_{1:k}) = \mathcal{N}(c; c_k, L_k)$ where (see, e.g., Bishop (2006))

$$c_k = c_{k-1} + L_{k-1}H_k^\top (H_k L_{k-1} H_k^\top + G_k)^{-1}(y_k - H_k c_{k-1}), \tag{B.1}$$
$$L_k = L_{k-1} - L_{k-1}H_k^\top (H_k L_{k-1} H_k^\top + G_k)^{-1} H_k L_{k-1}. \tag{B.2}$$

In order to obtain an efficient matrix-variate update rule using this vector-form update, we first rewrite the likelihood as

$$\tilde{p}(y_k|c, y_{1:k-1}) = \mathcal{N}(y_k; H_k c, G_k) \tag{B.3}$$

where $H_k = \bar{\mu}_k^\top \otimes I_d$ and $G_k = \eta_k \otimes I_d$. We note that, we have $L_0 = V_0 \otimes I_d$ and we assume as an induction hypothesis that $L_{k-1} = V_{k-1} \otimes I_d$. We start by showing that the update (B.2) can be greatly simplified using the special structure we impose. By the mixed product property (A.4) and the inversion property (A.5) we obtain

$$\left[H_k L_{k-1} H_k^\top + G_k\right]^{-1} = \left[(\bar{\mu}_k^\top \otimes I_d)(V_{k-1} \otimes I_d)(\bar{\mu}_k \otimes I_d) + \eta_k \otimes I_d\right]^{-1} = (\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k)^{-1} \otimes I_d \tag{B.4}$$

and therefore,

$$L_k = (V_{k-1} \otimes I_d) - (V_{k-1}\bar{\mu}_k \otimes I_d) \times ((\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k)^{-1} \otimes I_d) \times (\bar{\mu}_k^\top V_{k-1} \otimes I_d). \tag{B.5}$$

One more use of the mixed product property (A.4) yields

$$L_k = \left(V_{k-1} - \frac{V_{k-1}\bar{\mu}_k \bar{\mu}_k^\top V_{k-1}}{\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k} \right) \otimes I_d. \tag{B.6}$$

Thus, we have $L_k = V_k \otimes I_d$ where,

$$V_k = V_{k-1} - \frac{V_{k-1}\bar{\mu}_k \bar{\mu}_k^\top V_{k-1}}{\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k}. \tag{B.7}$$

We have shown that the sequence $(L_k)_{k\geq 1}$ preserves the Kronecker structure. Next, we substitute $L_{k-1} = V_{k-1}\otimes I_d$, $H_k = \bar{\mu}_k^\top \otimes I_d$ and $G_k = \eta_k \otimes I_d$ into (B.1) and we obtain

$$c_k = c_{k-1} + (V_{k-1}\otimes I_d)(\bar{\mu}_k \otimes I_d) \times \left((\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k)^{-1} \otimes I_d \right) \times (y_k - (\bar{\mu}_k^\top \otimes I_d) c_{k-1}). \tag{B.8}$$

The use of the mixed product property (A.4) leaves us with

$$c_k = c_{k-1} + (V_{k-1}\bar{\mu}_k \otimes I_d) \left((\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k) \otimes I_d \right)^{-1} \times (y_k - (\bar{\mu}_k^\top \otimes I_d) c_{k-1}). \tag{B.9}$$

Using (A.5) and again (A.4) yields

$$c_k = c_{k-1} + \left[\frac{V_{k-1}\bar{\mu}_k}{\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k} \otimes I_d \right] \times (y_k - (\bar{\mu}_k^\top \otimes I_d) c_{k-1}). \tag{B.10}$$

Using (A.3), we get

$$c_k = c_{k-1} + \left[\frac{V_{k-1}\bar{\mu}_k}{\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k} \otimes I_d \right] (y_k - C_{k-1}\bar{\mu}_k). \tag{B.11}$$

We now note that $(y_k - C_{k-1}\bar{\mu}_k)$ and $\frac{V_{k-1}\bar{\mu}_k}{\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k}$ are vectors. Hence, rewriting the above expression as

$$c_k = c_{k-1} + \left[\operatorname{vec}\left(\frac{V_{k-1}\bar{\mu}_k}{\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k} \right) \otimes I_d \right] \times \operatorname{vec}(y_k - C_{k-1}\bar{\mu}_k), \tag{B.12}$$

we can apply (A.3) and obtain

$$c_k = c_{k-1} + \operatorname{vec}\left(\frac{(y_k - C_{k-1}\bar{\mu}_k)\bar{\mu}_k^\top V_{k-1}^\top}{\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k} \right). \tag{B.13}$$

Hence up to a reshaping operation, we have the update rule (20) and conclude the proof. ■

C PROOF OF PROPOSITION 2

Recall that we have a posterior of the form at time $k-1$

$$p(c|y_{1:k-1}) = \mathcal{N}(c; c_{k-1}, V_{k-1} \otimes I_d), \tag{C.1}$$

and we are given the likelihood

$$p(y_k|c, x_k) = \mathcal{N}(y_k; (x_k \otimes I_d)c, R_k). \tag{C.2}$$

We are interested in computing

$$p(y_k|y_{1:k-1}, x_k) = \int p(c|y_{1:k-1}) p(y_k|c, x_k)\, \mathrm{d}c. \tag{C.3}$$

This integral is analytically tractable since both distributions are Gaussian and it is given by Bishop (2006)

$$p(y_k|y_{1:k-1}, x_k) = \mathcal{N}(y_k; (x_k^\top \otimes I_d)c_k, R_k + (x_k^\top \otimes I_d)(V_{k-1}\otimes I_d)(x_k \otimes I_d)). \tag{C.4}$$

Using the mixed product property (A.4), one obtains

$$p(y_k|y_{1:k-1}, x_k) = \mathcal{N}(y_k; C_{k-1}x_k, R_k + x_k^\top V_{k-1} x_k \otimes I_d). \tag{C.5}$$

D DERIVATION OF THE NEGATIVE LOG-LIKELIHOOD

We obtain the marginal likelihood as

$$
\begin{aligned}
\tilde{p}_\theta(y_k|y_{1:k-1}) &= \int \tilde{p}(y_k|y_{1:k-1}, c)\tilde{p}(c|y_{1:k-1})\,\mathrm{d}c && \text{(D.1)}\\
&= \mathcal{N}(y_k; C\bar{\mu}_k, \eta_k \otimes I_d)\mathcal{N}(c; c_{k-1}, V_{k-1} \otimes I_d) && \text{(D.2)}\\
&= \mathcal{N}(y_k; (\bar{\mu}_k^\top \otimes I_d)c, \eta_k \otimes I_d)\mathcal{N}(c; c_{k-1}, V_{k-1} \otimes I_d) && \text{(D.3)}\\
&= \mathcal{N}(y_k; (\bar{\mu}_k^\top \otimes I_d)c_{k-1}, (\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k) \otimes I_d) && \text{(D.4)}\\
&= \mathcal{N}\left(y_k; C_{k-1}f_\theta(\mu_{k-1}), \left(\|f_\theta(\mu_{k-1})\|^2_{V_{k-1}} + \eta_k\right) \otimes I_d\right). && \text{(D.5)}
\end{aligned}
$$

where in the last line we have used the fact that $\bar{\mu}_k = f_\theta(\mu_{k-1})$ and properties from Supp. A. It is then straightforward to show that

$$
\begin{aligned}
-\log \tilde{p}_\theta(y_k \mid y_{1:k-1}) = -\log \Big[& (2\pi)^{-d/2} \cdot |(\|f_\theta(\mu_{k-1})\|^2_{V_{k-1}} + \eta_k) \otimes I_d|^{-1/2} && \text{(D.6)}\\
& \cdot \exp\left(-\tfrac{1}{2}(y_k - C_{k-1}f_\theta(\mu_{k-1}))^\top \left(\|f_\theta(\mu_{k-1})\|^2_{V_{k-1}} + \eta_k) \otimes I_d\right)^{-1} (y_k - C_{k-1}f_\theta(\mu_{k-1})\right)\Big] && \text{(D.7)}
\end{aligned}
$$

which simplifies to

$$
-\log \tilde{p}_\theta(y_k \mid y_{1:k-1}) = \frac{d}{2}\log(2\pi) + \frac{d}{2}\log(\|f_\theta(\mu_{k-1})\|^2_{V_{k-1}} + \eta_k) + \tfrac{1}{2}\frac{\|y_k - C_{k-1}f_\theta(\mu_{k-1})\|^2}{\|f_\theta(\mu_{k-1})\|^2_{V_{k-1}} + \eta_k}. \tag{D.8}
$$

