
eXponential FAmily Dynamical Systems (XFADS): Large-scale nonlinear Gaussian state-space modeling

Matthew Dowling

Champalimaud Research, Champalimaud Foundation, Portugal
matthew.dowling@research.fchampalimaud.org

Yuan Zhao

National Institute of Mental Health, USA
yuan.zhao@nih.gov

Il Memming Park

Champalimaud Research, Champalimaud Foundation, Portugal
memming.park@research.fchampalimaud.org

Abstract

State-space graphical models and the variational autoencoder framework provide a principled apparatus for learning dynamical systems from data. State-of-the-art probabilistic approaches are often able to scale to large problems at the cost of flexibility of the variational posterior or expressivity of the dynamics model. However, those consolidations can be detrimental if the ultimate goal is to learn a generative model capable of explaining the spatiotemporal structure of the data and making accurate forecasts. We introduce a low-rank structured variational autoencoding framework for nonlinear Gaussian state-space graphical models capable of capturing dense covariance structures that are important for learning dynamical systems with predictive capabilities. Our inference algorithm exploits the covariance structures that arise naturally from sample based approximate Gaussian message passing and low-rank amortized posterior updates – effectively performing approximate variational smoothing with time complexity scaling linearly in the state dimensionality. In comparisons with other deep state-space model architectures our approach consistently demonstrates the ability to learn a more predictive generative model. Furthermore, when applied to neural physiological recordings, our approach is able to learn a dynamical system capable of forecasting population spiking and behavioral correlates from a small portion of single trials.

1 Introduction

State-space models (SSM) are invaluable for understanding the temporal structure of complex natural phenomena through their underlying dynamics¹⁻³. While engineering or physics problems often assume the dynamical laws of the system of interest are known to a high degree of accuracy, in an unsupervised data-driven investigation, they have to be learned from the observed data. The variational autoencoder (VAE) framework makes it possible to jointly learn the parameters of the state-space description and an inference network to amortize posterior computation of the unknown latent state⁴⁻⁷. However, it can be challenging to structure the variational approximation and design an inference network that permits fast evaluation of the loss function (evidence lower bound or ELBO) while preserving the temporal structure of the posterior.

In this work, we develop a structured variational approximation, approximate ELBO, and inference network architecture for generative models specified by nonlinear dynamical systems with Gaussian state noise. Our main contributions are as follows,

- (i) A structured amortized variational approximation that combines the prior dynamics with low-rank data updates to parameterize Gaussian distributions with dense covariance matrices,
- (ii) Conceptualizing the approximate smoothing problem as an approximate filtering problem for pseudo-observations that encode a representation of current and future data, and,
- (iii) An inference algorithm that scales $\mathcal{O}(TL(Sr + S^2 + r^2))$ – made possible by exploiting the low-rank structure of the amortization network as well as Monte Carlo integration of the latent state through the dynamics.

2 Background

State-space models are probabilistic graphical models where observations \mathbf{y}_t in discrete time are conditionally independent given a continuous latent state, \mathbf{z}_t , evolving according to Markovian dynamics, so that the complete data likelihood for T consecutive observations factorizes as,

$$p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T}) = p_{\boldsymbol{\theta}}(\mathbf{z}_1) p_{\boldsymbol{\psi}}(\mathbf{y}_1 | \mathbf{z}_1) \times \prod_{t=2}^T p_{\boldsymbol{\psi}}(\mathbf{y}_t | \mathbf{z}_t) p_{\boldsymbol{\theta}}(\mathbf{z}_t | \mathbf{z}_{t-1})$$

where $\mathbf{z}_t \in \mathbb{R}^L$ are real-valued latent states, $\boldsymbol{\theta}$ parameterizes the dynamics and initial condition, and $\boldsymbol{\psi}$ parameterizes the observation model. When the generative model, $(\boldsymbol{\theta}, \boldsymbol{\psi})$, is known, the statistical inference problem is to compute the smoothing posterior, $p(\mathbf{z}_{1:T} | \mathbf{y}_{1:T})^2$. Otherwise, $(\boldsymbol{\theta}, \boldsymbol{\psi})$ have to be learned from the data – known as system identification⁸.

Variational inference makes it possible to accomplish these goals in a harmonious way. The variational expectation maximization (vEM) algorithm iterates two steps: first, we maximize a lower bound to the log-marginal likelihood – the ELBO – with respect to the parameters of an approximate posterior, $q(\mathbf{z}_{1:T}) \approx p(\mathbf{z}_{1:T} | \mathbf{y}_{1:T})$; then, with the approximate posterior fixed, the ELBO is maximized with respect to parameters of the generative model⁹. For large scale problems, vEM can be slow due to the need to fully optimize the variational parameters before taking gradient steps on parameters of the generative model. Therefore, the variational autoencoder (VAE) is better suited for large scale problems for its ability to simultaneously learn the generative model and inference network – an expressive parametric function that maps data to the parameters of approximate posterior^{10,11}.

Model specifications. Although our approach is applicable to any exponential family state-space process, given their ubiquity, we focus on dynamical systems driven by Gaussian noise so that,

$$p_{\boldsymbol{\theta}}(\mathbf{z}_t | \mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t | \mathbf{m}_{\boldsymbol{\theta}}(\mathbf{z}_{t-1}), \mathbf{Q}_{\boldsymbol{\theta}}) \quad (1)$$

where $\mathbf{m}_{\boldsymbol{\theta}} : \mathbb{R}^L \rightarrow \mathbb{R}^L$ might be a nonlinear neural network function with learnable parameters $\boldsymbol{\theta}$, and $\mathbf{Q}_{\boldsymbol{\theta}} \in \mathbb{R}^{L \times L}$ is a learnable state-noise covariance matrix. Given the favorable properties of exponential family distributions¹²⁻¹⁵, especially in the context of variational inference, we write the prior dynamics in their exponential family representation (natural parameter form),

$$p_{\boldsymbol{\theta}}(\mathbf{z}_t | \mathbf{z}_{t-1}) = h(\mathbf{z}_t) \exp(\mathcal{T}(\mathbf{z}_t)^\top \boldsymbol{\lambda}_{\boldsymbol{\theta}}(\mathbf{z}_{t-1}) - A(\boldsymbol{\lambda}_{\boldsymbol{\theta}}(\mathbf{z}_{t-1}))) \quad (2)$$

where h is the base measure, $\mathcal{T}(\mathbf{z}_t)$ the sufficient statistics, $A(\cdot)$ the log-partition function, and $\boldsymbol{\lambda}_{\boldsymbol{\theta}}(\cdot)$ is a map $\mathbb{R}^L \mapsto \mathbb{R}^{L^2+L}$ that transforms \mathbf{z}_{t-1} to natural parameters for \mathbf{z}_t . For a Gaussian distribution, the sufficient statistics can be defined as $\mathcal{T}(\mathbf{z}_t)^\top = [\mathbf{z}_t^\top \quad -\frac{1}{2}\mathbf{z}_t\mathbf{z}_t^\top]$, so that $\boldsymbol{\lambda}_{\boldsymbol{\theta}}(\cdot)$ for (1) is given by,

$$\boldsymbol{\lambda}_{\boldsymbol{\theta}}(\mathbf{z}_{t-1}) = \begin{bmatrix} \mathbf{Q}_{\boldsymbol{\theta}}^{-1} \mathbf{m}_{\boldsymbol{\theta}}(\mathbf{z}_{t-1}) \\ \mathbf{Q}_{\boldsymbol{\theta}}^{-1} \end{bmatrix} \quad (\text{dynamics model in natural parameter form}) \quad (3)$$

As it will simplify subsequent analysis, the *mean parameter* mapping corresponding to this natural parameter mapping (guaranteed to exist as long as the exponential family is minimal¹²) is given by

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}_{t-1}) = \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{z}_t | \mathbf{z}_{t-1})} [\mathcal{T}(\mathbf{z}_t)] = \begin{bmatrix} \mathbf{m}_{\boldsymbol{\theta}}(\mathbf{z}_{t-1}) \\ -\frac{1}{2} (\mathbf{m}_{\boldsymbol{\theta}}(\mathbf{z}_{t-1}) \mathbf{m}_{\boldsymbol{\theta}}(\mathbf{z}_{t-1})^\top + \mathbf{Q}_{\boldsymbol{\theta}}) \end{bmatrix} \quad \begin{pmatrix} \text{mean} \\ \text{parameter} \\ \text{form} \end{pmatrix} \quad (4)$$

Furthermore, we make the following assumptions **i)** the state-noise, $\mathbf{Q}_{\boldsymbol{\theta}}$, is diagonal or structured for efficient matrix-vector multiplications. **ii)** $\mathbf{m}_{\boldsymbol{\theta}}(\cdot)$, is a nonlinear smooth function. **iii)** the likelihood, $p_{\boldsymbol{\psi}}(\mathbf{y}_t | \mathbf{z}_t)$, may be non-conjugate. **iv)** L may be large enough so that L^3 is comparable to T .

Amortized inference for state-space models. A useful property of SSMs is that, \mathbf{z}_t conditioned on \mathbf{z}_{t-1} and $\mathbf{y}_{1:T}$, is independent of $\mathbf{y}_{1:t-1}$, i.e., $p(\mathbf{z}_t | \mathbf{z}_{t-1}, \mathbf{y}_{1:T}) = p(\mathbf{z}_t | \mathbf{z}_{t-1}, \mathbf{y}_{t:T})$ ^{5,16}. It thus suffices to construct an approximate posterior that factorizes forward in time,

$$q(\mathbf{z}_{1:T}) = q(\mathbf{z}_1) \prod q(\mathbf{z}_t | \mathbf{z}_{t-1}) \quad (5)$$

and introduce learnable function approximators to amortize inference by mapping \mathbf{z}_{t-1} and $\mathbf{y}_{t:T}$ to the parameters of $q(\mathbf{z}_t | \mathbf{z}_{t-1})$. This makes it simple to sample $\mathbf{z}_{1:T}$ from the approximate posterior (using the reparameterization trick) and evaluate the ELBO (a.k.a. negative variational free energy),

$$\mathcal{L}(q) = \sum \mathbb{E}_{q_t} [\log p(\mathbf{y}_t | \mathbf{z}_t)] - \mathbb{E}_{q_{t-1}} [\mathbb{D}_{\text{KL}}(q(\mathbf{z}_t | \mathbf{z}_{t-1}) || p_{\theta}(\mathbf{z}_t | \mathbf{z}_{t-1}))] \leq \log p(\mathbf{y}_{1:T}) \quad (6)$$

where $\mathbb{D}_{\text{KL}}(\cdot || \cdot)$ is the Kullback-Leibler (KL) divergence and $\mathbb{E}_{q_t} \equiv \mathbb{E}_{\mathbf{z}_t \sim q(\mathbf{z}_t; \mathbf{y}_{1:T})}$, so that the generative model and inference network parameters can be learned through stochastic backpropagation. Many works for Gaussian $q(\mathbf{z}_t | \mathbf{z}_{t-1})$, such as Krishnan et al.⁶, Alaa and van der Schaar¹⁷, Girin et al.¹⁸, Hafner et al.¹⁹, construct inference networks that parameterize the variational posterior as

$$q(\mathbf{z}_t | \mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{m}_{\phi}(\mathbf{z}_{t-1}, \mathbf{y}_{1:T}), \mathbf{P}_{\phi}(\mathbf{z}_{t-1}, \mathbf{y}_{1:T})). \quad (7)$$

There are limitless ways to construct $\mathbf{m}_{\phi}(\cdot)$ and $\mathbf{P}_{\phi}(\cdot)$ so ϕ can be learned through gradient ascent on the ELBO, but a straightforward and illustrative approach^{6,17} is to transform future data, $\mathbf{y}_{t:T}$, using a recurrent neural network (RNN), or any efficient autoregressive sequence to sequence model, and then mapping the preceding latent state, \mathbf{z}_{t-1} , using a feed-forward neural network, so that a complete inference network description could be,

$$(\mathbf{m}_{\phi}(\mathbf{z}_{t-1}, \mathbf{y}_{t:T}), \mathbf{P}_{\phi}(\mathbf{z}_{t-1}, \mathbf{y}_{t:T})) = \text{NN}([\mathbf{z}_{t-1}, \mathbf{u}_t]), \quad \mathbf{u}_t = \text{S2S}([\mathbf{u}_{t+1}, \mathbf{y}_t]) \quad (8)$$

where $\text{S2S}(\cdot)$ is a parametric sequence-to-sequence function that maintains a hidden state \mathbf{u}_t and takes as input \mathbf{y}_t , and $\text{NN}(\cdot)$ is a parametric function designed to output approximate posterior parameters. This leads to a backward-forward algorithm, meaning that data $\mathbf{y}_{1:T}$ are mapped to $\mathbf{u}_{1:T}$ in reverse time, and then samples are drawn from $q(\mathbf{z}_t | \mathbf{z}_{t-1})$ forward in time.

Possible drawbacks of this inference framework are **i)** missing observations obstruct inference (the example networks cannot naturally accommodate missing data); **ii)** sampling entire trajectories to approximate the expected KL term can potentially lead to high-variance gradient estimators, and **iii)** statistics of the marginals (e.g. second moments) can only be approximated through sample averages.

3 Related works

Many existing works also explore inference and data-driven learning for state-space graphical models within the VAE framework. We highlight the most closely related studies and note specific limitations that our work seeks to address. The structured variational autoencoder (SVAE)²⁰ makes it possible to efficiently evaluate the ELBO while preserving the temporal structure of the posterior by restricting the prior to a *linear dynamical system* (LDS) and then constructing the approximation as $q(\mathbf{z}_{1:T}) \propto p_{\theta}(\mathbf{z}_{1:T}) \prod \exp(t(\mathbf{z}_t)^{\top} \psi(\mathbf{y}_t))$ so that its statistics can be obtained using efficient message passing algorithms. However, the SVAE is not directly applicable when the dynamics are nonlinear since the joint prior will no longer be Gaussian (thereby not allowing for efficient conjugate updates). Recently, Zhao and Linderman²¹ expanded on the SVAE framework by exploiting the LDS structure and associative scan operations to improve its scalability.

The deep Kalman filter (DKF)⁶ uses black-box inference networks to make drawing joint samples from the full posterior simple. However, pure black-box amortization networks such as those can make learning the parameters of the generative model dynamics difficult because their gradients will not propagate through the expected log-likelihood term⁵. In contrast, we consider inference networks inspired by the fundamental importance of the prior for evaluating Bayesian conjugate updates. The deep variational Bayes filter (DVBF) also considers inference and learning in state-space graphical models⁵. Difficulties of learning the generative model that arise as a result of more standard VAE implementations defining inference networks independent of the prior are handled by forcing samples from the approximate posterior to traverse through the dynamics. Our work extends this concept, by directly specifying the parameters of the variational approximation in terms of the prior.