E THE PROBABILISTIC MODEL TO HANDLE MISSING DATA

To obtain update rules that can explicitly handle missing data, we only need to modify the likelihood. When we receive an observation vector with missing entries, we model it as $z_k = m_k \odot y_k$ where $m_k \in \{0,1\}^d$ is a mask vector that contains zeros for missing entries and ones otherwise. We note that $z_k = M_k y_k$ where $M_k = \mathrm{diag}(m_k)$, which results in the likelihood $p(z_k|c, x_k) = \mathcal{N}(z_k; M_k C x_k, M_k R_k M_k^\top)$. The update rules for PSMF and the robust model, rPSMF, can be easily re-derived using this likelihood and are essentially identical to Algorithm 1 with masks. Here we discuss the case of PSMF with missing values, rPSMF with missing values is discussed in Supp. F.

We define the probabilistic model with missing data as

$$
\begin{aligned}
p(C) &= \mathcal{MN}(C; C_0, I_d, V_0), && \text{(E.1)}\\
p(x_0) &= \mathcal{N}(x_0; \mu_0, P_0), && \text{(E.2)}\\
p_\theta(x_k|x_{k-1}) &= \mathcal{N}(x_k; f_\theta(x_{k-1}), Q_k), && \text{(E.3)}\\
p(z_k|x_k, C) &= \mathcal{N}(z_k; M_k C x_k, M_k R_k M_k^\top). && \text{(E.4)}
\end{aligned}
$$

This model can explicitly handle the missing data when $(M_k)_{k\geq 1}$ (the missing data patterns) are given. The update rules for this model are defined using masks and are similar to the full data case. In what follows, we derive the update rules for this model by explicitly handling the masks and placing them into our updates formally. For the missing-data case, however, we need a minor approximation in the covariance update rule in order to keep the method efficient. Assume that we are given $\tilde{p}(c|z_{1:k-1}) = \mathcal{N}(c; c_{k-1}, V_{k-1} \otimes I_d)$ and the likelihood

$$
\tilde{p}(z_k|c, z_{1:k-1}) = \mathcal{N}(z_k; M_k C\bar{\mu}_k, \eta_k \otimes I_d) \tag{E.5}
$$

where

$$
\eta_k = \frac{\mathrm{Tr}(M_k R_k M_k^\top + M_k C_{k-1}\bar{P}_k C_{k-1}^\top M_k^\top)}{m}. \tag{E.6}
$$

In the sequel, we derive the update rules corresponding to the our method with missing data. The derivation relies on the proof of Prop. 1. We note that using (A.2), we can obtain the likelihood

$$
\tilde{p}(z_k|c, z_{1:k-1}) = \mathcal{N}(z_k; H_k c, \eta_k \otimes I_d) \tag{E.7}
$$

where $c = \mathrm{vec}(C)$ and $H_k = \bar{\mu}_k^\top \otimes M_k$. Deriving the posterior in the same way as in the proof of Prop. 1, and using the approximation $\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k \otimes M_k \approx \bar{\mu}_k^\top V_{k-1}\bar{\mu}_k \otimes I_d$, leaves us with the covariance update in the form

$$P_k = V_{k-1} \otimes I_d - \frac{V_{k-1}\bar{\mu}_k\bar{\mu}_k^\top V_{k-1}}{\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k} \otimes M_k. \tag{E.8}$$

Unlike the previous case, this covariance does not simplify to a form $P_k = V_k \otimes I_d$ easily. For this reason, we approximate it as

$$P_k \approx V_k \otimes I_d, \tag{E.9}$$

where V_k is in the same form of missing-data free updates. To update the mean, we proceed in a similar way as in the proof of Prop. 1 as well. Straightforward calculations lead to the update

$$C_k = C_{k-1} + \frac{(z_k - M_k C_{k-1}\bar{\mu}_k)\bar{\mu}_k^\top V_{k-1}}{\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k}, \qquad \text{for } k \geq 1. \tag{E.10}$$

To update x_k, once we fix C_{k-1}, everything straightforwardly follows by replacing C_{k-1} by $M_k C_{k-1}$ in the update rules for $(x_k)_{k\geq 1}$. Finally, the negative log-likelihood $\tilde{p}_\theta(z_k|z_{1:k-1})$ can be derived similarly to the non-missing case in Sec. 3.2.5, and equals

$$-\log \tilde{p}_\theta(z_k|z_{1:k-1}) \stackrel{c}{=} \tfrac{1}{2}\sum_{j=1}^{d} \log u_{jk} + \tfrac{1}{2}(z_k - M_k C_{k-1} f_\theta(\mu_{k-1}))^\top U_k^{-1}(z_k - M_k C_{k-1} f_\theta(\mu_{k-1})), \tag{E.11}$$

where $\stackrel{c}{=}$ denotes equality up to constants that do not depend on θ and $U_k = \|f_\theta(\mu_{k-1})\|^2_{V_{k-1}} \otimes M_k + \eta_k \otimes I_d$ is a d-dimensional diagonal matrix with elements u_{jk} for $j = 1, \dots, d$.

F THE ROBUST MODEL

Recall that the model definitions for robust PSMF are as follows

$$p(s) = \mathcal{IG}(s; \lambda_0/2, \lambda_0/2) \tag{F.1}$$
$$p(C \mid s) = \mathcal{MN}(C; C_0, I_d, sV_0)), \tag{F.2}$$
$$p(x_0 \mid s) = \mathcal{N}(x_0; \mu_0, sP_0), \tag{F.3}$$
$$p_\theta(x_k \mid x_{k-1}, s) = \mathcal{N}(x_k; f_\theta(x_{k-1}), sQ_0), \tag{F.4}$$
$$p(y_k \mid x_k, C, s) = \mathcal{N}(y_k; Cx_k, sR_0), \tag{F.5}$$

Before we present the derivation, we recall the following definitions.

Definition 1 (Inverse-Gamma Distribution). *The inverse-gamma distribution is given by*

$$\mathcal{IG}(s; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\left(\frac{1}{s}\right)^{\alpha+1} \exp\left(-\beta/s\right) \tag{F.6}$$

for $\alpha, \beta > 0$, and with $\Gamma(\cdot)$ the Gamma function.

Definition 2 (Multivariate t Distribution). *For $y \in \mathbb{R}^d$ the multivariate t distribution with λ degrees of freedom is*

$$\mathcal{T}(y; \mu, \Sigma, \lambda) = \frac{1}{(\pi\lambda)^{d/2}|\Sigma|^{1/2}}\frac{\Gamma((\lambda+d)/2)}{\Gamma(\lambda/2)}\left(1 + \frac{\Delta^2}{\lambda}\right)^{-(\lambda+d)/2} \tag{F.7}$$

where $\Delta^2 = (y-\mu)^\top\Sigma^{-1}(y-\mu)$.