Our approach constructs an inference network infused with the prior similar to the SVAE and DVBF but **i)** avoids restrictions to LDS and **ii)** affords easy access to approximations of the marginal statistics (such as the dense latent state covariances) without having to average over sampled trajectories (or store them directly which would be prohibitive as the latent dimensionality becomes large).

4 Method

An alternative to constructing variational approximations through specification of conditional distributions, as in Eq. (7), involves the use of data-dependent Gaussian potentials, that we refer to as pseudo-observations:

$$p(\tilde{\mathbf{y}}_t | \mathbf{z}_t) \propto \exp(\tilde{\boldsymbol{\lambda}}_\phi(\mathbf{y}_{1:T})^\top \mathcal{T}(\mathbf{z}_t)) \equiv \exp(\tilde{\boldsymbol{\lambda}}_t^\top \mathcal{T}(\mathbf{z}_t)) = \exp(\mathbf{k}_t^\top \mathbf{z}_t - \frac{1}{2} \|\mathbf{K}_t \mathbf{z}_t\|^2) \quad (9)$$

These Gaussian potentials can then be combined with the prior through Bayes' rule, yielding the approximation

$$q(\mathbf{z}_{1:T}) = \frac{\prod p(\tilde{\mathbf{y}}_t | \mathbf{z}_t) p_\theta(\mathbf{z}_{1:T})}{p(\tilde{\mathbf{y}}_{1:T})} \quad (10)$$

A benefit of this formulation, is that it inherently imposes the latent dependency structure of the generative model onto the amortized posterior. This parameterization, was introduced in Johnson et al.²⁰, where an important point highlighted, is that the Gaussian potentials can encode any arbitrary subset of observations; for example, $p(\tilde{\mathbf{y}}_t | \mathbf{z}_t)$ could be made to depend on \mathbf{y}_t alone, or even the entire dataset, $\mathbf{y}_{1:T}$. Regardless of that particular design choice, the corresponding ELBO for the variational approximation of Eq. (10) is

$$\mathcal{L}(q) = \sum \mathbb{E}_{q_t} [\log p(\mathbf{y}_t | \mathbf{z}_t)] - \mathbb{E}_{q_t} [\log p(\tilde{\mathbf{y}}_t | \mathbf{z}_t)] + \log p(\tilde{\mathbf{y}}_{1:T}) \quad (11)$$

For linear Gaussian latent dynamics, conjugate potentials could be efficiently integrated with the prior using exact message passing, yielding filtered and smoothed marginal statistics. The smoothed statistics can be used to evaluate the first two terms on the right-hand side, while the filtered statistics can be used to evaluate the final term, the log-marginal likelihood of the pseudo-observations.

However, this approach does not directly apply to nonlinear dynamical systems, where the variational posterior is no longer Gaussian. Since evaluating the smoothed marginals and the log-marginal likelihood of pseudo-observations relies on first obtaining the filtered marginals, a logical starting point is to develop a method for approximating these filtered marginals. To this end, we propose a differentiable approximate message passing algorithm specifically designed to compute filtered posterior statistics in models characterized by nonlinear latent dynamics and observations represented as Gaussian potentials. Building on this foundation, we then return to the subsequent challenges of efficiently computing smoothed posterior statistics and evaluating the ELBO.

Differentiable nonlinear filtering. Bayesian filtering is often conceptualized as a two step procedure². In the *predict* step, our belief of the latent state is integrated through the dynamics, yielding $q(\mathbf{z}_t | \tilde{\mathbf{y}}_{1:t-1}) = \int p_\theta(\mathbf{z}_t | \mathbf{z}_{t-1}) q(\mathbf{z}_{t-1} | \tilde{\mathbf{y}}_{1:t-1}) d\mathbf{z}_{t-1}$ (a.k.a. the predictive distribution). Then, applying Bayes' rule, $q(\mathbf{z}_t | \tilde{\mathbf{y}}_{1:t}) \propto p(\tilde{\mathbf{y}}_t | \mathbf{z}_t) q(\mathbf{z}_t | \tilde{\mathbf{y}}_{1:t-1})$, we *update* our belief. Evidently then, developing an approximate filtering algorithm that exploits the conjugacy of the pseudo observations can be recast as the problem: given an approximation $\pi(\mathbf{z}_{t-1}) \approx q(\mathbf{z}_{t-1} | \tilde{\mathbf{y}}_{1:t-1})$, find an approximation to the predictive distribution, $\bar{\pi}(\mathbf{z}_t) \approx q(\mathbf{z}_t | \tilde{\mathbf{y}}_{1:t-1})$. The recursion would continue forward by updating our belief analytically, setting $\pi(\mathbf{z}_t) \propto p(\tilde{\mathbf{y}}_t | \mathbf{z}_t) \bar{\pi}(\mathbf{z}_t)$, then finding a Gaussian approximation of $\bar{\pi}(\mathbf{z}_{t+1})$ and so forth.

With the problem restated this way, we propose an approximate filtering solution designed to exploit two key factors at play **i**) the approximate beliefs are constrained to the same exponential family as the latent state transitions **ii**) the pseudo observations are encoded as conjugate potentials. Our approach involves recursively solving intermediary variational problems (their fixed point solutions on the right),

— Variational filtering —

$$(i) \bar{\pi}(\mathbf{z}_t) = \operatorname{argmin} \mathbb{D}_{\text{KL}}(\mathbb{E}_{\pi_{t-1}} [p_\theta(\mathbf{z}_t | \mathbf{z}_{t-1})] || \bar{\pi}(\mathbf{z}_t)) \implies \bar{\boldsymbol{\mu}}_t = \mathbb{E}_{\pi_{t-1}} [\boldsymbol{\mu}_\theta(\mathbf{z}_{t-1})] \quad (12)$$

$$(ii) \pi(\mathbf{z}_t) = \operatorname{argmin} \mathbb{D}_{\text{KL}}(\pi(\mathbf{z}_t) || p(\tilde{\mathbf{y}}_t | \mathbf{z}_t) \bar{\pi}(\mathbf{z}_t)) \implies \boldsymbol{\lambda}_t = \bar{\boldsymbol{\lambda}}_t + \tilde{\boldsymbol{\lambda}}_t \quad (13)$$

Steps (i) and (ii) can be thought of as variational analogues of the predict/update steps of Bayesian filtering, and importantly, finding their fixed point solutions does not require an iterative procedure because of our problem specifications. Reassuringly, iterating (i) and (ii) in the case of an LDS generative model would exactly recover the information form Kalman filtering equations. In the case of nonlinear dynamical systems, directly taking the expectation in (i) is intractable. We can overcome

this by employing the reparameterization trick to obtain a differentiable sample approximation of the parameters, $\tilde{\boldsymbol{\mu}}_t$, of the fixed point solution. Naturally now, the statistics of the approximate filtered beliefs can be used to approximate the log-marginal likelihood of the pseudo observations as,

$$\log p(\tilde{\mathbf{y}}_{1:T}) = \sum \log \int p(\tilde{\mathbf{y}}_t | \mathbf{z}_t) q(\mathbf{z}_t | \tilde{\mathbf{y}}_{1:t-1}) d\mathbf{z}_t \approx \sum \log \int p(\tilde{\mathbf{y}}_t | \mathbf{z}_t) \bar{\pi}(\mathbf{z}_t) d\mathbf{z}_t \quad (14)$$

where the last integral can be evaluated analytically as a result of the Gaussian forms of the approximations and pseudo observations. While formulating step (ii) as a variational problem may appear superfluous given the conjugate structure, it hints at the possibility of approximating smoothed posterior marginal statistics within a variational framework. However, pursuing this idea further reveals significant challenges. Trying to develop a backward recursion for the distribution q_t minimizing $\mathbb{D}_{\text{KL}}(\mathbb{E}_{q_{t+1}}[q_{t|t+1}] || q_t)$, by using the forward KL divergence (as in step (i)), leads to an intractable problem because the backward Markov transitions, $q_{t|t+1}$, are not conditionally Gaussian. Conversely, a fixed point solution of the reverse KL objective (as in step(ii)), $\mathbb{D}_{\text{KL}}(q_t || \mathbb{E}_{q_{t+1}}[q_{t|t+1}])$, necessitates an iterative procedure, which can be computationally expensive.

Smoothing as filtering. In light of these difficulties, we offer a simple solution that exploits the flexibility in choosing the pseudo observation data dependence: define the parameters, \mathbf{k}_t and \mathbf{K}_t , of each pseudo observation, $\tilde{\mathbf{y}}_t$, to be a function of current *and* future data, $\mathbf{y}_{t:T}$, so that,

$$p(\tilde{\mathbf{y}}_t | \mathbf{z}_t) \propto \exp(\tilde{\boldsymbol{\lambda}}_\phi(\mathbf{y}_{t:T})^\top \mathcal{T}(\mathbf{z}_t)) \quad (15)$$

With this choice, filtered statistics of the latent state—relative to the pseudo-observations—can be used to approximate posterior smoothed marginals, i.e. $\pi(\mathbf{z}_t) \approx q(\mathbf{z}_t | \tilde{\mathbf{y}}_{1:t}) \approx p(\mathbf{z}_t | \mathbf{y}_{1:T})$. This solution circumvents the challenges associated with backward message computation and only requires a single pass through the pseudo observations to obtain approximate smoothed posterior statistics. Substituting, π_t , as an approximation to q_t , in Eq. (11), leads to the following approximation of the ELBO.

Variational smoothing ELBO

$$\hat{\mathcal{L}}(\pi) = \sum \mathbb{E}_{\pi_t} [\log p(\mathbf{y}_t | \mathbf{z}_t)] - \mathbb{E}_{\pi_t} [\log p(\tilde{\mathbf{y}}_t | \mathbf{z}_t)] + \log \mathbb{E}_{\pi_t} [p(\tilde{\mathbf{y}}_t | \mathbf{z}_t)] \quad (16)$$

$$= \sum \mathbb{E}_{\pi_t} [\log p(\mathbf{y}_t | \mathbf{z}_t)] - \mathbb{D}_{\text{KL}}(\pi(\mathbf{z}_t) || \bar{\pi}(\mathbf{z}_t)) \quad (17)$$

By expressing the approximate ELBO compactly as Eq.(17), we highlight that it promotes learning models where the posterior at time t aligns closely with the one-step posterior predictive at that time (which depends on the generative model and the posterior at time $t - 1$). The KL term in Eq.(17) can be evaluated in closed form, while the expected log-likelihood term can be approximated using the reparameterization trick⁴. However, a point of practical importance should be raised now: every filtering step and evaluation of the KL term has a time complexity of $\mathcal{O}(L^3)$, which may become a bottleneck for large L . In the following discussion, we will explore effective strategies to parameterize the Gaussian potential inference network that produces $\tilde{\boldsymbol{\lambda}}_{1:T}$, to reduce the computational burden that filtering and evaluating $\hat{\mathcal{L}}(\pi)$ pose in the large L regime.

Local and backward encoders. For state-space models, inferences about the latent state should be possible even with missing observations. To enable the amortized inference network to process missing observations in a principled way, we decompose the natural parameter update into two additive components: **i**) a *local* encoder, $\boldsymbol{\alpha}_\phi(\cdot)$, for current observation, and **ii**) a *backward* encoder, $\boldsymbol{\beta}_\phi(\cdot)$, for future observations, i.e.,

$$\tilde{\boldsymbol{\lambda}}_\phi(\mathbf{y}_{t:T}) = \boldsymbol{\alpha}_\phi(\mathbf{y}_t) + \boldsymbol{\beta}_\phi(\mathbf{y}_{t+1:T}) \quad (\text{or for the sake of brevity}) \quad \tilde{\boldsymbol{\lambda}}_t = \boldsymbol{\alpha}_t + \boldsymbol{\beta}_{t+1} \quad (18)$$

Furthermore, by building the dependence of $\boldsymbol{\beta}_\phi(\cdot)$ on $\mathbf{y}_{t+1:T}$ through their representation as $\boldsymbol{\alpha}_{t+1:T}$, so that $\boldsymbol{\beta}_\phi(\mathbf{y}_{t+1:T}) = \boldsymbol{\beta}_\phi(\boldsymbol{\alpha}_{t+1:T})$, a missing observation at time t is handled by setting $\boldsymbol{\alpha}_t = \mathbf{0}$. While a data dependent natural parameter update of $\mathbf{0}$ faithfully represents a missing observation – in the absence of data, the prior should not be updated – alternatively setting $\mathbf{y}_t = \mathbf{0}$ would introduce a harmful inductive bias into the inference network, since an observation of $\mathbf{0}$ can be arbitrarily informative. Given the impracticality of $\mathcal{O}(TL^2)$ memory requirements, it is appealing to consider a low-rank parameterization for the local and backward encoders – we consider

$$\boldsymbol{\alpha}_t = \begin{pmatrix} \mathbf{a}_t \\ \mathbf{A}_t \mathbf{A}_t^\top \end{pmatrix} := \begin{pmatrix} \mathbf{a}(\mathbf{y}_t) \\ \mathbf{A}(\mathbf{y}_t) \mathbf{A}(\mathbf{y}_t)^\top \end{pmatrix} \quad \boldsymbol{\beta}_t = \begin{pmatrix} \mathbf{b}_t \\ \mathbf{B}_t \mathbf{B}_t^\top \end{pmatrix} := \begin{pmatrix} \mathbf{b}(\boldsymbol{\alpha}_{t:T}) \\ \mathbf{B}(\boldsymbol{\alpha}_{t:T}) \mathbf{B}(\boldsymbol{\alpha}_{t:T})^\top \end{pmatrix} \quad (19)$$

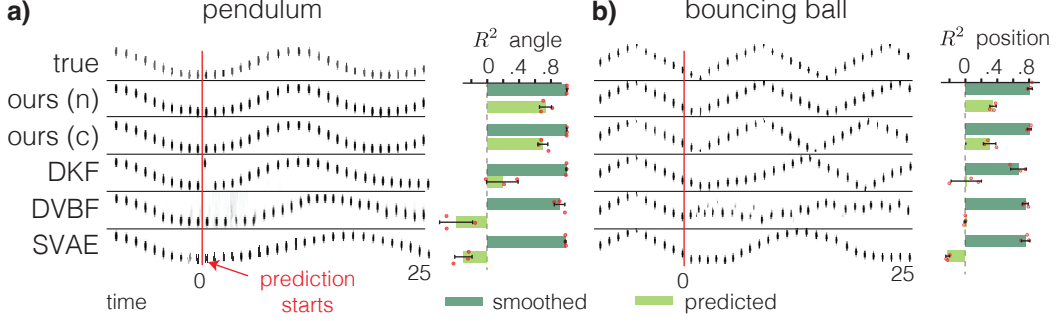


Figure 1: **Smoothing and predictive performance on bouncing ball and pendulum.** To the left of the red line are samples from the posterior during the data window projected to image space, to the right of the red line are samples unrolled from $p_{\theta}(\mathbf{z}_t | \mathbf{z}_{t-1})$. **a)** while all methods are adept at smoothing in the context window, our methods predictive performance is better by a noticeable margin as measured by the R^2 . **b)** similar results hold for the bouncing ball dataset.

where $\mathbf{A}_t \in \mathbb{R}^{L \times r_{\alpha}}$ with $\mathbf{B}_t \in \mathbb{R}^{L \times r_{\beta}}$ parameterize low-rank local/backward precision updates, and $\mathbf{a}_t \in \mathbb{R}^L$ with $\mathbf{b}_t \in \mathbb{R}^L$ parameterize local/backward precision-scaled mean updates. Using these descriptions and the additive decomposition (18), the parameters of a single pseudo observation are,

$$\tilde{\lambda}_t = \begin{pmatrix} \mathbf{k}_t \\ \mathbf{K}_t \mathbf{K}_t^{\top} \end{pmatrix} := \begin{pmatrix} \mathbf{k}(\mathbf{y}_{t:T}) \\ \mathbf{K}(\mathbf{y}_{t:T}) \mathbf{K}(\mathbf{y}_{t:T})^{\top} \end{pmatrix} = \begin{pmatrix} \mathbf{a}_t + \mathbf{b}_t \\ [\mathbf{A}_t \ \mathbf{B}_t] [\mathbf{A}_t \ \mathbf{B}_t]^{\top} \end{pmatrix} \quad (20)$$

where $\mathbf{K} \in \mathbb{R}^{L \times r}$ if $r = r_{\alpha} + r_{\beta}$ and $\mathbf{k} \in \mathbb{R}^L$. The low-rank structure of the natural parameter updates will be a key component to develop an efficient approximate message passing algorithm for obtaining sufficient statistics of the approximate posterior and evaluating the ELBO. Analogous to the inference network description (8), a differentiable architecture producing $\alpha_{1:T}$ and $\beta_{1:T}$ could be,

$$\alpha_t = \text{NN}(\mathbf{y}_t) \quad \beta_t = \text{S2S}([\beta_{t+1}, \alpha_t]), \quad (21)$$

which overall defines the map $\mathbf{y}_{1:T} \mapsto (\alpha_{1:T}, \beta_{1:T})$. In addition, the separation of local and backward encoders can reduce the complexity of the backward encoder for $L < N$. Those familiar with sequential Monte-Carlo (SMC) methods²² can view the backward encoder similar to twisting functions used to combine future information with filtered state beliefs to produce smoothing approximations^{23,24}.

Exploiting structure for efficient filtering. A benefit of using the forward KL to design a variational analogue to the exact Bayesian predict step is immediate access to the fixed point solution. While nonlinear specification of the latent dynamical system make the expectation of Eq. (12) intractable, using the reparameterization trick with $\mathbf{z}_{t-1}^s \sim \pi(\mathbf{z}_{t-1})$, gives the approximation,

$$\bar{\mu}_t = 1/S \sum_{s=1}^S \left[-\frac{1}{2} (\mathbf{m}_{\theta}(\mathbf{z}_{t-1}^s) \mathbf{m}_{\theta}(\mathbf{z}_{t-1}^s)^{\top} + \mathbf{Q}_{\theta}) \right] \quad (22)$$

where S is the total number of samples. Converting this finite sample estimate from mean parameter coordinates to a mean/covariance representation, we get that,

$$\bar{\mathbf{m}}_t = 1/S \sum_{s=1}^S \mathbf{m}_{\theta}(\mathbf{z}_{t-1}^s) \quad \bar{\mathbf{P}}_t = \bar{\mathbf{M}}_t^c \bar{\mathbf{M}}_t^{c\top} + \mathbf{Q}_{\theta} \quad (23)$$

where $\bar{\mathbf{M}}_t^c$ is the $L \times S$ matrix of samples passed through the dynamics function, then centered by the mean, defined for convenience as,

$$\bar{\mathbf{M}}_t^c = 1/\sqrt{S} [\mathbf{m}_{\theta}(\mathbf{z}_{t-1}^1) - \bar{\mathbf{m}}_t, \dots, \mathbf{m}_{\theta}(\mathbf{z}_{t-1}^S) - \bar{\mathbf{m}}_t] \in \mathbb{R}^{L \times S} \quad (24)$$

Writing the covariance estimate as it is in Eq. (23), reveals that it can alternatively be represented by the pair $(\bar{\mathbf{M}}_t^c, \mathbf{Q}_{\theta})$. In the regime where $L > S$, significant computational savings can be afforded by capitalizing on the low-rank structure of the covariance as estimated via the reparameterization trick. This structure can be exploited for efficient linear algebraic operations involving $\bar{\mathbf{P}}_t$ (and its inverse, after application of the Woodbury identity). Consequently, the natural parameters of π_t , after updating $\bar{\pi}_t$, are

$$\mathbf{P}_t^{-1} \mathbf{m}_t = \bar{\mathbf{P}}_t^{-1} \bar{\mathbf{m}}_t + \mathbf{k}_t \quad \mathbf{P}_t^{-1} = \bar{\mathbf{P}}_t^{-1} + \mathbf{K}_t \mathbf{K}_t^{\top} \quad (25)$$

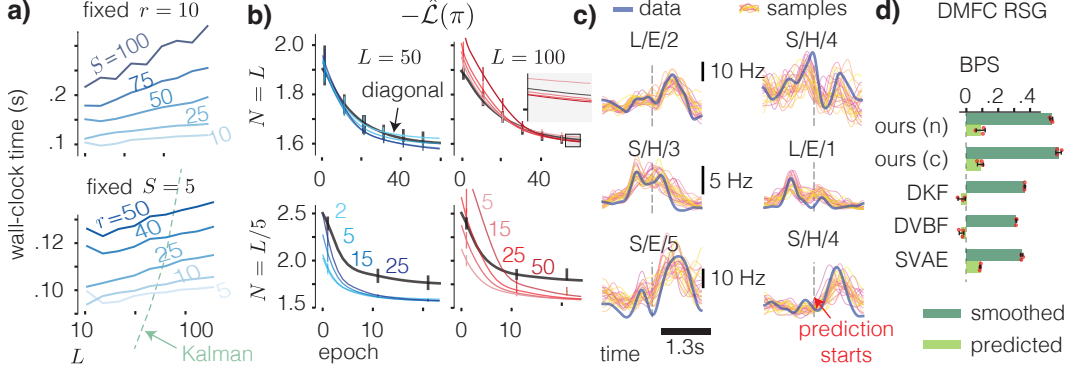


Figure 2: **a)** Empirical time complexity scaling. Since complexity is a function of L , S , and r , we vary L (top) for fixed $r = 10$ and (bottom) for fixed $S = 5$; we examine several values of the variable not fixed. Examining wall-clock time shows empirically our implementation scales linearly in L ; on the (bottom) we plot wall-clock time for a Kalman filter implementation, showing the standard cubic dependence on L . **b)** (top) Negative ELBO as a function of training epoch when $N = L$ (bottom) when $N = L/5$; the left column shows the case $L = 50$ and the right when $L = 100$. Different colors indicate different settings of the local/backward encoder rank; zooming in for $L = 100$, shows low-rank updates can match diagonal ones. **c)** Peristimulus time histogram (PSTH) for the DMFC RSG dataset for different trial condition averages; we consider a context window of 1.3s and a prediction window of 1.3s. **d)** BPS for each method for context/prediction windows.

and also admit structured representations. Without *both* the sample approximation structure (during the variational predict step) and low-rank parameterization (during the variational update step), the cost of approximate filtering each time-step would be dominated by an $\mathcal{O}(L^3)$ cost. Instead, recognizing the potential computational advantages of exploiting these structures, and never instantiating the predictive/updated covariance and precision matrices, makes it possible to develop an approximate filtering algorithm, where in the case L is significantly larger than S or r , has complexity of $\mathcal{O}(L(Sr + r^2 + S^2))$ per step. More details regarding time complexity are in App. B.5.

Efficient sampling and ELBO evaluation. When $\mathbb{E}_{\pi_t} [\log p(\mathbf{y}_t | \mathbf{z}_t)]$ can not be evaluated in closed form, Monte-Carlo integration can be used as a differentiable approximation. To sample from $\pi(\mathbf{z}_t)$ without explicitly constructing \mathbf{P}_t , we can take $\bar{\mathbf{z}}_t^s \sim \mathcal{N}(\mathbf{0}, \bar{\mathbf{P}}_t)$ and $\mathbf{w}_t^s \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{L+S})$ and set,

$$\mathbf{z}_t^s = \mathbf{m}_t + \bar{\mathbf{z}}_t^s - \mathbf{K}_t \Upsilon_t \Upsilon_t^\top (\mathbf{K}_t^\top \bar{\mathbf{z}}_t^s + \mathbf{w}_t^s). \quad (26)$$

While more details are provided in App. B.4, this can be done efficiently since samples can be drawn cheaply from $\bar{\pi}(\mathbf{z}_t)$ using Eq. (23). Whereas Monte-Carlo approximations of the expected log-likelihood term might be unavoidable, the closed form solution for the KL between two Gaussian distributions should be used to avoid further stochastic approximations. The only difficulty, is that the time complexity of naively evaluating the KL term,

$$\mathbb{D}_{\text{KL}}(\pi(\mathbf{z}_t) || \bar{\pi}(\mathbf{z}_t)) = \frac{1}{2} [(\bar{\mathbf{m}}_t - \mathbf{m}_t)^\top \bar{\mathbf{P}}_t^{-1} (\bar{\mathbf{m}}_t - \mathbf{m}_t) + \text{tr}(\bar{\mathbf{P}}_t^{-1} \mathbf{P}_t) + \log(|\bar{\mathbf{P}}_t|/|\mathbf{P}_t|) - L] \quad (27)$$

scales $\mathcal{O}(L^3)$. However, since matrix vector multiplies with $\bar{\mathbf{P}}_t^{-1}$ can be performed efficiently and the trace/log-determinant terms can be rewritten using the square-root factors acquired during the forward pass, as we describe in App. C.1, it is possible to evaluate the KL in $\mathcal{O}(LSr + LS^2 + Lr^2)$ time. After a complete forward pass through the encoded data, we acquire the samples $\mathbf{z}_{1:T}^s$ and all necessary quantities for efficient ELBO evaluation. We detail the variational filtering algorithm in Alg. 2 in App. C.2 and the complete end-to-end learning procedure in Alg. 1.

Causal amortized inference for streaming data. In constructing a fully differentiable variational approximation, the parameters of the approximate marginals were effectively amortized according to a recursion in the natural parameter space by iterating Eqs. (12) and (13). This recursion can be recognized more easily by introducing the function, $\mathfrak{F}_\theta(\cdot)$, and writing

$$\lambda_t = \mathfrak{F}_\theta(\lambda_{t-1}) + \alpha_t + \beta_{t+1} \text{ with } \mathfrak{F}_\theta(\lambda_{t-1}) = \nabla A^* \left(\int \pi(\mathbf{z}_{t-1}; \lambda_{t-1}) \boldsymbol{\mu}_\theta(\mathbf{z}_{t-1}) d\mathbf{z}_{t-1} \right) \quad (28)$$

Algorithm 1 End-to-end learning

Input: $\mathbf{y}_{1:T}$
while not converged **do**
 for $t = T$ **to** 1 **do**
 $\boldsymbol{\alpha}_t = \text{NN}(\mathbf{y}_t)$ # local encoder
 $\boldsymbol{\beta}_t = \text{S2S}([\boldsymbol{\beta}_{t+1} \ \boldsymbol{\alpha}_t])$ # backward encoder
 $\mathbf{k}_t = \mathbf{a}_t + \mathbf{b}_t$
 $\mathbf{K}_t = [\mathbf{A}_t \ \mathbf{B}_t]$
 end for
 $\mathbf{z}_{1:T}^{1:S}, \mathbf{m}_{1:T}, \bar{\mathbf{m}}_{1:T}, \boldsymbol{\Upsilon}_{1:T} = \text{Alg. 2}(\mathbf{k}_{1:T}, \mathbf{K}_{1:T})$
 $\hat{\mathcal{L}}(\pi) = \sum [S^{-1} \sum \log p(\mathbf{y}_t | \mathbf{z}_t^s) - \mathbb{D}_{\text{KL}}(\pi_t || \bar{\pi}_t)]$
 $(\phi, \boldsymbol{\theta}, \boldsymbol{\psi}) \leftarrow (\phi, \boldsymbol{\theta}, \boldsymbol{\psi}) - \nabla \hat{\mathcal{L}}(\pi)$
end while
Output: $\mathbf{z}_{1:T}^{1:S}, \mathbf{m}_{1:T}, \bar{\mathbf{m}}_{1:T}, \boldsymbol{\Upsilon}_{1:T}$

$\nabla A^* : (\mathbf{m}, -\frac{1}{2}(\mathbf{P} + \mathbf{m}\mathbf{m}^\top)) \mapsto (\mathbf{P}^{-1}\mathbf{m}, \mathbf{P}^{-1})$, and $A^*(\cdot)$ is the convex conjugate of the log-partition function¹². So that, $\mathfrak{F}_\theta(\cdot)$ can be thought of as mapping $\boldsymbol{\lambda}_{t-1}$ forward in time by first taking the expectation of (4) with respect to $\pi(\mathbf{z}_{t-1}; \boldsymbol{\lambda}_{t-1})$, and then applying the mean-to-natural coordinate transformation.

One limitation of amortizing inference through the recursion (28) is its inability to produce approximations for the filtering distributions, $p(\mathbf{z}_t | \mathbf{y}_{1:t})$, which can be valuable in streaming or online settings, as well as for testing hypotheses of causality. However, since (17) only depends on the posterior and posterior predictive marginal statistics, we have the freedom to alter our inference network in a way such that filtered marginal statistics are a by-product of obtaining smoothed marginal statistics. For example, an alternative sequence-to-sequence map for $\boldsymbol{\lambda}_t$ could be defined by,

$$\boldsymbol{\lambda}_t = \underbrace{\mathfrak{F}_\theta(\boldsymbol{\lambda}_{t-1} - \boldsymbol{\beta}_t)}_{\text{smoothed-future} = \text{filtered}} + \boldsymbol{\alpha}_t + \boldsymbol{\beta}_{t+1}, \quad (29)$$

so that $\check{\boldsymbol{\lambda}}_t \equiv \boldsymbol{\lambda}_t - \boldsymbol{\beta}_{t+1}$ obey the recursion $\check{\boldsymbol{\lambda}}_t = \mathfrak{F}_\theta(\check{\boldsymbol{\lambda}}_{t-1}) + \boldsymbol{\alpha}_t$ and are natural parameters of an approximate filtering distribution, $\check{\pi}(\mathbf{z}_t) \approx p(\mathbf{z}_t | \mathbf{y}_{1:t})$. Consequently the approximations to posterior and predictive distributions will have a more complicated relationship than they previously did; while efficient sampling and ELBO evaluation are more intricate as a result – linear time scaling in the state-dimension can still be achieved with additional algebraic manipulations, as we show in App. C.2.