Since we again assume the model to be Markovian, we extend the conditional independence and Markov properties (see, e.g., Särkkä, 2013) to the case with a scale variable

Property 1 (Conditional independence). *The measurement y_k given the coefficient x_k and scale variable s, is conditionally independent of past measurements and coefficients*

$$p(y_k \mid x_{1:k}, y_{1:k-1}, s) = p(y_k \mid x_k, s). \tag{F.8}$$

Property 2 (Markov property of coefficients)**.** *When conditioning on s the coefficients x_k form a Markov sequence, such that*

$$p(x_k \mid x_{1:k-1}, y_{1:k-1}, s) = p(x_k \mid x_{k-1}, s). \tag{F.9}$$

We also present the following lemma's used in the derivation.

Lemma 1. *For $y \in \mathbb{R}^d$ with $p(y \mid s) = \mathcal{N}(y; \mu, \Sigma)$ and $p(s) = \mathcal{IG}(s; \alpha, \beta)$ we have*

$$p(y) = \frac{1}{(2\pi\beta)^{d/2}|\Sigma|^{1/2}} \frac{\Gamma(\alpha + d/2)}{\Gamma(\alpha)} \left(1 + \frac{\Delta^2}{2\beta}\right)^{-(\alpha+d/2)} \tag{F.10}$$

$$p(s|y) = \mathcal{IG}(s; \alpha + d/2, \beta + \tfrac{1}{2}\Delta^2). \tag{F.11}$$

In particular, if $\alpha = \beta = \lambda/2$ then $p(y) = \mathcal{T}(y; \mu, \Sigma, \lambda)$.

Lemma 2. *If $p(s) = \mathcal{IG}(s; \alpha, \beta)$ and $\omega = \beta/\alpha$, then $\omega \cdot p(\omega s) = \mathcal{IG}(s; \alpha, \alpha)$.*

Proof.

$$\frac{\beta}{\alpha} p\left(\frac{\beta}{\alpha}s\right) = \frac{\beta}{\alpha}\frac{\beta^\alpha}{\Gamma(\alpha)}\left(\frac{\alpha}{\beta s}\right)^{\alpha+1} \exp\left(-\frac{\beta\alpha}{\beta s}\right) = \frac{\alpha^\alpha}{\Gamma(\alpha)}\left(\frac{1}{s}\right)^{\alpha+1} \exp\left(-\frac{\alpha}{s}\right) = \mathcal{IG}(s; \alpha, \alpha). \tag{F.12}$$

■

Lemma 3. *For a partitioned random variable $y = [y_a, y_b]^\top$ with $y_a \in \mathbb{R}^{d_a}$ and $y_b \in \mathbb{R}^{d_b}$ that follows a multivariate t distribution given by*

$$p(y) = p(y_a, y_b) = \mathcal{T}\left(\begin{bmatrix} y_a \\ y_b \end{bmatrix}; \begin{bmatrix} \mu_a \\ \mu_b \end{bmatrix}, \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ab}^\top & \Sigma_{bb} \end{bmatrix}, \lambda\right), \tag{F.13}$$

the marginal and conditional densities are given by

$$p(y_b) = \mathcal{T}(y_b; \mu_b, \Sigma_{bb}, \lambda) \tag{F.14}$$

$$p(y_a \mid y_b) = \mathcal{T}(y_a; \mu_{a|b}, \Sigma_{a|b}, \lambda_{a|b}), \tag{F.15}$$

with

$$\lambda_{a|b} = \lambda + d_b \tag{F.16}$$

$$\mu_{a|b} = \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(y_b - \mu_b) \tag{F.17}$$

$$\Sigma_{a|b} = \frac{\lambda + (y_b - \mu_b)^\top \Sigma_{bb}^{-1}(y_b - \mu_b)}{\lambda + d_b}\left(\Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ab}^\top\right). \tag{F.18}$$

Proof. See Roth (2012) for a derivation. ■

To derive inference in the robust model, we start from $k = 1$ and show how we perform filtering for an entire iteration. While this makes the description longer, we believe it to be more informative for the reader. We begin with prediction of x_1 given no history ($y_{1:0} = \emptyset$). The predictive distribution of x_1 is then

$$\tilde{p}(x_1 \mid y_{1:0}, s) = \int p(x_1 \mid x_0, s) p(x_0 \mid y_{1:0}, s)\, \mathrm{d}x_0 \tag{F.19}$$

$$\tilde{p}(x_1 \mid s) = \int p(x_1 \mid x_0, s) p(x_0 \mid s)\, \mathrm{d}x_0 \tag{F.20}$$

$$= \int \mathcal{N}(x_1; f_\theta(x_0), sQ_0)\mathcal{N}(x_0; \mu_0, sP_0)\, \mathrm{d}x_0 \tag{F.21}$$

$$= \mathcal{N}(x_1; f_\theta(\mu_0), s(Q_0 + F_1 P_0 F_1^\top)), \tag{F.22}$$

where F_1 is defined as in the main text. Writing $\bar{\mu}_1 = f_\theta(\mu_0)$ and $\bar{P}_1 = Q_0 + F_1 P_0 F_1^\top$ we get $\tilde{p}(x_1 \,|\, s) = \mathcal{N}(x_1; \bar{\mu}_1, s\bar{P}_1)$. Next, we move to the dictionary update. We first have

$$\tilde{p}(y_1 \,|\, c, y_{1:0}, s) = \int p(y_1 \,|\, c, x_1, s) p(x_1 \,|\, y_{1:0}, s) \,\mathrm{d}x_1 \tag{F.23}$$

$$\tilde{p}(y_1 \,|\, c, s) = \int p(y_1 \,|\, c, x_1, s) p(x_1 \,|\, s) \,\mathrm{d}x_1 \tag{F.24}$$

$$= \int \mathcal{N}(y_1; Cx_1, sR_0) \mathcal{N}(x_1; \bar{\mu}_1, s\bar{P}_1) \,\mathrm{d}x_1 \tag{F.25}$$

$$= \mathcal{N}(y_1; C\bar{\mu}_1, s(R_0 + C\bar{P}_1 C^\top)). \tag{F.26}$$

As in PSMF, we use the approximation $C\bar{P}_1 C^\top \approx \eta_1 \otimes I_d$ where $\eta_1 = \mathrm{Tr}(R_0 + C_0 \bar{P}_1 C_0^\top)/d$. We write this as $\tilde{p}(y_1 \,|\, c, s) = \mathcal{N}(y_1; H_1 c, sG_1)$ with $H_1 = \bar{\mu}_1^\top \otimes I_d$ and $G_1 = \eta_1 \otimes I_d$. We again assume $\tilde{p}(c \,|\, y_{1:0}, s) = \mathcal{N}(c; c_0, sL_0)$ using $L_0 = V_0 \otimes I_d$, such that

$$\tilde{p}(c, y_1 \,|\, y_{1:0}, s) = \tilde{p}(y_1 \,|\, c, y_{1:0}, s) \tilde{p}(c \,|\, y_{1:0}, s) \tag{F.27}$$

$$\tilde{p}(c, y_1 \,|\, s) = \tilde{p}(y_1 \,|\, c, s) \tilde{p}(c \,|\, s) \tag{F.28}$$

$$= \mathcal{N}(y_1; H_1 c, sG_1) \mathcal{N}(c; c_0, sL_0) \tag{F.29}$$

$$= \mathcal{N}\left(\begin{bmatrix} c \\ y_1 \end{bmatrix}; \begin{bmatrix} c_0 \\ H_1 c_0 \end{bmatrix}, s \begin{bmatrix} L_0 & L_0 H_1^\top \\ H_1 L_0 & H_1 L_0 H_1^\top + G_1 \end{bmatrix} \right) \tag{F.30}$$

Integrating out s in this expression gives

$$\tilde{p}(c, y_1) = \mathcal{T}\left(\begin{bmatrix} c \\ y_1 \end{bmatrix}; \begin{bmatrix} c_0 \\ H_1 c_0 \end{bmatrix}, \begin{bmatrix} L_0 & L_0 H_1^\top \\ H_1 L_0 & H_1 L_0 H_1^\top + G_1 \end{bmatrix}, \lambda_0 \right) \tag{F.31}$$