5 Experiments

Time complexity & low-rank precision updates. We first investigated the properties of low-rank variational Gaussian approximations in the large L regime. To guide us, we had several questions in mind such as: i) how does the performance of low-rank approximations compare to full-rank and diagonal covariance approximations, ii) how large compared to L should the rank of precision updates be to achieve satisfactory results, and iii) how does convergence using low-rank approximations compare to diagonal approximations, considering that they require a larger number of parameters. We expect that full-rank approximations would perform best (given a sufficient amount of data), since the true posterior will have dense second-order statistics due to the interactions of latent states in both the dynamics and observation models. However, it remains unclear how many dimensions are necessary for a low-rank approximation to achieve similar performance and whether this number will be practical.

We simulated data from 50D and 100D linear dynamical systems and compared the convergence between dense and diagonal approximations (Fig. 2b); we examined the ELBO for different rank parameterizations in two regimes i) observations and states are of the same dimensionality, $N = L$ (Fig. 2b - top), and ii) observations are lower dimensional than states, $N = L/5$ (Fig. 2b - bottom). While not surprising that dense variational Gaussian approximations achieve superior performance, message passing in latent Gaussian models with dense covariance scales like $\mathcal{O}(L^3)$ ²⁵ and becomes prohibitive for large L ; thus it is reassuring that in both regimes, low-rank approximate posterior parameterizations achieve comparable results for precision matrix updates of relatively low rank

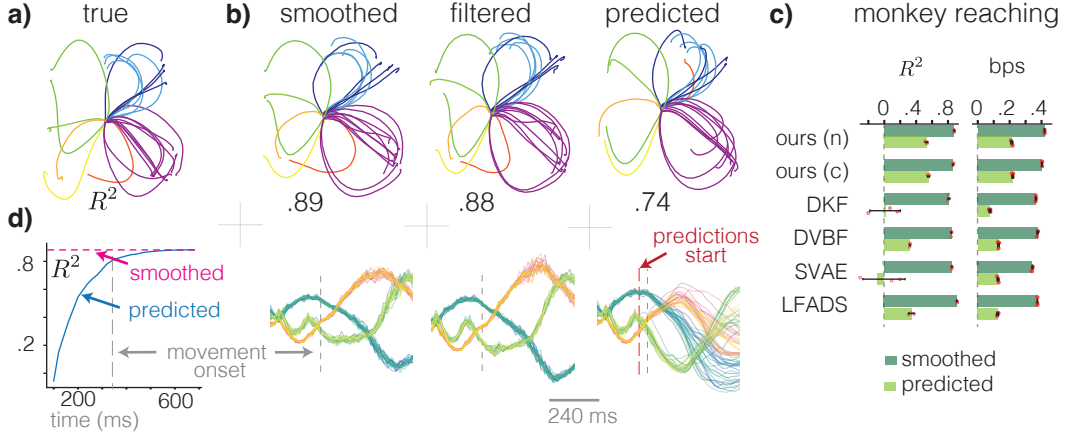


Figure 3: **Predict behavior from a causally inferred initial condition.** **a)** Actual reaches. **b)** (top) Reaches linearly decoded from smoothed ($R^2 = 0.89$), causally filtered ($R^2 = 0.88$), & predicted ($R^2 = 0.74$) latent trajectories starting from an initial condition causally inferred during the preparatory period. (bottom) Top 3 principal latent dimensions per regime (smoothing/filtering/prediction) for three example trials. **c)** bps / R^2 of predicted hand velocity using rates inferred from the 700ms context window and the 500ms prediction window. **d)** Velocity decoding R^2 using predicted trajectories as a function of how far into the trial the latent state was filtered until it was only sampled from the autonomous dynamics; by the the movement onset, behavioral predictions using latent trajectory predictions are nearly on par with behavior decoded from the smoothed posterior.

compared to L . To empirically examine time complexity scaling as a function of L , S , and r , in Fig. 2a, we plot wall-clock times for fixed r while varying S and L (bottom), and for fixed S while varying r and L (top); reassuringly, inference time complexity scales $\mathcal{O}(L)$.

Baseline comparisons – pendulum & bouncing ball. Next, we wondered how our approach fared against other modern deep state-space models when it came to learning complex dynamical systems from data. To explore this, we considered two popular datasets: **i)** a pendulum system²⁶ and **ii)** a bouncing ball^{27,28}. Each dataset consists of sequences of observations that are 16×16 pixel images that are governed by a low-dimensional dynamical system. An interesting aspect of these datasets is that images can be reconstructed with impartial knowledge of the latent state, but for accurate long-term predictions, the dynamics will need to propagate features of the latent state that are irrelevant to the likelihood (e.g. pendulum angular velocity). For benchmarks, three other deep SSM approaches were included: **i)** deep variational Bayes filter (DVBF)⁵ **ii)** deep Kalman filter (DKF)²⁹ **iii)** structured VAE (SVAE)²⁰. We denote our causal amortization network with (c) and the noncausal version with (n).

We trained all models in context windows of 50 consecutive images and then sampled future 50 / 25 time-step latent states from the learned dynamical system for pendulum / bouncing ball. To measure quality of learned latent representation and dynamics, we fit angular velocity / position decoders from training set latent states inferred from pendulum / bouncing ball observations. Then, on held-out test data, we measured the R^2 of velocity / position predictions during the context (smoothing) and forecast (prediction) windows. Fig. 1 shows that all methods are able to reconstruct well in context windows, however, when prediction is concerned, where the underlying dynamics would need to be learned well for accurate forecasts, our method consistently performs better.

Neural population dynamics. We consider two neuroscientific datasets where previous studies have shown the importance of population dynamics in generating plausible hypothesis about underlying neural computation. In addition to DKF, DVBF, and SVAE, we include the LFADS method⁷. First, we considered recordings from motor cortex of a monkey performing a reaching task³⁰ and evaluate each methods' ability to forecast neural spiking and behavioral correlates. We measure the performance by bits-per-spike (BPS) using inferred spike-train rates³¹ and R^2 for decoding hand velocity. Similar to the previous experiment, we evaluate the performance in two regimes: **i)** a 700ms context window and **ii)** a 500ms prediction window following an initial context window of 200ms. Fig. 3c shows that, while all the methods excel at smoothing in the context window, our method makes more informative

predictions in terms of R^2 and BPS. Next, we examined how well the monkey’s behavior could be predicted given only causal estimates of the latent state; we trained a model using the causal amortized inference network, given by Eq. (29), then use learned inference network to infer latent states to predict behavior in three regimes: smoothing, filtering, and prediction. Fig. 3b shows that hand velocity can be decoded nearly as well in the filtering regime (without access to future data) as in the smoothing regime. In Fig. 3d, we plot how the quality of predictions change as filtered latent states are unrolled through the learned dynamics at different points in the trial; showing that forecasts starting prior to movement onset exhibit strong predictive capability.

Next, we investigated our method’s performance with data exhibiting a more intricate trial structure. Specifically, we analyzed physiological recordings from the DMFC region of a monkey engaged in a timing interval reproduction task³². During this task, the monkey observes a random interval of time (termed the ‘ready’-‘set’ period) demarcated by two cues, and the goal of the monkey is to reproduce that interval (termed the ‘set’-‘go’ period). We perform a similar procedure as before, but for this experiment we use the period before ‘set’ as the context window, and use the learned dynamics to make predictions onward; in Fig. 2d we show the BPS measured on test data during the context and forecast windows. To further investigate the predictive capabilities, we examined condition averaged PSTH produced by samples from the latent state posterior during the joint context/prediction windows. Using the trained model, we sample spike valued observations for the context/prediction windows and then computed condition averaged PSTHs; the results shown in Fig. 2c, show that PSTHs sampled from the model remain true to the data, even during the lengthy prediction window.

6 Discussion

We presented a new approximate variational filtering/smoothing algorithm, variational learning objective, and Gaussian inference network parameterization for nonlinear state-space models. Focusing on approximations parameterized by dense covariances forced us to consider strategies that ensured computational feasibility for inference and learning with large L . The introduced approximate variational filtering algorithm, while especially useful for the nonlinear dynamics we considered, could also be applied to high-dimensional linear systems where exact computations might be infeasible for large L . Although our variational objective loses the property of lower-bounding the original data log-marginal likelihood, experiments showed that our method consistently outperforms approaches using potentially tight lower bounds. Quantifying this gap or considering potential corrections present interesting directions for future work. Furthermore, while the same variational approach is applicable to any exponential family dynamical system, specific distributions will have their own associated challenges, offering avenues for further research.

Given that neural computation is inherently nonlinear, system identification methods capable of modeling nonlinear dynamical systems are essential for advancing neuroscience. General SSMs can perform well on inferring smoothed latent state trajectories *without* learning a good model of the nonlinear dynamics. Our proposed method, XFADS, can not only perform efficient system identification and smoothing but also forecast future state evolution for population recordings—a hallmark of a meaningful nonlinear dynamical model; using a causal inference network, XFADS can be used for real-time monitoring, feedback control, and online optimal experimental design, opening the door for new kinds of basic and clinical neuroscience experiments. Future work will focus on developing network architectures for precision matrix updates that are more parameter efficient when the rank those updates become moderate. Moreover, while Alg. 1 remains applicable in the confines of the generative model constraints considered, in certain scenarios, such as when $S > L$, modifications will need to be made to Alg. 2 to minimize time complexity.

Acknowledgements

We would like to thank the anonymous reviewers and Scott Linderman for their insightful comments and constructive feedback, which greatly improved the quality of this paper. We thank Mahmoud Elmakki for help with organizing the code base and developing helpful demos. Yuan Zhao was supported in part by the National Institute of Mental Health Intramural Research Program (ZIC-MH002968). This work was supported by NIH RF1-DA056404 and the Portuguese Recovery and Resilience Plan (PPR), through project number 62, Center for Responsible AI, and the Portuguese national funds, through FCT – Fundação para a Ciência e a Tecnologia – in the context of the project UIDB/04443/2020.

References

- [1] Kailath, T., Sayed, A. H. & Hassibi, B. *Linear estimation*. Prentice Hall, 2000.
- [2] Särkkä, S. *Bayesian filtering and smoothing*. Cambridge University Press, 2013. ISBN 9781107619289.
- [3] Anderson, B. D. O. & Moore, J. B. *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, N.J., 1979. ISBN 978-0-13-638122-8.
- [4] Kingma, D. P. & Welling, M. Auto-Encoding variational bayes. In *International Conference on Learning Representation*, May 2014.
- [5] Karl, M., Soelch, M., Bayer, J. & Smagt, P.van der . Deep variational Bayes filters: Unsupervised learning of state space models from raw data. In *International Conference on Learning Representations*, 2017.
- [6] Krishnan, R. G., Shalit, U. & Sontag, D. A. Structured inference networks for nonlinear state space models. In *AAAI Conference on Artificial Intelligence*, 2016.
- [7] Pandarinath, C., O’Shea, D. J., Collins, J., Jozefowicz, R., Stavisky, S. D., Kao, J. C., Trautmann, E. M., Kaufman, M. T., Ryu, S. I., Hochberg, L. R. & others, . Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.
- [8] Van Overschee, P. & De Moor, B. *Subspace identification for linear systems*. Springer US, Boston, MA, 1996. ISBN 9781461380610,9781461304654.
- [9] Turner, R. E. & Sahani, M. Two problems with variational expectation maximisation for time-series models. In Barber, D., Cemgil, T. & Chiappa, S., editors, *Bayesian Time series models*, chapter 5, pages 109–130. Cambridge University Press, 2011.
- [10] Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [11] Rezende, D. J., Mohamed, S. & Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.
- [12] Wainwright, M. J. & Jordan, M. I. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.
- [13] Diaconis, P. & Ylvisaker, D. Conjugate priors for exponential families. *The Annals of statistics*, pages 269–281, 1979.
- [14] Seeger, M. Expectation propagation for exponential families. Technical report, University of California at Berkeley, 2005.
- [15] Khan, M. & Lin, W. Conjugate-Computation Variational Inference : Converting Variational Inference in Non-Conjugate Models to Inferences in Conjugate Models. In Singh, A. & Zhu, J., editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 878–887. PMLR, 20–22 Apr 2017.
- [16] Beal, M. J. *Variational algorithms for approximate Bayesian inference*. University of London, University College London (United Kingdom), 2003.
- [17] Alaa, A. M. & Schaar, M.van der . Attentive state-space modeling of disease progression. *Advances in neural information processing systems*, 32, 2019.
- [18] Girin, L., Leglaive, S., Bie, X., Diard, J., Hueber, T. & Alameda-Pineda, X. Dynamical variational autoencoders: A comprehensive review. *arXiv preprint arXiv:2008.12595*, 2020.
- [19] Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H. & Davidson, J. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019.
- [20] Johnson, M. J., Duvenaud, D., Wiltchko, A. B., Datta, S. R. & Adams, R. P. Structured VAEs: Composing probabilistic graphical models and variational autoencoders. *arXiv preprint arXiv:1603.06277*, 2:2016, 2016.
- [21] Zhao, Y. & Linderman, S. Revisiting structured variational autoencoders. In *International Conference on Machine Learning*, pages 42046–42057. PMLR, 2023.

- [22] Douc, R., Moulines, E. & Stoffer, D. *Nonlinear time series: theory, methods and applications with R examples*. CRC press, 2014.
- [23] Whiteley, N. & Lee, A. Twisted particle filters. 2014.
- [24] Lawson, D., Raventós, A., Warrington, A. & Linderman, S. Sixo: Smoothing inference with twisted objectives. *Advances in Neural Information Processing Systems*, 35:38844–38858, 2022.
- [25] Cseke, B. & Heskes, T. Approximate marginals in latent gaussian models. *The Journal of Machine Learning Research*, 12:417–454, 2011.
- [26] Botev, A., Jaegle, A., Wirnsberger, P., Hennes, D. & Higgins, I. Which priors matter? Benchmarking models for learning latent dynamics. *arXiv preprint arXiv:2111.05458*, 2021.
- [27] Jiang, X., Missel, R., Li, Z. & Wang, L. Sequential latent variable models for few-shot high-dimensional time-series forecasting. In *The Eleventh International Conference on Learning Representations*.
- [28] Missel, R. Torch-Neural-SSM. <https://github.com/qu-gg/torchssm>, 2022.
- [29] Krishnan, R. G., Shalit, U. & Sontag, D. Deep Kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.
- [30] Churchland, M. M., Cunningham, J. P., Kaufman, M. T., Foster, J. D., Nuyujukian, P., Ryu, S. I. & Shenoy, K. V. Neural population dynamics during reaching. *Nature*, 487(7405):51–56, 2012.
- [31] Pei, F., Ye, J., Zoltowski, D. M., Wu, A., Chowdhury, R. H., Sohn, H., O’Doherty, J. E., Shenoy, K. V., Kaufman, M. T., Churchland, M., Jazayeri, M., Miller, L. E., Pillow, J., Park, I. M., Dyer, E. L. & Pandarinath, C. Neural latents benchmark ’21: evaluating latent variable models of neural population activity. In *Advances in Neural Information Processing Systems (NeurIPS), Track on Datasets and Benchmarks*, 2021.
- [32] Sohn, H., Narain, D. & Jazayeri, N. M. M. Bayesian computation through cortical latent dynamics. *Neuron*, 103(5):934–947, sep 2019.
- [33] Khan, M. E. & Rue, H. The bayesian learning rule. *arXiv preprint arXiv:2107.04562*, 2021.
- [34] Cong, Y., Chen, B. & Zhou, M. Fast simulation of hyperplane-truncated multivariate normal distributions. 2017.
- [35] Elfving, S., Uchibe, E. & Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.
- [36] Jiang, X., Missel, R., Li, Z. & Wang, L. Sequential latent variable models for few-shot high-dimensional time-series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023.
- [37] Li, X., Wong, T.-K. L., Chen, R. T. & Duvenaud, D. Scalable gradients for stochastic differential equations. In *International Conference on Artificial Intelligence and Statistics*, pages 3870–3882. PMLR, 2020.

Contents

A	Nomenclature	14
B	Variational filtering	14
B.1	Variational predict step	14
B.2	Variational update step	15
B.3	Variational smoothing step	15
B.4	Efficiently sampling structured marginals	16
B.5	Efficient filtering	16
C	Evaluating the KL using low-rank structure	16
C.1	Smoothing inference network	16
C.2	Causal/streaming inference network	17
D	Comparison method details	18
D.1	SVAE	18
D.2	DVBF	19
D.3	DKF	19
E	Experimental details	19
E.1	High-dimensional linear dynamical system	19
E.2	Time complexity	19
E.3	Pendulum	19
E.4	Bouncing ball	20
E.5	MC_Maze	21
E.6	DMFC_RSG	22
E.7	Collated results	23
F	Useful expressions	23
G	Forward KL fixed point	25
H	Linear Gaussian (information form) smoothing	25

A Nomenclature

Symbol	Description
SSM	state-space model
LGSSM	linear and Gaussian state-space model
$\pi(\mathbf{z}_t)$	variational approximation, $\pi(\mathbf{z}_t) \approx p(\mathbf{z}_t \mathbf{y}_{1:T})$
λ_t	natural parameters of $\pi(\mathbf{z}_t)$
$\boldsymbol{\mu}_t$	mean parameters of $\pi(\mathbf{z}_t)$
$\mathbf{m}_t / \mathbf{P}_t$	mean / covariance of $\pi(\mathbf{z}_t)$
$\boldsymbol{\alpha}_t$	local natural parameter update, $\boldsymbol{\alpha}_\phi(\mathbf{y}_t)$
$\boldsymbol{\beta}_{t+1}$	backward natural parameter update, $\boldsymbol{\beta}_\phi(\mathbf{y}_{t+1:T})$
$\bar{\pi}(\mathbf{z}_t)$	variational approximation with mean parameters $\bar{\boldsymbol{\mu}}_t = \mathbb{E}_{\pi_{t-1}}[\boldsymbol{\mu}_\theta(\mathbf{z}_{t-1})]$
$\bar{\lambda}_t$	natural parameters of $\bar{\pi}(\mathbf{z}_t)$
$\bar{\boldsymbol{\mu}}_t$	mean parameters of $\bar{\pi}(\mathbf{z}_t)$
$\bar{\mathbf{m}}_t / \bar{\mathbf{P}}_t$	mean / covariance of $\bar{\pi}(\mathbf{z}_t)$
$\boldsymbol{\theta}$	parameters of the dynamics and initial condition
$p_\theta(\mathbf{z}_t \mathbf{z}_{t-1})$	prior over state transitions
$p_\theta(\mathbf{z}_1)$	prior over initial condition
ψ	parameters of the observation model/likelihood
$p_\psi(\mathbf{y}_t \mathbf{z}_t)$	observation model/likelihood

B Variational filtering

Principles of Bayesian inference make it straightforward to write down an algorithm recursively computing the filtering posterior, $p(\mathbf{z}_t | \mathbf{y}_{1:t})$. Given, $p(\mathbf{z}_{t-1} | \mathbf{y}_{1:t-1})$, updating our belief to $p(\mathbf{z}_t | \mathbf{y}_{1:t})$ after observing \mathbf{y}_t can be broken down into two steps: first, we marginalize $p(\mathbf{z}_{t-1} | \mathbf{y}_{1:t-1})$ through the dynamics to obtain the predictive distribution,

$$\bar{p}(\mathbf{z}_t | \mathbf{y}_{1:t-1}) = \mathbb{E}_{p(\mathbf{z}_{t-1} | \mathbf{y}_{1:t-1})} [p_\theta(\mathbf{z}_t | \mathbf{z}_{t-1})] \quad \text{(predict step)} \quad (30)$$

Then, we update our belief by incorporating \mathbf{y}_t through Bayes' rule,

$$p(\mathbf{z}_t | \mathbf{y}_{1:t}) \propto p_\psi(\mathbf{y}_t | \mathbf{z}_t) \bar{p}(\mathbf{z}_t | \mathbf{y}_{1:t-1}) \quad \text{(update step)} \quad (31)$$

With all of the filtered/predictive beliefs, the **smoothing step** is given by,

$$p(\mathbf{z}_t | \mathbf{y}_{1:T}) = \mathbb{E}_{p(\mathbf{z}_{t+1} | \mathbf{y}_{1:T})} [p(\mathbf{z}_t | \mathbf{z}_{t+1}, \mathbf{y}_{1:t})] = p(\mathbf{z}_t | \mathbf{y}_{1:t}) \frac{\mathbb{E}_{p(\mathbf{z}_{t+1} | \mathbf{y}_{1:T})} [p_\theta(\mathbf{z}_{t+1} | \mathbf{z}_t)]}{\mathbb{E}_{p(\mathbf{z}_t | \mathbf{y}_{1:t})} [p_\theta(\mathbf{z}_{t+1} | \mathbf{z}_t)]} \quad (32)$$

However, these steps can usually not be evaluated in closed form when we depart from assumptions of Gaussianity and linearity. For nonlinear Gaussian dynamics the predict step can not be carried out exactly, and for nonlinear or non-Gaussian observations neither can the update step.

Alternatively, by considering variational analogues of the predict / update steps, we can develop a recursive and fully differentiable procedure for finding approximations $\pi(\mathbf{z}_t) \approx p(\mathbf{z}_t | \mathbf{y}_{1:t})$. In developing the variational analogues, it is assumed the approximations belong to an exponential family of distributions, (i.e. $\pi \in \mathcal{Q}$ where \mathcal{Q} is an exponential family distribution) – not necessarily Gaussian.

B.1 Variational predict step

Similar to developing a recursive algorithm as in the exact case, given $\pi(\mathbf{z}_{t-1}) \approx p(\mathbf{z}_{t-1} | \mathbf{y}_{1:t-1})$, we approximately marginalize $\pi(\mathbf{z}_{t-1})$ through the dynamics, by solving the following variational (forward KL / moment-matching) problem,

$$\bar{\pi}(\mathbf{z}_t) = \operatorname{argmin}_{\bar{\pi} \in \mathcal{Q}} \mathbb{D}_{\text{KL}}(\mathbb{E}_{\pi(\mathbf{z}_{t-1})} [p_\theta(\mathbf{z}_t | \mathbf{z}_{t-1})] || \bar{\pi}(\mathbf{z}_t)) \quad (33)$$

So that if $p_\theta(\mathbf{z}_t | \mathbf{z}_{t-1}) \in \mathcal{Q}$, the optimization problem is minimized when the mean parameters of $\bar{\pi}(\mathbf{z}_t)$, denoted $\bar{\boldsymbol{\mu}}_t$, are set to the expected mean parameter transformation under $\pi(\mathbf{z}_{t-1})$,

$$\bar{\boldsymbol{\mu}}_t = \mathbb{E}_{\pi(\mathbf{z}_{t-1})} [\boldsymbol{\mu}_\theta(\mathbf{z}_{t-1})] \quad \text{(variational predict step)} \quad (34)$$

For a LGSSM with $p_{\theta}(\mathbf{z}_t | \mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{F}\mathbf{z}_{t-1}, \mathbf{Q})$, using the fact that,

$$\boldsymbol{\mu}_{\theta}(\mathbf{z}_{t-1}) = \begin{bmatrix} \mathbf{F}\mathbf{z}_{t-1} \\ -\frac{1}{2}(\mathbf{F}\mathbf{z}_{t-1}\mathbf{z}_{t-1}^{\top}\mathbf{F}^{\top} + \mathbf{Q}_{\theta}) \end{bmatrix} \quad (35)$$

means that if $\pi(\mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{m}_{t-1}, \mathbf{P}_{t-1})$, setting the mean and variance of $\bar{\pi}(\mathbf{z}_t)$ to,

$$\bar{\mathbf{m}}_t = \mathbf{F}\mathbf{m}_{t-1} \quad (36)$$

$$\bar{\mathbf{P}}_t = \mathbf{F}\mathbf{P}_{t-1}\mathbf{F}^{\top} + \mathbf{Q}_{\theta} \quad (37)$$

minimizes the forward KL objective, and reassuringly, recovers the familiar Kalman filter predict step.

B.2 Variational update step

For the variational analogue of the filtering update step, we use $\bar{\pi}(\mathbf{z}_t)$ as a prior for the latest observation, \mathbf{y}_t , and solve the following variational (reverse KL) problem,

$$\pi(\mathbf{z}_t) = \underset{\pi \in \mathcal{Q}}{\operatorname{argmin}} \mathbb{D}_{\text{KL}}(\pi(\mathbf{z}_t) || p_{\psi}(\mathbf{y}_t | \mathbf{z}_t)\bar{\pi}(\mathbf{z}_t)) \quad (38)$$

If we denote the natural parameters of $\pi(\mathbf{z}_t)$ by $\boldsymbol{\lambda}_t$, then the optimal $\boldsymbol{\lambda}_t$ satisfy the implicit equation³³,

$$\boldsymbol{\lambda}_t = \nabla_{\boldsymbol{\mu}_t} \mathbb{E}_{\pi_t} [\log p_{\psi}(\mathbf{y}_t | \mathbf{z}_t)] + \bar{\boldsymbol{\lambda}}_t \quad (\text{variational update step}) \quad (39)$$

This usually requires an iterative optimization procedure, except when the likelihood is conjugate to $\pi(\mathbf{z}_t)$ in which case, the likelihood must take the following form with respect to \mathbf{z}_t ,

$$p_{\psi}(\mathbf{y}_t | \mathbf{z}_t) \propto \exp(\mathcal{T}(\mathbf{z}_t)^{\top} \tilde{\boldsymbol{\lambda}}_t) \quad (40)$$

so that, as expected, the natural parameters of the solution are given as Bayes' rule would suggest – by adding the data dependent update to the natural parameters of the prior so that,

$$\boldsymbol{\lambda}_t = \tilde{\boldsymbol{\lambda}}_t + \bar{\boldsymbol{\lambda}}_t \quad (41)$$

For a LGSSM with $p_{\psi}(\mathbf{y}_t | \mathbf{z}_t) = \mathcal{N}(\mathbf{C}\mathbf{z}_t, \mathbf{R})$, this results in the following updates,

$$\mathbf{h}_t = \bar{\mathbf{h}}_t + \mathbf{C}^{\top}\mathbf{R}^{-1}\mathbf{y}_t \quad (42)$$

$$\mathbf{J}_t = \bar{\mathbf{J}}_t + \mathbf{C}^{\top}\mathbf{R}^{-1}\mathbf{C} \quad (43)$$

which reassuringly recover the information form of the Kalman filter update step¹⁶.

B.3 Variational smoothing step

Letting $\check{q}(\mathbf{z}_t) \approx p(\mathbf{z}_t | \mathbf{y}_{1:t})$ and $q(\mathbf{z}_{t+1}) \approx p(\mathbf{z}_{t+1} | \mathbf{y}_{1:T})$, we can calculate the statistics of $q(\mathbf{z}_t)$ approximating the smoothed marginal posterior, by minimizing the following objective with respect to the marginal distribution, $q(\mathbf{z}_t)$,

$$\mathcal{L}_S(\check{q}) = \mathbb{D}_{\text{KL}}(\check{q}(\mathbf{z}_t) || \mathbb{E}_{\check{q}(\mathbf{z}_{t+1})} [q(\mathbf{z}_t | \mathbf{z}_{t+1})]) \quad (44)$$

$$= \mathbb{D}_{\text{KL}}(\check{q}(\mathbf{z}_t) || q(\mathbf{z}_t)) - \mathbb{E}_{\check{q}(\mathbf{z}_t)} \left[\log \mathbb{E}_{\check{q}(\mathbf{z}_{t+1})} \left[\frac{p_{\theta}(\mathbf{z}_{t+1} | \mathbf{z}_t)}{\mathbb{E}_{q(\mathbf{z}_t)} [p_{\theta}(\mathbf{z}_{t+1} | \mathbf{z}_t)]} \right] \right] \quad (45)$$

$$\approx \mathbb{D}_{\text{KL}}(\check{q}(\mathbf{z}_t) || q(\mathbf{z}_t)) - \mathbb{E}_{\check{q}(\mathbf{z}_t)} \left[\log \mathbb{E}_{\check{q}(\mathbf{z}_{t+1})} \left[\frac{p_{\theta}(\mathbf{z}_{t+1} | \mathbf{z}_t)}{\bar{q}(\mathbf{z}_{t+1})} \right] \right] := \hat{\mathcal{L}}_S(\check{q}) \quad (46)$$

taking natural gradients, we find that at a fixed point, the smoothed marginal posterior parameters satisfy the implicit relationship,

$$\boldsymbol{\lambda}_t = \check{\boldsymbol{\lambda}}_t + \nabla_{\boldsymbol{\mu}_t} \mathbb{E}_{q(\mathbf{z}_t)} \left[A(\boldsymbol{\lambda}_{\theta}(\mathbf{z}_t) + \boldsymbol{\lambda}_{t+1} - \bar{\boldsymbol{\lambda}}_{t+1}) - A(\boldsymbol{\lambda}_{\theta}(\mathbf{z}_t)) \right] \quad (47)$$