Conditioning on y_1 and using Lemma 3 yields $\tilde{p}(c \,|\, y_1) = \mathcal{T}(c; c_1, L_1, \lambda_0 + d)$ with

$$c_1 = c_0 + L_0 H_1^\top \left[H_1 L_0 H_1^\top + G_1 \right]^{-1} (y_1 - H_1 c_0) \tag{F.32}$$

$$L_1 = \phi_1 \left(L_0 H_1^\top \left[H_1 L_0 H_1^\top + G_1 \right]^{-1} H_1 L_0 \right) \tag{F.33}$$

$$\varphi_1 = \frac{\lambda_0 + (y_1 - H_1 c_0)^\top \left[H_1 L_0 H_1^\top + G_1 \right]^{-1} (y_1 - H_1 c_0)}{\lambda_0 + d}. \tag{F.34}$$

This is the **robust PSMF dictionary update**. We see that the mean is updated as in PSMF by comparing to (B.1), and that the covariance update has an additional multiplicative factor φ_1. These expressions can be simplified by plugging in the definitions of L_0, H_1, and G_1, as in Supp. B. Observe that $\tilde{p}(c \,|\, y_1)$ can no longer be written as an infinite scale mixture with scale variable s, as they now differ in degrees of freedom. We will revisit this point below.

For the coefficient update we proceed analogously. First, note that

$$\tilde{p}(y_1 \,|\, x_{0:1}, s) = \int p(y_1 \,|\, c, x_{0:1}, s) p(c \,|\, y_{1:0}, s) \,\mathrm{d}c \tag{F.35}$$

$$\tilde{p}(y_1 \,|\, x_1, s) = \int p(y_1 \,|\, c, x_1, s) p(c \,|\, s) \,\mathrm{d}c \tag{F.36}$$

$$= \int \mathcal{N}(y_1; (x_1^\top \otimes I_d) c, sR_0) \mathcal{N}(c; c_0, sL_0) \,\mathrm{d}c \tag{F.37}$$

$$= \mathcal{N}(y_1; (x_1^\top \otimes I_d) c_0, s(R_0 + x_1^\top V_0 x_1 \otimes I_d)) \tag{F.38}$$

As in the main text, we use the approximation $x_1^\top V_0 x_1 \approx \bar{\mu}_1^\top V_0 \bar{\mu}_1$ and introduce

$$\bar{R}_0 = R_0 + \bar{\mu}_1^\top V_0 \bar{\mu}_1, \tag{F.39}$$

such that $\tilde{p}(y_1 \mid x_1, s) = \mathcal{N}(y_1; C_0 x_1, s\bar{R}_0)$. We then find the joint distribution between x_1 and y_1 as follows

$$\tilde{p}(x_1, y_1 \mid y_{1:0}, s) = \tilde{p}(y_1 \mid y_{1:0}, x_1, s)\tilde{p}(x_1 \mid y_{1:0}, s) \tag{F.40}$$
$$\tilde{p}(x_1, y_1 \mid s) = \tilde{p}(y_1 \mid x_1, s)\tilde{p}(x_1 \mid s) \tag{F.41}$$
$$= \mathcal{N}(y_1; C_0 x_1, s\bar{R}_0)\mathcal{N}(x_1; \bar{\mu}_1, s\bar{P}_1) \tag{F.42}$$
$$= \mathcal{N}\left(\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}; \begin{bmatrix} \bar{\mu}_1 \\ C_0\bar{\mu}_1 \end{bmatrix}, s\begin{bmatrix} \bar{P}_1 & \bar{P}_1 C_0^\top \\ C_0\bar{P}_1 & C_0\bar{P}_1 C_0^\top + \bar{R}_0 \end{bmatrix}\right). \tag{F.43}$$

Integrating out s in this expression gives

$$\tilde{p}(x_1, y_1) = \mathcal{T}\left(\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}; \begin{bmatrix} \bar{\mu}_1 \\ C_0\bar{\mu}_1 \end{bmatrix}, \begin{bmatrix} \bar{P}_1 & \bar{P}_1 C_0^\top \\ C_0\bar{P}_1 & C_0\bar{P}_1 C_0^\top + \bar{R}_0 \end{bmatrix}, \lambda_0\right). \tag{F.44}$$

Conditioning on y_1 and using Lemma 3 gives $p(x_1 \mid y_1) = \mathcal{T}(x_1; \mu_1, P_1, \lambda_0 + d)$ with

$$\mu_1 = \bar{\mu}_1 + \bar{P}_1 C_0^\top \left[C_0\bar{P}_1 C_0^\top + \bar{R}_0\right]^{-1}(y_1 - C_0\bar{\mu}_1) \tag{F.45}$$
$$P_1 = \omega_1\left(\bar{P}_1 - \bar{P}_1 C_0^\top \left[C_0\bar{P}_1 C_0^\top + \bar{R}_0\right]^{-1} C_0\bar{P}_1\right) \tag{F.46}$$
$$\omega_1 = \frac{\lambda_0 + (y_1 - C_0\bar{\mu}_1)^\top \left[C_0\bar{P}_1 C_0^\top + \bar{R}_0\right]^{-1}(y_1 - C_0\bar{\mu}_1)}{\lambda_0 + d}. \tag{F.47}$$

This is the **robust PSMF coefficient update**. Again we see that the mean update for μ_1 is the same as in vanilla PSMF, while the covariance update has an additional multiplicative factor ω_1. By introducing $\Delta_1^2 = (y_1 - C_0\bar{\mu}_1)^\top \left[C_0\bar{P}_1 C_0^\top + \bar{R}_0\right]^{-1}(y_1 - C_0\bar{\mu}_1)$ we can simplify this factor to $\omega_1 = (\lambda_0 + \Delta_1^2)/(\lambda_0 + d)$.

Finally, we can compute the posterior of the scale variable, s, using Bayes' theorem,

$$\tilde{p}(s \mid y_1) = \frac{\tilde{p}(y_1 \mid s)p(s)}{\tilde{p}(y_1)}. \tag{F.48}$$

We can obtain $\tilde{p}(y_1 \mid s)$ from (F.43), which yields

$$\tilde{p}(y_1 \mid s) = \mathcal{N}(y_1; C_0\bar{\mu}_1, s(C_0\bar{P}_1 C_0^\top + \bar{R}_0)). \tag{F.49}$$

Integrating out s gives $\tilde{p}(y_1) = \mathcal{T}(y_1; C_0\bar{\mu}_1, C_0\bar{P}_1 C_0^\top + \bar{R}_0, \lambda_0)$. Thus, by Lemma 1 we have

$$\tilde{p}(s \mid y_1) = \mathcal{IG}(s; (\lambda_0 + d)/2, (\lambda_0 + \Delta_1^2)/2). \tag{F.50}$$

Having now observed y_1, we proceed with the next iteration. Note that from the coefficient update we have obtained $p(x_1 \mid y_1) = \mathcal{T}(x_1; \mu_1, P_1, \lambda_0 + d)$. We can write this as a infinite scale mixture by defining $u \sim \mathcal{IG}(u; (\lambda_0 + d)/2, (\lambda_0 + d)/2)$ and introducing $p(x_1 \mid y_1, u) = \mathcal{N}(x_1; \mu_1, uP_1)$. The model definitions give the coefficient dynamics in terms of s, as $p(x_2 \mid x_1, s) = \mathcal{N}(x_2; f_\theta(x_1), sQ_0)$. This can be written in terms of u by a simple change of variables $u = \omega_1^{-1}s$ and using Lemma 2 and (F.50), since

$$\tilde{p}(x_2 \mid x_1, y_1) = \int p(x_2 \mid x_1, s)p(s \mid y_1)\,\mathrm{d}s \tag{F.51}$$
$$= \int \mathcal{N}(x_2; f_\theta(x_1), sQ_0)\mathcal{IG}(s; (\lambda_0 + d)/2, (\lambda_0 + \Delta_1^2)/2)\,\mathrm{d}s \tag{F.52}$$
$$= \int \mathcal{N}(x_2; f_\theta(x_1), u \cdot \omega_1 Q_0)\mathcal{IG}(u; (\lambda_0 + d)/2, (\lambda_0 + d)/2)\,\mathrm{d}u \tag{F.53}$$

where we find $p(x_2 \mid x_1, u) = \mathcal{N}(x_2; f_\theta(x_1), u \cdot \omega_1 Q_0)$. We then have that

$$\tilde{p}(x_2 \mid y_1, u) = \int \tilde{p}(x_2 \mid x_1, u)\tilde{p}(x_1 \mid y_1, u)\,\mathrm{d}x_1 \tag{F.54}$$
$$= \int \mathcal{N}(x_2; f_\theta(x_1), u \cdot \omega_1 Q_0)\mathcal{N}(x_1; \mu_1, uP_1)\,\mathrm{d}x_1, \tag{F.55}$$

which we recognize to be analogous to (F.21). This expression also reveals how the noise covariance Q_0 is updated, as we may simply define $Q_1 = \omega_1 Q_0$. This gives $\tilde{p}(x_2 \,|\, y_1, u) = \mathcal{N}(x_2; \bar{\mu}_2, u\bar{P}_2)$ with $\bar{\mu}_2$ and $\bar{P}_2$ analogous to $\bar{\mu}_1$ and $\bar{P}_1$ above.

Similar reasoning can be applied to obtain the predictive distribution of y_2. From the dictionary update we have obtained $\tilde{p}(c \,|\, y_1) = \mathcal{T}(c; c_1, L_1, \lambda_0 + d)$, which we can also write as a scale mixture with u as $\tilde{p}(c \,|\, y_1, u) = \mathcal{N}(c; c_1, uL_1)$. The model definition gives $p(y_2 \,|\, x_2, C, s) = \mathcal{N}(y_2; Cx_2, sR_0)$. Again writing this in terms of u by using the change of variables $u = \omega_1^{-1} s$ and Lemma 2 and (F.50), yields $p(y_2 \,|\, x_2, C, u) = \mathcal{N}(y_2; Cx_2, u \cdot \omega_1 R_0)$. Combining these expressions gives

$$\tilde{p}(y_2 \,|\, c, y_1, u) = \int \tilde{p}(y_2 \,|\, x_2, C, u)\tilde{p}(c \,|\, y_1, u) \,\mathrm{d}c = \int \mathcal{N}(y_2; (x_2^\top \otimes I_d)c, u \cdot \omega_1 R_0)\mathcal{N}(c; c_1, uL_1) \,\mathrm{d}c, \tag{F.56}$$

which is analogous to (F.37). We also see that we can define $R_1 = \omega_1 R_0$ to update the measurement noise covariance.

We observe in the above derivation that after completing an entire iteration we have obtained a new scale variable $u \sim \mathcal{IG}(u; (\lambda_0 + d)/2, (\lambda_0 + d)/2)$, and that we have found update rules for the noise covariances Q and R. This procedure is repeated at every step, and we can define appropriate notation for this process by setting $s_0 = s$ and $s_1 = u$, and generally have scale variables $s_k \sim \mathcal{IG}(s_k; \lambda_k/2, \lambda_k/2)$ with $\lambda_k = \lambda_{k-1} + d$. Thus, $s_k = \omega_k^{-1} s_{k-1}$ with $\omega_k = (\lambda_{k-1} + \Delta_k^2)/(\lambda_{k-1} + d)$, which corresponds to Tronarp et al. (2019). The noise covariances are clearly updated as $Q_k = \omega_k Q_{k-1}$ and $R_k = \omega_k R_{k-1}$. Algorithm 2 summarizes the steps of robust PSMF, including steps for parameter estimation using both the iterative and recursive approaches.

For completeness, we give the approximate negative marginal likelihood $\tilde{p}_\theta(y_k \,|\, y_{1:k-1})$, similar to Sec. 3.2.5. It follows that

$$\tilde{p}_\theta(y_k \,|\, y_{1:k-1}) = \iint \tilde{p}(y_k \,|\, y_{1:k-1}, c, s_{k-1})\tilde{p}(c \,|\, y_{1:k-1}, s_{k-1}) \,\mathrm{d}c \,\mathrm{d}s_{k-1} \tag{F.57}$$

$$= \iint \mathcal{N}(y_k; H_k c, s_{k-1} G_k)\mathcal{N}(c; c_{k-1}, s_{k-1} L_{k-1}) \,\mathrm{d}c \,\mathrm{d}s_{k-1} \tag{F.58}$$

$$= \int \mathcal{N}(y_k; H_k c_{k-1}, s_{k-1}(H_k L_{k-1} H_k^\top + G_k)) \,\mathrm{d}s_{k-1} \tag{F.59}$$

$$= \mathcal{T}(y_k; H_k c_{k-1}, H_k L_{k-1} H_k^\top + G_k, \lambda_{k-1}) \tag{F.60}$$

With $H_k c_{k-1} = C_{k-1}\bar{\mu}_k$ and $H_k L_{k-1} H_k^\top + G_k = (\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k) \otimes I_d$ where $\bar{\mu}_k = f_\theta(\mu_{k-1})$, we find after a brief algebraic exercise that

$$-\log \tilde{p}_\theta(y_k \,|\, y_{1:k-1}) \stackrel{c}{=} \frac{d}{2} \log\left(\|f_\theta(\mu_{k-1})\|_{V_{k-1}}^2 + \eta_k\right) \tag{F.61}$$

$$+ \left(\frac{\lambda_{k-1} + d}{2}\right) \log\left(1 + \frac{\|y_k - C_{k-1} f_\theta(\mu_{k-1})\|^2}{\lambda_{k-1}\left(\|f_\theta(\mu_{k-1})\|_{V_{k-1}}^2 + \eta_k\right)}\right) \tag{F.62}$$

where $\stackrel{c}{=}$ again denotes equality up to terms independent of θ. Finally, we note that handling missing values in rPSMF is straightforward and follows the same reasoning as for PSMF in Supp. E.

G ADDITIONAL DETAILS FOR THE EXPERIMENTS

G.1 Experiment 1

Optimization In this experiment, we have used the Adam optimizer Kingma and Ba (2015). In particular, instead of implementing the gradient step (26), we replace it with the Adam optimizer. In order to do so, we define the gradient as $g_i = \nabla \log \tilde{p}_\theta(y_{1:n})\Big|_{\theta=\theta_{i-1}}$. Upon computing the gradient g_i, we first compute the running averages

$$m_i = \beta_1 m_{i-1} + (1 - \beta_1) g_i \tag{G.1}$$

$$v_i = \beta_2 v_{i-1} + (1 - \beta_2)(g_i \odot g_i), \tag{G.2}$$

Algorithm 2 Iterative and recursive rPSMF

Initialize γ, θ_0, C_0, V_0, μ_0, P_0, Q_0, R_0.
for $i \geq 1$ **do**
  $C_0 = C_T$, $\mu_0 = \mu_T$, $P_0 = P_T$, $V_0 = V_T$.
  for $1 \leq k \leq T$ **do**
    Predictive mean of x_k: $\bar{\mu}_k = f_{\theta_{i-1}}(\mu_{k-1})$ or $\bar{\mu}_k = f_{\theta_{k-1}}(\mu_{k-1})$
    Predictive covariance of x_k

$$\bar{P}_k = F_k P_{k-1} F_k^\top + Q_{k-1}, \quad \text{where} \quad F_k = \frac{\partial f(x)}{\partial x}\Big|_{x=\bar{\mu}_{k-1}}$$