B.4 Efficiently sampling structured marginals

Algorithm 2 Nonlinear variational filtering

Input: $\mathbf{k}_{1:T}, \mathbf{K}_{1:T}$
for $t = 1$ **to** T **do**
 $\bar{\mathbf{m}}_t = S^{-1} \sum \mathbf{m}_\theta(\mathbf{z}_{t-1}^s)$
 $\bar{\mathbf{M}}_t^c = S^{-1/2} [\mathbf{m}_\theta(\mathbf{z}_{t-1}^1) - \bar{\mathbf{m}}_t \cdots \mathbf{m}_\theta(\mathbf{z}_{t-1}^S) - \bar{\mathbf{m}}_t]$
 $\bar{\mathbf{\Upsilon}}_t = \text{Cholesky}(\mathbf{I}_S + \bar{\mathbf{M}}_t^{c\top} \mathbf{Q}^{-1} \bar{\mathbf{M}}_t^c)^{-1}$
 $\bar{\mathbf{h}}_t = \bar{\mathbf{P}}_t^{-1} \bar{\mathbf{m}}_t$ # using $[\mathbf{Q}, \bar{\mathbf{M}}_t^c, \bar{\mathbf{\Upsilon}}_t]$ and Eq. (50)
 $\mathbf{\Upsilon}_t = \text{Cholesky}(\mathbf{I}_r + \mathbf{K}_t^\top \bar{\mathbf{P}}_t \mathbf{K}_t)^{-1}$ # using $[\mathbf{Q}, \bar{\mathbf{M}}_t^c]$ and Eq. (23)
 $\mathbf{h}_t = \bar{\mathbf{h}}_t + \mathbf{k}_t$
 $\mathbf{m}_t = \mathbf{P}_t \mathbf{h}_t$ # using $[\mathbf{Q}, \bar{\mathbf{M}}_t^c, \mathbf{K}_t, \mathbf{\Upsilon}_t]$ and Eqs. (52) and (23)
 $\bar{\mathbf{w}}_t^s \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{L+S})$
 $\bar{\mathbf{z}}_t^s = \bar{\mathbf{P}}_t^{1/2} \bar{\mathbf{w}}_t^s$ # using $[\mathbf{Q}, \bar{\mathbf{M}}_t^c]$ and Eq. (23)
 $\mathbf{w}_t^s \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_r)$
 $\mathbf{z}_t^s = \mathbf{m}_t + \bar{\mathbf{z}}_t^s - \mathbf{K}_t \mathbf{\Upsilon}_t \mathbf{\Upsilon}_t^\top (\mathbf{K}_t^\top \bar{\mathbf{z}}_t^s + \mathbf{w}_t^s)$
end for
Output: $\mathbf{z}_{1:T}^{1:S}, \mathbf{m}_{1:T}, \bar{\mathbf{m}}_{1:T}, \mathbf{\Upsilon}_{1:T}$

While \mathbf{P}_t has a structured representation, drawing samples from $\pi(\mathbf{z}_t)$ is not straightforward because we do not have a structured representation for a square-root of \mathbf{P}_t . However, using the factorization of $\bar{\mathbf{P}}_t$ in Eq. (23), it is possible to sample from $\mathcal{N}(\mathbf{0}, \bar{\mathbf{P}}_t)$ efficiently since $\bar{\mathbf{P}}_t^{1/2} = [\bar{\mathbf{M}}_t^c \mathbf{Q}^{1/2}]$. Now, combining the fact that the posterior marginal can be written as,

$$\begin{aligned} \pi(\mathbf{z}_t) &= \mathcal{N}(\mathbf{m}_t, \mathbf{P}_t) \\ &= \mathcal{N}(\mathbf{m}_t, (\bar{\mathbf{P}}_t^{-1} + \mathbf{K}_t \mathbf{K}_t^\top)^{-1}) \end{aligned} \quad (48)$$

with the result from Cong et al.³⁴, stating that sampling $\mathbf{z}_t^s \sim \pi(\mathbf{z}_t)$ is equivalent to sampling $\mathbf{w}_t^s \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{L+S})$ and $\bar{\mathbf{z}}_t^s \sim \mathcal{N}(\mathbf{0}, \bar{\mathbf{P}}_t)$, and then setting

$$\mathbf{z}_t^s = \mathbf{m}_t + \bar{\mathbf{z}}_t^s - \mathbf{K}_t \mathbf{\Upsilon}_t \mathbf{\Upsilon}_t^\top (\mathbf{K}_t^\top \bar{\mathbf{z}}_t^s + \mathbf{w}_t^s), \quad (49)$$

makes it possible to efficiently draw samples from the posterior marginal.

B.5 Efficient filtering

Evaluating $\bar{\mathbf{h}}_t = \bar{\mathbf{P}}_t^{-1} \bar{\mathbf{m}}_t$ and MVMs with $\bar{\mathbf{P}}_t^{-1}$, can be carried out in $\mathcal{O}(LS + S^2)$ time, after an initial cost of $\mathcal{O}(LS^2 + S^3)$ to factorize $\bar{\mathbf{\Upsilon}}_t \bar{\mathbf{\Upsilon}}_t^\top = (\mathbf{I}_S + \bar{\mathbf{M}}_t^{c\top} \mathbf{Q}_\theta^{-1} \bar{\mathbf{M}}_t^c)^{-1}$, by applying the Woodbury identity to (23),

$$\bar{\mathbf{P}}_t^{-1} = \mathbf{Q}_\theta^{-1} - \mathbf{Q}_\theta^{-1} \bar{\mathbf{M}}_t^c \bar{\mathbf{\Upsilon}}_t \bar{\mathbf{\Upsilon}}_t^\top \bar{\mathbf{M}}_t^{c\top} \mathbf{Q}_\theta^{-1} \quad (50)$$

Since this specifies all quantities that characterize $\bar{\pi}(\mathbf{z}_t)$, following (13), next is to update our belief by adding the information from the pseudo observation to them,

$$\mathbf{h}_t = \bar{\mathbf{h}}_t + \mathbf{k}_t \quad \mathbf{P}_t^{-1} = \bar{\mathbf{P}}_t^{-1} + \mathbf{K}_t \mathbf{K}_t^\top \quad (51)$$

As a result, the multiplication in $\mathbf{m}_t = \mathbf{P}_t \mathbf{h}_t$ requires $\mathcal{O}(LS + Lr)$ time, and, analogous to (50), the square root factor $\mathbf{\Upsilon}_t$ in

$$\mathbf{P}_t = \bar{\mathbf{P}}_t - \bar{\mathbf{P}}_t \mathbf{K}_t \mathbf{\Upsilon}_t \mathbf{\Upsilon}_t^\top \mathbf{K}_t^\top \bar{\mathbf{P}}_t \quad (52)$$

requires $\mathcal{O}(r^3 + LSr + S^2r)$ time where $\mathbf{\Upsilon}_t \mathbf{\Upsilon}_t^\top = (\mathbf{I}_r + \mathbf{K}_t^\top \bar{\mathbf{P}}_t \mathbf{K}_t)^{-1}$.

C Evaluating the KL using low-rank structure

C.1 Smoothing inference network

Efficient training of the generative model and inference networks require efficient numerical evaluation of the ELBO. We take advantage of the structured precision matrices arising from low-rank updates

and sample approximations. The expected log likelihood can be evaluated using a Monte Carlo approximation from samples during the filtering pass. The KL term,

$$\mathbb{D}_{\text{KL}}(\pi(\mathbf{z}_t) \parallel \bar{\pi}(\mathbf{z}_t)) = \frac{1}{2} \left[(\bar{\mathbf{m}}_t - \mathbf{m}_t)^\top \bar{\mathbf{P}}_t^{-1} (\bar{\mathbf{m}}_t - \mathbf{m}_t) + \text{tr}(\bar{\mathbf{P}}_t^{-1} \mathbf{P}_t) + \log \frac{|\bar{\mathbf{P}}_t|}{|\mathbf{P}_t|} - L \right] \quad (53)$$

can be evaluated in closed form and we can expand each term as,

Log-determinant. Writing

$$\log |\mathbf{P}_t| = -\log |\bar{\mathbf{P}}_t^{-1} + \mathbf{K}_t \mathbf{K}_t^\top| \quad (54)$$

$$= \log |\bar{\mathbf{P}}_t| - \log |\mathbf{I} + \mathbf{K}_t^\top \bar{\mathbf{P}}_t \mathbf{K}_t| \quad (55)$$

gives

$$\log \frac{|\bar{\mathbf{P}}_t|}{|\mathbf{P}_t|} = \log |\mathbf{I} + \mathbf{K}_t^\top \bar{\mathbf{P}}_t \mathbf{K}_t| \quad (56)$$

$$= -2 \sum_{i=1}^r [\boldsymbol{\Upsilon}_t]_{i,i} \quad (57)$$

Trace. Writing,

$$\text{tr}(\bar{\mathbf{P}}_t^{-1} \mathbf{P}_t) = \text{tr}(\mathbf{I}_L - \mathbf{K}_t (\mathbf{I} + \mathbf{K}_t^\top \bar{\mathbf{P}}_t \mathbf{K}_t)^{-1} \mathbf{K}_t^\top \bar{\mathbf{P}}_t) \quad (58)$$

$$= L - \text{tr}(\bar{\mathbf{P}}_t^{\top/2} \mathbf{K}_t (\mathbf{I} + \mathbf{K}_t^\top \bar{\mathbf{P}}_t \mathbf{K}_t)^{-1} \mathbf{K}_t^\top \bar{\mathbf{P}}_t^{1/2}) \quad (59)$$

$$= L - \text{tr}(\boldsymbol{\Upsilon}_t^\top \mathbf{K}_t^\top \bar{\mathbf{P}}_t \mathbf{K}_t \boldsymbol{\Upsilon}_t) \quad (60)$$

which by taking $\boldsymbol{\Upsilon}_t$ to be an $r \times r$ square root such that,

$$\boldsymbol{\Upsilon}_t \boldsymbol{\Upsilon}_t^\top = (\mathbf{I} + \mathbf{K}_t^\top \bar{\mathbf{P}}_t \mathbf{K}_t)^{-1} \quad (61)$$

further simplifies to

$$\text{tr}(\bar{\mathbf{P}}_t^{-1} \mathbf{P}_t) = L - \text{tr}(\bar{\mathbf{M}}_t^{c\top} \mathbf{K}_t \boldsymbol{\Upsilon}_t \boldsymbol{\Upsilon}_t^\top \mathbf{K}_t^\top \bar{\mathbf{M}}_t^c) - \text{tr}(\mathbf{Q}^{\top/2} \mathbf{K}_t \boldsymbol{\Upsilon}_t \boldsymbol{\Upsilon}_t^\top \mathbf{K}_t^\top \mathbf{Q}^{1/2}) \quad (62)$$

$$= L - \text{tr}(\bar{\mathbf{M}}_t^{c\top} \mathbf{K}_t \boldsymbol{\Upsilon}_t \boldsymbol{\Upsilon}_t^\top \mathbf{K}_t^\top \bar{\mathbf{M}}_t^c) - \text{tr}(\boldsymbol{\Upsilon}_t^\top \mathbf{K}_t^\top \mathbf{Q}^{1/2} \mathbf{Q}^{\top/2} \mathbf{K}_t \boldsymbol{\Upsilon}_t) \quad (63)$$

note that the size of the triple product, $\bar{\mathbf{M}}_t^{c\top} \mathbf{K}_t \boldsymbol{\Upsilon}_t$, is $S \times r$.

C.2 Causal/streaming inference network

When the real-time parameterization of the inference network is used the expressions become slightly more complicated due to the more intricate relationship between the posterior at time t and the posterior predictive at time t .

Log-determinant. We need to first find $\log |\bar{\mathbf{P}}_{t|T}| - \log |\mathbf{P}_t|$ so begin with using the matrix-determinant lemma to write,

$$|\bar{\mathbf{P}}_{t|T}| = |\mathbf{M}_{t|T}^c \mathbf{M}_{t|T}^{c\top} + \mathbf{Q}| \quad (64)$$

$$= |\mathbf{I}_S + \mathbf{M}_{t|T}^{c\top} \mathbf{Q}^{-1} \mathbf{M}_{t|T}^c| \times |\mathbf{Q}| \quad (65)$$

then expand the smoothed covariance to write,

$$|\mathbf{P}_t| = |(\check{\mathbf{P}}_t^{-1} + \mathbf{B}_t \mathbf{B}_t^\top)^{-1}| \quad (66)$$

$$= |\check{\mathbf{P}}_t^{-1} + \mathbf{B}_t \mathbf{B}_t^\top|^{-1} \quad (67)$$

$$= (|\mathbf{I}_{r_\beta} + \mathbf{B}_t^\top \check{\mathbf{P}}_t \mathbf{B}_t| \times |\check{\mathbf{P}}_t^{-1}|)^{-1} \quad (68)$$

$$= |\mathbf{I}_{r_\beta} + \mathbf{B}_t^\top \check{\mathbf{P}}_t \mathbf{B}_t|^{-1} |\check{\mathbf{P}}_t| \quad (69)$$

and another time to write,

$$|\check{\mathbf{P}}_t^{-1}| = |\bar{\mathbf{P}}_t^{-1} + \mathbf{A}_t \mathbf{A}_t^\top| \quad (70)$$

$$= |\mathbf{I}_{r_\alpha} + \mathbf{A}_t^\top \bar{\mathbf{P}}_t \mathbf{A}_t| \times |\bar{\mathbf{P}}_t^{-1}| \quad (71)$$

and another time,

$$|\bar{\mathbf{P}}_t| = |\mathbf{M}_t^c \mathbf{M}_t^{c\top} + \mathbf{Q}| \quad (72)$$

$$= |\mathbf{I}_S + \mathbf{M}_t^{c\top} \mathbf{Q}^{-1} \mathbf{M}_t^c| \times |\mathbf{Q}| \quad (73)$$

When combined we are finally able to write

$$\log |\bar{\mathbf{P}}_{t|T}| - \log |\mathbf{P}_t| = \log |\mathbf{I}_S + \mathbf{M}_{t|T}^{c\top} \mathbf{Q}^{-1} \mathbf{M}_{t|T}^c| + \log |\mathbf{I}_{r_\beta} + \mathbf{B}_t^\top \check{\mathbf{P}}_t \mathbf{B}_t| \quad (74)$$

$$+ \log |\mathbf{I}_{r_\alpha} + \mathbf{A}_t^\top \bar{\mathbf{P}}_t \mathbf{A}_t| - \log |\mathbf{I}_S + \mathbf{M}_t^{c\top} \mathbf{Q}^{-1} \mathbf{M}_t^c| \quad (75)$$