    Compute scaling factor for the dictionary update

$$\varphi_k = \frac{\lambda_{k-1}}{\lambda_{k-1}+d} + \frac{(y_k - C_{k-1}\bar{\mu}_k)^\top (y_k - C_{k-1}\bar{\mu}_k)}{(\lambda_{k-1}+d)(\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k)}$$

    where $\eta_k = \mathrm{Tr}(C_{k-1}\bar{P}_k C_{k-1}^\top + R_{k-1})/d$.
    Mean and covariance updates of the dictionary

$$C_k = C_{k-1} + \frac{(y_k - C_{k-1}\bar{\mu}_k)\bar{\mu}_k^\top V_{k-1}^\top}{\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k} \quad \text{and} \quad V_k = \varphi_k \left(V_{k-1} - \frac{V_{k-1}\bar{\mu}_k \bar{\mu}_k^\top V_{k-1}}{\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k} \right)$$

    Compute scaling factor for the coefficient update

$$\omega_k = \frac{\lambda_{k-1} + (y_k - C_{k-1}\bar{\mu}_k)^\top S_k^{-1}(y_k - C_{k-1}\bar{\mu}_k)}{\lambda_{k-1}+d}$$

    where $S_k = C_{k-1}\bar{P}_k C_{k-1}^\top + \bar{R}_{k-1}$ and $\bar{R}_{k-1} = R_{k-1} + \bar{\mu}_k^\top V_{k-1}\bar{\mu}_k \otimes I_d$.
    Mean and covariance updates of coefficients

$$\mu_k = \bar{\mu}_k + \bar{P}_k C_{k-1}^\top S_k^{-1}(y_k - C_{k-1}\bar{\mu}_k) \quad \text{and} \quad P_k = \omega_k(\bar{P}_k - \bar{P}_k C_{k-1}^\top S_k^{-1} C_{k-1}\bar{P}_k)$$

    Update noise covariances: $Q_k = \omega_k Q_{k-1}$ and $R_k = \omega_k R_{k-1}$
    Update degrees of freedom: $\lambda_k = \lambda_{k-1} + d$.
    Parameter update: $\theta_k = \theta_{k-1} + \gamma \nabla \log \tilde{p}_\theta(y_k | y_{1:k-1})\big|_{\theta=\theta_{k-1}}$ ▷ recursive version
  Parameter update: $\theta_i = \theta_{i-1} + \gamma \sum_{k=1}^T \nabla \log \tilde{p}_\theta(y_k | y_{1:k-1})\big|_{\theta=\theta_{i-1}}$. ▷ iterative version

which is then corrected as

$$\hat{m}_i = \frac{m_i}{1-\beta_1^i} \tag{G.3}$$

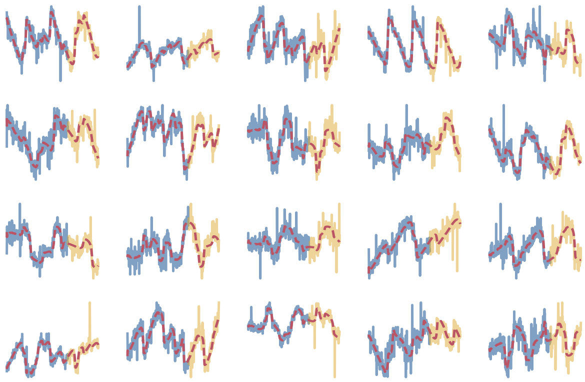
$$\hat{v}_i = \frac{v_i}{1-\beta_2^i}. \tag{G.4}$$

Finally the parameter update is computed as

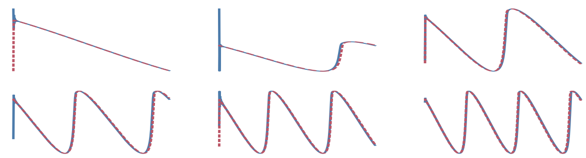
$$\theta_i = \mathsf{Proj}_\Theta \left(\theta_{i-1} + \gamma \frac{\hat{m}_i}{\sqrt{\hat{v}_i} + \epsilon} \right), \tag{G.5}$$

where Proj denotes the projection operator which constrains the parameter to stay positive in each dimension where $\Theta = \mathbb{R}_+ \times \cdots \times \mathbb{R}_+ \subset \mathbb{R}^6$ which is implemented by simple max operators. We choose the standard parameterization with $\gamma = 10^{-3}, \beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 10^{-8}$.

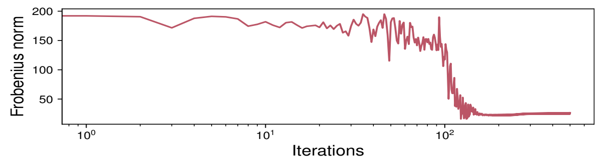
In these experiments we use an observed time series of length 500 and a series of unobserved future data of length 250. Fig. 1 corresponds to the figure in the main text, but additionally shows how the underlying subspace is recovered and how the Frobenius norm between the reconstructed data and the true data decreases with the number of iterations. Fig. 2 shows a similar result for the PSMF method on normally-distributed data.



(a) Observed time series (blue) with unobserved future data (yellow) and the reconstruction (red).

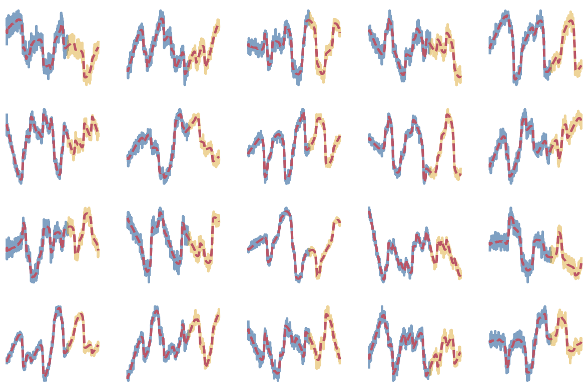


(b) True (blue) and predicted (red) subspace.

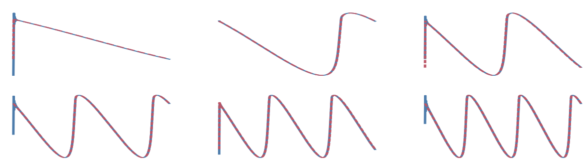


(c) Reconstruction error $\|Y - CX\|_F^2$.

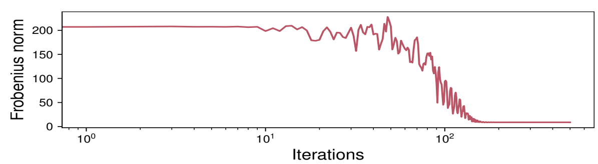
Figure 1: Fitting rPSMF on synthetic data with t-distributed noise. Figure (a) illustrates the fit to the observed and unobserved measurements. Figure (b) contains the true and reconstructed subspace, and (c) shows the reconstruction error over outer iterations of the iterative algorithm.



(a) Observed time series (blue) with unobserved future data (yellow) and the reconstruction (red).



(b) True (blue) and predicted (red) subspace.



(c) Reconstruction error $\|Y - CX\|_F^2$.

Figure 2: Fitting PSMF on synthetic data with normally distributed noise. Figure (a) illustrates the fit to the observed and unobserved measurements. Figure (b) contains the true and reconstructed subspace, and (c) shows the reconstruction error over outer iterations of the iterative algorithm.

G.2 Experiment 2

G.2.1 Data generation and the experimental setup

We generate periodic time series using pendulum differential equations as the true subspace. For this experiment, we generate $d = 20$ dimensional data where $d_2 = 3$ of them undergo a structural change. In order to test the method, we generate 1000 synthetic datasets. One such dataset is given in Fig. 3. We generate data with $n = 1200$ and use the data after the data point $n_0 = 400$ to estimate changepoints, as PSMF has to converge to a stable regime before it can be used to detect changepoints. The true changepoint is at $n_c = 601$.

G.2.2 The GP subspace model

In this subsection, we provide the details of the discretization of the Matérn-3/2 SDE. Particularly, we consider the SDE Särkkä et al. (2013)

$$\frac{\mathrm{d}\mathsf{x}_i(t)}{\mathrm{d}t} = \mathsf{F}\mathsf{x}_i(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_i(t) \tag{G.6}$$

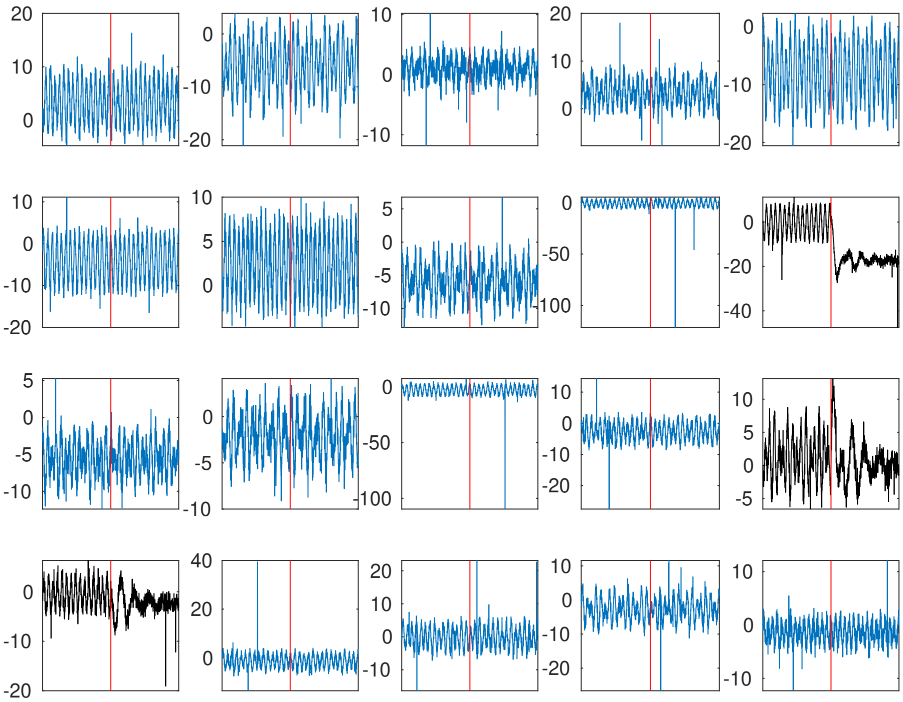


Figure 3: One instance of the 1000 different synthetic datasets used in Sec. 4.3. The dimensions which exhibit a structural change can be seen in black. The data contain outliers and the true changepoint can be seen as marked by the vertical red line.

where $\mathsf{x}_i(t) = [x_i(t),\, \mathrm{d}x_i(t)/\,\mathrm{d}t]$ and $\kappa = \sqrt{2\nu}/\ell$ and

$$\mathsf{F} = \begin{bmatrix} 0 & 1 \\ -\kappa^2 & -2\kappa \end{bmatrix}. \tag{G.7}$$

Given a step-size γ, the SDE (G.6) can be written as a linear dynamical system

$$x_{i,k} = A_i x_{i,k-1} + Q_i^{1/2} u_{i,k} \tag{G.8}$$

where $A_i = \mathrm{expm}(\gamma F)$ where expm denotes the matrix exponential and $Q_i = P_\infty - A_i P_\infty A_i^\top$ and

$$P_\infty = \begin{bmatrix} \sigma^2 & 0 \\ 0 & 3\sigma^2/\ell^2 \end{bmatrix}. \tag{G.9}$$

Finally, we construct our dynamical system as

$$x_k = A x_{k-1} + Q^{1/2} u_k \tag{G.10}$$

where $x_k = [x_{1,k}, \ldots, x_{r,k}]^\top \in \mathbb{R}^{2r}$ and

$$A = I_r \otimes A_i \qquad \text{and} \qquad Q = I_r \otimes Q_i. \tag{G.11}$$

Using these system matrices, we define $H_i = [1, 0]$ and $H = I_r \otimes H_i$ and finally define the probabilistic model

$$p(C) = \mathcal{MN}(C; C_0, I_d, V_0), \tag{G.12}$$
$$p(x_0) = \mathcal{N}(x_0; \mu_0, P_0), \tag{G.13}$$
$$p(x_k|x_{k-1}) = \mathcal{N}(x_k; A x_{k-1}, Q), \tag{G.14}$$
$$p(y_k|x_k, C) = \mathcal{N}(y_k; C H x_k, R). \tag{G.15}$$

Inference in this model can be done via a simple modification of the Algorithm 1 where H matrix is involved in the computations. Fig. 4 illustrates the learned GP features $r = 4$ and two change points.

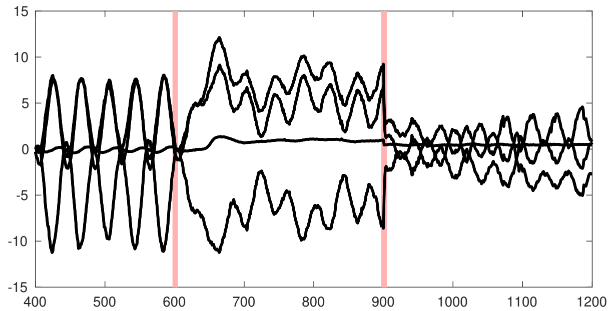


Figure 4: An illustration of the learned GP features vs. true changepoints for $r = 4$ and two changepoints.

G.3 Experiment 3

All experiments where run on a Linux machine with an AMD Ryzen 5 3600 processor and 32GB of memory. Additional results on different missing percentages are shown in Table 4 and Table 5. We again observe excellent imputation performance of the proposed methods.

Table 4: Imputation error and runtime on several datasets using 20% and 40% missing values, averaged over 100 random repetitions. An asterisk marks offline methods.

(a) 20% missing data

	Imputation RMSE					Runtime (s)				
	NO_2	PM10	PM25	S&P500	Gas	NO_2	PM10	PM25	S&P500	Gas
PSMF	**5.52** (0.10)	**7.26** (0.38)	3.42 (0.52)	9.95 (1.93)	**4.19** (0.57)	2.71	2.59	1.92	9.42	101.16
rPSMF	5.53 (0.14)	7.47 (0.46)	**3.40** (0.52)	**9.29** (1.41)	4.56 (0.51)	2.91	2.73	2.03	13.98	122.57
MLE-SMF	11.03 (0.51)	9.46 (0.37)	4.81 (0.63)	30.23 (1.02)	87.12 (14.84)	2.48	2.39	1.71	9.52	92.09
TMF	7.60 (0.14)	7.95 (0.30)	4.43 (0.50)	34.96 (1.00)	73.70 (8.85)	1.03	0.97	0.72	4.19	35.35
PMF*	10.47 (0.08)	10.46 (0.26)	3.97 (0.48)	40.07 (1.80)	23.54 (0.06)	2.14	1.90	0.68	3.12	31.78
BPMF*	9.03 (0.18)	8.39 (0.28)	3.61 (0.49)	27.36 (0.93)	17.70 (0.17)	3.11	4.48	3.05	4.15	92.50