For the initial condition we have,

$$\log |\bar{\mathbf{P}}_1| - \log |\mathbf{P}_1| = \log |\mathbf{I}_{r_\beta} + \mathbf{B}_1^\top \check{\mathbf{P}}_1 \mathbf{B}_1| + \log |\mathbf{I}_{r_\alpha} + \mathbf{A}_1^\top \bar{\mathbf{P}}_1 \mathbf{A}_1| \quad (76)$$

Trace. For the trace,

$$\text{tr}(\bar{\mathbf{P}}_{t|T}^{-1} \mathbf{P}_t) = \text{tr}(\mathbf{Q}^{-1} \mathbf{P}_t) - \text{tr}(\mathbf{Q}^{-1} \mathbf{M}_{t|T}^c (\mathbf{I}_S + \mathbf{M}_{t|T}^{c\top} \mathbf{Q}^{-1} \mathbf{M}_{t|T}^c)^{-1} \mathbf{M}_{t|T}^{c\top} \mathbf{Q}^{-1} \mathbf{P}_t) \quad (77)$$

$$= \text{tr}(\mathbf{Q}^{-1} \mathbf{P}_t) - \text{tr}(\mathbf{Q}^{-1} \mathbf{M}_{t|T}^c \check{\mathbf{\Upsilon}}_{t|T} \check{\mathbf{\Upsilon}}_{t|T}^\top \mathbf{M}_{t|T}^{c\top} \mathbf{Q}^{-1} \mathbf{P}_t) \quad (78)$$

$$= \text{tr}(\mathbf{P}_t \mathbf{Q}^{-1}) - \text{tr}([\mathbf{P}_t \mathbf{Q}^{-1} \mathbf{M}_{t|T}^c \check{\mathbf{\Upsilon}}_{t|T}] \check{\mathbf{\Upsilon}}_{t|T}^\top \mathbf{M}_{t|T}^{c\top} \mathbf{Q}^{-1}) \quad (79)$$

where we expand the first rhs term for a numerically efficient implementation by writing

$$\text{tr}(\mathbf{P}_t \mathbf{Q}^{-1}) = \text{tr}(\bar{\mathbf{P}}_t \mathbf{Q}^{-1}) - \text{tr}(\bar{\mathbf{P}}_t \mathbf{A}_t (\mathbf{I}_{r_\alpha} + \mathbf{A}_t^\top \bar{\mathbf{P}}_t \mathbf{A}_t)^{-1} \mathbf{A}_t^\top \bar{\mathbf{P}}_t \mathbf{Q}^{-1}) \quad (80)$$

$$- \text{tr}(\check{\mathbf{P}}_t \mathbf{B}_t (\mathbf{I}_{r_\beta} + \mathbf{B}_t^\top \check{\mathbf{P}}_t \mathbf{B}_t)^{-1} \mathbf{B}_t^\top \check{\mathbf{P}}_t \mathbf{Q}^{-1}) \quad (81)$$

To reduce notational clutter we define,

$$\check{\mathbf{\Upsilon}}_t \check{\mathbf{\Upsilon}}_t^\top = (\mathbf{I}_S + \mathbf{M}_t^{c\top} \mathbf{Q}^{-1} \mathbf{M}_t^c)^{-1} \quad (82)$$

$$\check{\mathbf{\Upsilon}}_{t|T} \check{\mathbf{\Upsilon}}_{t|T}^\top = (\mathbf{I}_S + \mathbf{M}_{t|T}^{c\top} \mathbf{Q}^{-1} \mathbf{M}_{t|T}^c)^{-1} \quad (83)$$

$$\check{\mathbf{\Upsilon}}_t \check{\mathbf{\Upsilon}}_t^\top = (\mathbf{I}_{r_\alpha} + \mathbf{A}_t^\top \bar{\mathbf{P}}_t \mathbf{A}_t)^{-1} \quad (84)$$

$$\mathbf{\Upsilon}_t \mathbf{\Upsilon}_t^\top = (\mathbf{I}_{r_\beta} + \mathbf{B}_t^\top \bar{\mathbf{P}}_t \mathbf{B}_t)^{-1} \quad (85)$$

and so the trace is

$$\text{tr}(\bar{\mathbf{P}}_{t|T}^{-1} \mathbf{P}_t) = L + \text{tr}(\mathbf{M}_t^{c\top} \mathbf{Q}^{-1} \mathbf{M}_t^c) - \text{tr}(\mathbf{Q}^{-1/2} \bar{\mathbf{P}}_t \mathbf{A}_t \check{\mathbf{\Upsilon}}_t \check{\mathbf{\Upsilon}}_t^\top \mathbf{A}_t^\top \bar{\mathbf{P}}_t \mathbf{Q}^{-1/2}) \quad (86)$$

$$- \text{tr}(\mathbf{Q}^{-1/2} \check{\mathbf{P}}_t \mathbf{B}_t \mathbf{\Upsilon}_t \mathbf{\Upsilon}_t^\top \mathbf{B}_t^\top \check{\mathbf{P}}_t \mathbf{Q}^{-1/2}) \quad (87)$$

$$- \text{tr}([\mathbf{P}_t \mathbf{Q}^{-1} \mathbf{M}_{t|T}^c \check{\mathbf{\Upsilon}}_{t|T}] \check{\mathbf{\Upsilon}}_{t|T}^\top \mathbf{M}_{t|T}^{c\top} \mathbf{Q}^{-1}) \quad (88)$$

which is now in a form that is easy to handle using fast MVMs with $\bar{\mathbf{P}}_t$, $\check{\mathbf{P}}_t$, \mathbf{P}_t .

For the initial condition term we use the fact that $\bar{\mathbf{P}}_1$ is diagonal,

$$\text{tr}(\mathbf{P}_1 \bar{\mathbf{P}}_1^{-1}) = L - \text{tr}(\bar{\mathbf{P}}_1^{1/2} \mathbf{A}_1 \check{\mathbf{\Upsilon}}_1 \check{\mathbf{\Upsilon}}_1^\top \mathbf{A}_1^\top \bar{\mathbf{P}}_1^{-1/2}) - \text{tr}(\bar{\mathbf{P}}_1^{-1/2} \check{\mathbf{P}}_1 \mathbf{B}_1 \mathbf{\Upsilon}_1 \mathbf{\Upsilon}_1^\top \mathbf{B}_1^\top \check{\mathbf{P}}_1 \bar{\mathbf{P}}_1^{-1/2}) \quad (89)$$

D Comparison method details

D.1 SVAE

For the SVAE²⁰, the latent state prior is a linear dynamical system parameterized as,

$$p_\theta(\mathbf{z}_t | \mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t | \mathbf{F} \mathbf{z}_{t-1}, \mathbf{Q}) \quad (90)$$

Using conjugate potentials, with likelihood $p(\tilde{\mathbf{y}}_t | \mathbf{z}_t) = \exp(t(\mathbf{z}_t)^\top \boldsymbol{\alpha}(\mathbf{y}_t))$, the approximate posterior is given by $q(\mathbf{z}_{1:T}) = \prod p(\tilde{\mathbf{y}}_t | \mathbf{z}_t) p_\theta(\mathbf{z}_{1:T})$ so that its statistics can be found by applying Kalman filtering/smoothing to the pseudo-observations. In this case, the ELBO can be evaluated as

$$\mathcal{L}(q) = \sum \mathbb{E}_{q_t} [\log p(\mathbf{y}_t | \mathbf{z}_t)] - \mathbb{E}_{q_t} [\log p(\tilde{\mathbf{y}}_t | \mathbf{z}_t)] + \log \mathbb{E}_{\bar{q}_t} [p(\tilde{\mathbf{y}}_t | \mathbf{z}_t)] \quad (91)$$

where $\bar{q}_t := \bar{q}(\mathbf{z}_t) = p(\mathbf{z}_t | \tilde{\mathbf{y}}_{1:t-1})$ is the filtering predictive distribution. These expressions can be evaluated in closed form and written concisely in terms of natural/mean parameters and log-partition functions as

$$\mathcal{L}(q) = \sum \mathbb{E}_{q_t} [\log p(\mathbf{y}_t | \mathbf{z}_t)] - \boldsymbol{\mu}_t^\top \boldsymbol{\alpha}_t + A(\bar{\boldsymbol{\lambda}}_t + \boldsymbol{\alpha}_t) - A(\bar{\boldsymbol{\lambda}}_t) \quad (92)$$

where $\boldsymbol{\mu}_t = \nabla A(\bar{\boldsymbol{\lambda}}_t + \boldsymbol{\alpha}_t + \boldsymbol{\beta}_{t+1})$. Using the identities given in App. F, these expressions can be written in more familiar mean/covariance parameters.

D.2 DVBF

For the DVBF⁵, we parameterize the latent state prior using a nonlinear dynamical system of the same form as Eq. (1). Then, using an inference network that encodes data in reverse-time to produce the parameters of a diagonal Gaussian distribution, $\mathbf{w}_t \sim q(\mathbf{w}_t) = \mathcal{N}(\mathbf{m}_t, \text{diag}(\mathbf{s}_t))$, we sample the latent trajectory forward using the recursion, $\mathbf{z}_t = \mathbf{m}_\theta(\mathbf{z}_{t-1}) + \mathbf{Q}^{1/2}\mathbf{w}_t$. Parameters of the generative model/inference network are learned jointly by minimizing the ELBO,

$$\mathcal{L}(q) = \sum \mathbb{E}_{q(\mathbf{z}_t)} [\log p_\psi(\mathbf{y}_t | \mathbf{z}_t)] - \mathbb{D}_{\text{KL}}(q(\mathbf{w}_t) || p(\mathbf{w}_t)) \quad (93)$$

where, $p(\mathbf{w}_t) = \mathcal{N}(\mathbf{0}, \mathbf{I})$.

D.3 DKF

For the DKF⁶, the latent state is also parameterized using a nonlinear dynamical system of the same form as Eq. (1). We follow the parameterization outlined in the text with S2S(\cdot) implemented using a recurrent neural network. We sample trajectories using the inference network and jointly train all parameters on the ELBO,

$$\mathcal{L}(q) = \sum \mathbb{E}_{q_t} [\log p(\mathbf{y}_t | \mathbf{z}_t)] - \mathbb{E}_{q_{t-1}} [\mathbb{D}_{\text{KL}}(q(\mathbf{z}_t | \mathbf{z}_{t-1}) || p_\theta(\mathbf{z}_t | \mathbf{z}_{t-1}))] \leq \log p(\mathbf{y}_{1:T}) \quad (94)$$

E Experimental details

In describing the neural network architectures used to parameterize the inference model, it will be useful to define the following multilayer perceptron (MLP), with SiLU nonlinearity³⁵, that gets used repeatedly:

$$- \text{MLP}(n_{\text{in}}, n_{\text{hidden}}, n_{\text{out}}) : [\text{Linear}(n_{\text{in}}, n_{\text{hidden}}), \text{SiLU}(), \text{Linear}(n_{\text{hidden}}, n_{\text{out}})]$$

$$\mathcal{L}(q) = \sum \mathbb{E}_{q_t} [\log p(\mathbf{y}_t | \mathbf{z}_t)] - \mathbb{E}_{q_{t-1}} [\mathbb{D}_{\text{KL}}(q(\mathbf{z}_t | \mathbf{z}_{t-1}) || p_\theta(\mathbf{z}_t | \mathbf{z}_{t-1}))] \leq \log p(\mathbf{y}_{1:T}) \quad (95)$$

E.1 High-dimensional linear dynamical system

We simulated data from an LDS generative model (with dynamics restricted to the set of matrices with singular values less than 1) for latent dimensions $L \in [20, 50, 100]$ over 3 random seeds for two scenarios **i**) $N = L$ and **ii**) $N = L/5$. For each scenario, we also vary the rank of the local/backward encoder precision updates.

E.2 Time complexity

We generate random LDS systems of appropriate dimension and measure the average time to complete one forward pass and take a gradient step. The system used for benchmarking wall-clock time was an RTX 4090 with 128GB of RAM with an AMD 5975WX processor.

E.3 Pendulum

We consider the pendulum system from²⁶. We generate 500/150/150 trials of length 100 for training/validation/testing. All methods are trained for 5000 epochs for 3 different random seeds. We consider a context window of 50 images and a forecast window of 50 images. A decoder was fit from the latent state on the training set during the context window; then, for held out data, we examine performance of the decoder during the context and forecast windows.

The generative model is parameterized as:

- $L = 4$
- $N = 256$
- $T = 50$
- likelihood

- $p_{\psi}(\mathbf{y}_t | \mathbf{z}_t) = \mathcal{N}(\mathbf{z}_t | \mathbf{C}_{\psi}(\mathbf{z}_t) + \mathbf{b}, \mathbf{R})$
- \mathbf{C}_{ψ} : MLP(4, 128, 256)

For each method, inference is amortized using the following neural network architectures:

- our inference network
 - $r_{\alpha} = 4$
 - $r_{\beta} = 4$
 - α_{ϕ} : MLP(256, 128, 20)
 - β_{ϕ} : [GRU(128), Linear(128, 20)]
- DKF inference network
 - \mathbf{u}_{ϕ} : GRU(128)
 - $(\mathbf{m}_{\phi}, \log \mathbf{P}_{\phi})$: MLP(132, 128, 8)
- DVBF inference network
 - \mathbf{u}_{ϕ} : [GRU(128)]
 - $(\boldsymbol{\mu}_{\phi}, \log \boldsymbol{\sigma}_{\phi})$: MLP(132, 128, 8)
- SVAE inference network
 - α_{ϕ} : MLP(256, 128, 20)

Optimization and training details:

- optimizer: Adam(lr = 0.001)
- batch size: 128

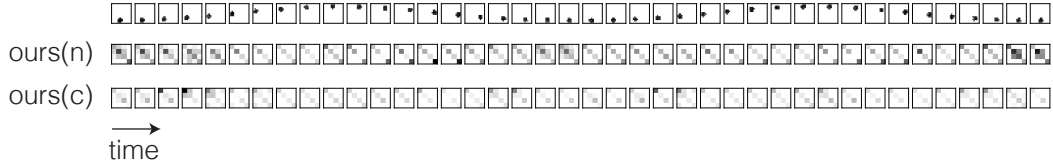


Figure 4: **Learned covariance of nonlinear SSMs.** (top) single trial of a pendulum. Below are the posterior covariances output by the causal and non-causal variants of XFADS.