(b) 40% missing data

	Imputation RMSE					Runtime (s)				
	NO_2	PM10	PM25	S&P500	Gas	NO_2	PM10	PM25	S&P500	Gas
PSMF	6.06 (0.18)	7.72 (0.28)	3.77 (0.23)	13.87 (3.06)	**8.75** (1.63)	2.77	2.62	1.92	9.12	100.68
rPSMF	**5.96** (0.27)	**7.68** (0.57)	**3.67** (0.29)	**12.36** (4.39)	9.03 (2.28)	2.92	2.77	2.02	13.30	109.38
MLE-SMF	11.30 (0.49)	9.55 (0.30)	4.93 (0.31)	30.14 (0.80)	125.54 (26.65)	2.54	2.38	1.70	9.59	85.11
TMF	7.90 (0.12)	8.27 (0.21)	4.86 (0.31)	34.78 (0.76)	66.27 (10.60)	0.98	0.97	0.73	4.13	32.01
PMF*	10.54 (0.05)	10.53 (0.15)	4.11 (0.13)	41.53 (1.81)	24.12 (0.06)	1.73	1.51	0.54	2.43	24.75
BPMF*	9.46 (0.21)	8.64 (0.18)	3.72 (0.12)	27.91 (0.64)	19.10 (0.37)	4.26	4.07	2.92	3.16	82.44

H CONVERGENCE DISCUSSION

To gain insights in the convergence of our method, we have designed a simplified setup where the latent state trajectory is a one-dimensional random walk and observations are four-dimensional, and we have simulated a dataset consisting of size 1,000 where $C \in \mathbb{R}^4$. We run the KF with the ground-truth value $C^\star$. We also run the iterative PSMF which also estimates C as well as the hidden states. We have computed the distance between the sequence of optimal (Gaussian) filters constructed by the KF and the filters of the iterative PSMF in terms of the

Table 5: Average coverage proportion of the missing data by the 2σ uncertainty bars of the posterior predictive estimates for 20% and 40% missing values, averaged over 100 repetitions.

(a) 20% missing data

	NO_2	PM10	PM25	S&P500	Gas
PSMF	0.79	0.79	**0.93**	0.85	**0.93**
rPSMF	**0.89**	**0.92**	0.91	**0.86**	0.90
MLE-SMF	0.46	0.59	0.83	0.51	0.61

(b) 40% missing data

	NO_2	PM10	PM25	S&P500	Gas
PSMF	0.71	0.73	**0.89**	0.78	**0.83**
rPSMF	**0.79**	**0.84**	0.81	**0.79**	0.79
MLE-SMF	0.40	0.53	0.77	0.44	0.49

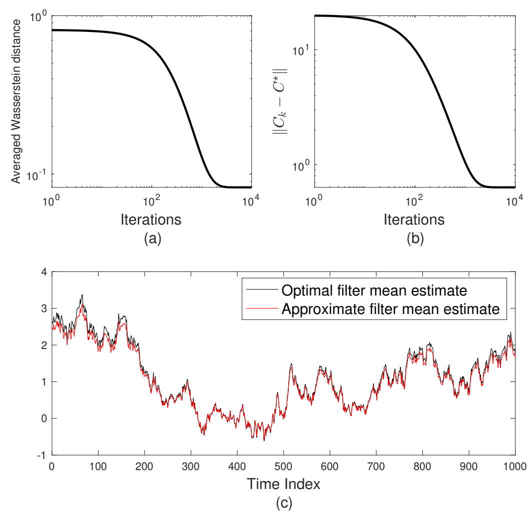


Figure 5: (a) Convergence of the approximate posterior and true posterior (with true $C^\star$) in averaged Wasserstein distance for iterative PSMF. (b) Convergence of the mean C_k to $C^\star$. (c) Filter estimates given by the iterative PSMF and the optimal filter.

averaged Wasserstein distance over the path:

$$\overline{W}_2(t) := \frac{1}{t}\sum_{k=1}^{t} W_2(p_\star(x_k|y_{1:k}), \tilde{p}(x_k|y_{1:k})). \tag{H.1}$$

We observe that the distance between the optimal and approximate filters over the entire path is uniformly bounded (see Fig. 5(a)). We also observe that $C_k \to C^\star$ for this case, see Fig. 5(b) and show the mean estimates are sufficiently close (Fig. 5(c)).

More precisely, we simulate the following state-space model

$$p(x_0) = \mathcal{N}(x_0; \mu_0, P_0), \tag{H.2}$$
$$p(x_k|x_{k-1}) = \mathcal{N}(x_k; x_{k-1}, Q), \tag{H.3}$$
$$p(y_k|x_k, C^\star) = \mathcal{N}(y_k; C^\star x_k, R), \tag{H.4}$$

where $C^\star \in \mathbb{R}^4$ and $x_k \in \mathbb{R}$, which leads to $y_k \in \mathbb{R}^4$. In this case, the identifiability problem is alleviated since $C^\star$ is a vector and we can test empirically whether the posterior provided by the PSMF for the states $p(x_k|y_{1:k})$ converges to the true posterior of the states $p_\star(x_k|y_{1:k})$.

Note that, the PSMF provides the filtering distribution of states as a Gaussian

$$\tilde{p}(x_k|y_{1:k}) = \mathcal{N}(x_k; \mu_k, P_k) \tag{H.5}$$

where μ_k, P_k are defined within Algorithm 1. Since the data is generated from the model using $C^\star$, we also compute the optimal Kalman filter with $C^\star$ which we denote as $p_\star(x_k|y_{1:k})$. In order to test the convergence between the approximate filter provided by the PSMF $\tilde{p}(x_k|y_{1:k})$ and the true filter $p_\star(x_k|y_{1:k})$, we use the Wasserstein-2 distance which is defined as

$$W_2(\mu,\nu) = \inf_{\Gamma \in \mathcal{C}(\mu,\nu)} \iint \|x-y\|^2 \Gamma(\,\mathrm{d}x,\,\mathrm{d}y) \tag{H.6}$$

where $\mathcal{C}(\mu,\nu)$ is the set of couplings whose marginals are μ and ν respectively. This Wasserstein-2 distance can be computed in closed form for two Gaussians, e.g., for $\mu = \mathcal{N}(\mu_1, \Sigma_1)$ and $\nu = \mathcal{N}(\mu_2, \Sigma_2)$, we have

$$W_2(\mu,\nu)^2 = \|\mu_1 - \mu_2\|^2 + \mathrm{Tr}(\Sigma_1 + \Sigma_2 - 2(\Sigma_2^{1/2}\Sigma_1\Sigma_2^{1/2})^{1/2}). \tag{H.7}$$

Hence, for a given sequence of filters $(\tilde{p}(x_k|y_{1:k}))_{k\geq 1}$ and $(p_\star(x_k|y_{1:k}))_{k\geq 1}$, we define the averaged Wasserstein distance for time t as

$$\overline{W}_2(t) = \frac{1}{t}\sum_{k=1}^{t} W_2(\tilde{p}(x_k|y_{1:k}), p_\star(x_k|y_{1:k})). \tag{H.8}$$

One can see from Fig. 5 that $\lim_{t\to\infty} \overline{W}_2(t) < \infty$ which implies that a convergence result can be proven for our method. We leave this exciting direction to future work.

I ADDITIONAL RESULTS FOR RECURSIVE PSMF

In this section, we present an additional result using recursive PSMF to demonstrate the scalability of our method in a purely streaming setting.

Using the same setting of Sec. 4.1, we use a longer sequence ($n = 4000$) with an additional prediction sequence of length 800. This presents a challenging setting as we do not iterate over data and the algorithm observes the training data only once (i.e. the streaming setting). As can be seen from Fig. 6, recursive PSMF learns the underlying dynamics and has a successful out-of-sample prediction performance, even with a relatively long sequence into the future. This demonstrates the recursive version of our method can be used in a setting where iterating over data multiple times is impractical.

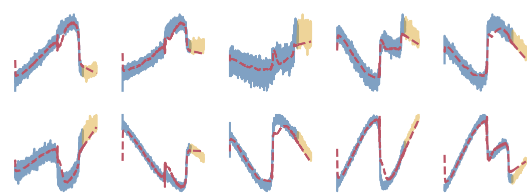


Figure 6: Recursive PSMF. Observed time series (blue) with unobserved future data (yellow) and the reconstruction from the model (red).