E.4 Bouncing ball

We consider a bouncing ball dataset commonly used as a baseline to benchmark the performance of inference and learning in deep state-space models^{26,36,36}. For this dataset we take 500/150/150 trials of length 75 for training/validation/testing. All methods are trained for 5000 epochs for 3 different random seeds. The generative model is parameterized as:

- $L = 8$
- $N = 256$
- $T = 50$
- likelihood
 - $p_{\psi}(\mathbf{y}_t | \mathbf{z}_t) = \mathcal{N}(\mathbf{z}_t | \mathbf{C}_{\psi}(\mathbf{z}_t) + \mathbf{b}, \mathbf{R})$
 - \mathbf{C}_{ψ} : MLP(8, 128, 256)

For each method, inference is amortized using the following neural network architectures:

- our inference network
 - $r_{\alpha} = 8$
 - $r_{\beta} = 4$
 - α_{ϕ} : MLP(256, 128, 72)

- β_ϕ : [GRU(128), Linear(128, 40)]
- DKF inference network
 - \mathbf{u}_ϕ : GRU(128)
 - $(\mathbf{m}_\phi, \log \mathbf{P}_\phi)$: MLP(132, 128, 16)
- DVBF inference network
 - \mathbf{u}_ϕ : [GRU(128)]
 - $(\boldsymbol{\mu}_\phi, \log \boldsymbol{\sigma}_\phi)$: MLP(132, 128, 16)
- SVAE inference network
 - $\boldsymbol{\alpha}_\phi$: MLP(256, 128, 72)

Optimization and training details:

- optimizer: Adam(lr = 0.001)
- batch size: 128



Figure 5: **Learned covariance of nonlinear SSMs.** (top) single trial of a bouncing ball. Below are the posterior covariances output by the nonlinear SSMs considered – qualitatively, we observe more complex covariance structures arise when the ball hits the wall that diagonal approximations cannot capture.

E.5 MC_Maze

The monkey reaching dataset of³⁰ was the first real dataset examined in the main text. For this dataset, we partitioned 1800/200/200 training/validation/testing trials sampled at 20ms per bin. All methods are trained for 1000 epochs for 3 different random seeds. The generative model is parameterized as:

- $L = 40$
- $N = 182$
- $T = 35$
- likelihood
 - $p_\psi(\mathbf{y}_t | \mathbf{z}_t) = \text{Poisson}(\mathbf{z}_t | \mathbf{C}_\psi(\mathbf{z}_t) + \mathbf{b})$
 - \mathbf{C}_ψ : Linear(40, 182)

For each method, inference is amortized using the following neural network architectures:

- our inference network
 - $r_\alpha = 15$
 - $r_\beta = 5$
 - $\boldsymbol{\alpha}_\phi$: MLP(182, 128, 640)
 - β_ϕ : [GRU(128), Linear(128, 240)]
- DKF inference network
 - \mathbf{u}_ϕ : GRU(128)
 - $(\mathbf{m}_\phi, \log \mathbf{P}_\phi)$: MLP(128, 128, 80)

- DVBF inference network
 - \mathbf{u}_ϕ : [GRU(128)]
 - $(\mathbf{m}_\phi, \log \mathbf{P}_\phi)$: MLP(128, 128, 80)
- SVAE inference network
 - α_ϕ : MLP(182, 256, 1640)

Optimization and training details:

- optimizer: Adam(lr = 0.001)
- batch size: 128

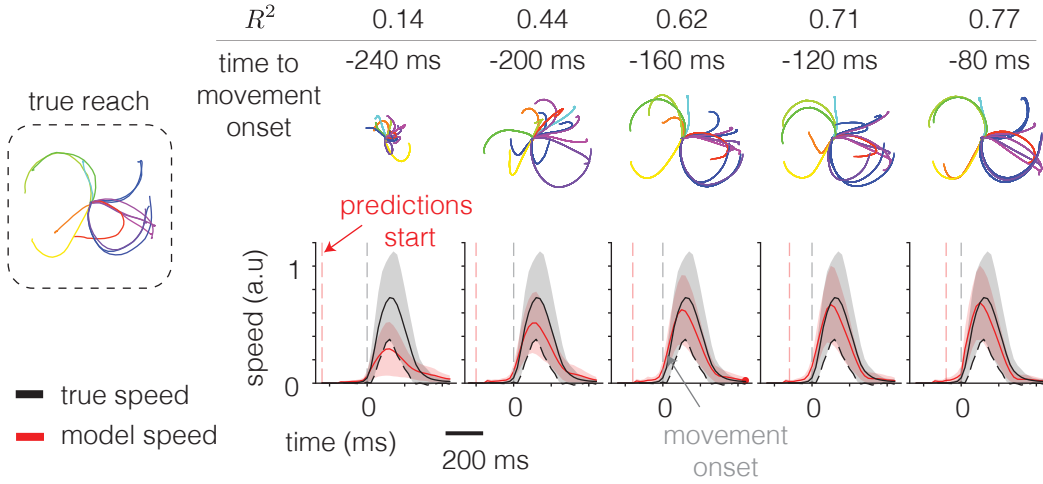


Figure 6: **XFADS predictive capabilities on real data.** On the left are example reaches the monkey made during several trials. Using the learned model from the monkey reaching experiment, we filtered neural activity starting from -240 ms up until -80 ms before movement onset (each vertical panel represents the result from filtering more and more data) and unroll the final filtered states through the dynamics (starting from the dashed red line) to make predictions.

E.6 DMFC_RSG

The second real dataset examined was the timing interval reproduction task of³² samples at 10ms bins. For this dataset, we partitioned 700/150/150 training/validation/testing trials. All methods are trained for 1000 epochs for 3 different random seeds. The generative model is parameterized as:

- $L = 40$
- $N = 54$
- $T = 130$
- likelihood
 - $p_\psi(\mathbf{y}_t | \mathbf{z}_t) = \text{Poisson}(\mathbf{z}_t | \mathbf{C}_\psi(\mathbf{z}_t) + \mathbf{b})$
 - \mathbf{C}_ψ : Linear(40, 54)

For each method, inference is amortized using the following neural network architectures:

- our inference network
 - $r_\alpha = 15$
 - $r_\beta = 5$
 - α_ϕ : MLP(54, 128, 640)
 - β_ϕ : [GRU(128), Linear(128, 240)]

- DKF inference network
 - \mathbf{u}_ϕ : GRU(128)
 - $(\mathbf{m}_\phi, \log \mathbf{P}_\phi)$: MLP(128, 128, 80)
- DVBF inference network
 - \mathbf{u}_ϕ : [GRU(128)]
 - $(\boldsymbol{\mu}_\phi, \log \boldsymbol{\sigma}_\phi)$: MLP(128, 128, 80)
- SVAE inference network
 - $\boldsymbol{\alpha}_\phi$: MLP(256, 128, 1640)

Optimization and training details:

- optimizer: Adam(lr = 0.001)
- batch size: 128

E.7 Collated results

	pendulum		bouncing ball		monkey reaching	
	smoothing	prediction	smoothing	prediction	smoothing	prediction
ours (n)	99.7 (.01)	72.7 (3.4)	80.7 (.70)	34.6 (1.5)	88.5 (.1)	52.7 (1.7)
ours (c)	99.6 (.01)	69.7 (3.0)	80.4 (.60)	29.1 (4.7)	87.8 (.1)	53.3 (1.1)
DKF	99.2 (.70)	19.1 (9.5)	65.6 (4.9)	0.32 (10.)	81.2 (.3)	1.7 (9.1)
DVBF	90.7 (2.8)	-19.6 (9.8)	75.3 (.70)	1.3 (.20)	86.3 (.1)	33.4 (.5)
SVAE	98.4 (.40)	-39.7 (10.)	76.5 (2.1)	-23.3 (.70)	87.5 (.1)	-2.4 (13.)
L-SDE	92.1 (2.3)	13.8 (12.)	81.3 (.70)	23.1 (1.3)	76.6 (.4)	18.7 (1.0)
LFADS					91.1 (.1)	23.2 (1.1)

Figure 7: **Collated results including the latent SDE (L-SDE)**³⁷. We ran an additional baseline on pendulum/bouncing ball/monkey reaching datasets. L-SDE achieves results with a similar level of performance as the other models. Listed are the R^2 values for decoding angular velocity/x-y position/reach velocity from the latent representations learned for each dataset in both smoothing and prediction regimes.

F Useful expressions

mean/natural parameter inner product One common expression that frequently arises is the inner product between a mean and natural parameter, i.e.

$$\boldsymbol{\mu}_t^\top \boldsymbol{\alpha}_t \tag{96}$$

where in the Gaussian case if the mean/natural parameter coordinates are,

$$\boldsymbol{\mu}_t = \left(-\frac{1}{2}(\mathbf{P}_t + \mathbf{m}_t \mathbf{m}_t^\top) \right) \quad \boldsymbol{\alpha}_t = \left(\begin{array}{c} \mathbf{a}_t \\ \mathbf{A}_t \mathbf{A}_t^\top \end{array} \right) \tag{97}$$

then,

$$\boldsymbol{\mu}_t^\top \boldsymbol{\alpha}_t = \mathbf{m}_t^\top \mathbf{a}_t - \frac{1}{2} \|\mathbf{A}_t^\top \mathbf{m}_t\|^2 - \frac{1}{2} \text{tr}(\mathbf{A}_t^\top \mathbf{P}_t \mathbf{A}_t) \tag{98}$$

difference of log partition functions Another common expression that frequently arises is given by,

$$A(\bar{\lambda}_t + \alpha_t) - A(\bar{\lambda}_t) \quad (99)$$

so that if,

$$\bar{\lambda}_t = \begin{pmatrix} \bar{\mathbf{P}}_t^{-1} \bar{\mathbf{m}}_t \\ \bar{\mathbf{P}}_t^{-1} \end{pmatrix} \quad (100)$$

then,

$$A(\bar{\lambda}_t + \alpha_t) - A(\bar{\lambda}_t) \quad (101)$$

$$= \frac{1}{2} \check{\mathbf{m}}_t^\top (\bar{\mathbf{P}}_t^{-1} + \mathbf{A}_t \mathbf{A}_t^\top) \check{\mathbf{m}}_t - \frac{1}{2} \log |\bar{\mathbf{P}}_t^{-1} + \mathbf{A}_t \mathbf{A}_t^\top| \quad (102)$$

$$- \frac{1}{2} \bar{\mathbf{m}}_t \bar{\mathbf{P}}_t^{-1} \bar{\mathbf{m}}_t + \frac{1}{2} \log |\bar{\mathbf{P}}_t^{-1}|$$

$$= \frac{1}{2} \left(\|\check{\mathbf{m}}_t\|_{\bar{\mathbf{P}}_t^{-1}}^2 - \|\bar{\mathbf{m}}_t\|_{\bar{\mathbf{P}}_t^{-1}}^2 + \|\mathbf{A}_t^\top \check{\mathbf{m}}_t\|^2 - \log |\mathbf{I} + \mathbf{A}_t^\top \bar{\mathbf{P}}_t \mathbf{A}_t| \right) \quad (103)$$

conditional linear Gaussian log partition For a Gaussian distribution, the log-partition function in terms of the natural parameters, $\lambda = (\mathbf{h}, \text{vec}(\mathbf{J}))$, is given by

$$A(\lambda) = \frac{1}{2} \mathbf{h}^\top \mathbf{J}^{-1} \mathbf{h} - \frac{1}{2} \log |\mathbf{J}| \quad (104)$$

so that, when

$$\mathbf{F} = \begin{pmatrix} \mathbf{Q}^{-1} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \quad \mathbf{f} = \begin{pmatrix} \mathbf{0} \\ \mathbf{Q}^{-1} \end{pmatrix} \quad (105)$$

and $\mathbf{u}^\top = (\mathbf{u}_1^\top \text{vec}(\mathbf{u}_2)^\top)$ is an arbitrary constant, we can write $A(\mathbf{F}t(\mathbf{z}) + \mathbf{f} + \mathbf{u}) = \mathbf{a}^\top t(\mathbf{z}) + b$ for some \mathbf{a} and b . To show this, we can just expand the log partition function

$$A(\mathbf{F}t(\mathbf{z}) + \mathbf{f} + \mathbf{u}) = \frac{1}{2} \mathbf{z}^\top \mathbf{A}^\top \mathbf{Q}^{-1} (\mathbf{Q}^{-1} + \mathbf{u}_2)^{-1} \mathbf{Q}^{-1} \mathbf{A} \mathbf{z} + \mathbf{z}^\top \mathbf{A}^\top \mathbf{Q}^{-1} (\mathbf{Q}^{-1} + \mathbf{u}_2)^{-1} \mathbf{u}_1 \quad (106)$$

$$+ \frac{1}{2} \mathbf{u}_1^\top (\mathbf{Q}^{-1} + \mathbf{u}_2)^{-1} \mathbf{u}_1 - \frac{1}{2} \log |\mathbf{u}_2 + \mathbf{Q}^{-1}|$$

$$= \mathbf{a}^\top t(\mathbf{z}) + b \quad (107)$$

then \mathbf{a} and b can be identified as

$$\mathbf{a} = \begin{pmatrix} \mathbf{A}^\top \mathbf{Q}^{-1} (\mathbf{Q}^{-1} + \mathbf{U}_2)^{-1} \mathbf{u}_1 \\ -\mathbf{A}^\top \mathbf{Q}^{-1} (\mathbf{Q}^{-1} + \mathbf{U}_2)^{-1} \mathbf{Q}^{-1} \mathbf{A} \end{pmatrix} \quad b = \frac{1}{2} \mathbf{u}_1^\top (\mathbf{Q}^{-1} + \mathbf{u}_2)^{-1} \mathbf{u}_1 - \frac{1}{2} \log |\mathbf{Q}^{-1} + \mathbf{u}_2| \quad (108)$$

Then, for a LGSSM, we can evaluate the following expression which describes the difference between predictive and smoothed marginals,

$$\lambda_t - \bar{\lambda}_t = \alpha_t + \nabla_{\mu_t} \mathbb{E}_{q(\mathbf{z}_t)} [A(\lambda_\theta(\mathbf{z}_t) + \lambda_{t+1} - \bar{\lambda}_{t+1}) - A(\lambda_\theta(\mathbf{z}_t))] \quad (109)$$

$$= \alpha_t + \begin{pmatrix} \mathbf{A}^\top \mathbf{Q}^{-1} (\mathbf{Q}^{-1} + \mathbf{P}_{t+1}^{-1} - \bar{\mathbf{P}}_{t+1}^{-1})^{-1} (\mathbf{h}_{t+1} - \bar{\mathbf{h}}_{t+1}) \\ \mathbf{A}^\top \mathbf{S}_{t+1} \mathbf{A} \end{pmatrix} \quad (110)$$

$$= \alpha_t + \beta_{t+1} \quad (111)$$

where $\mathbf{S}_{t+1} = \mathbf{Q}^{-1} - \mathbf{Q}^{-1} (\mathbf{Q}^{-1} + \mathbf{P}_{t+1}^{-1} - \bar{\mathbf{P}}_{t+1}^{-1})^{-1} \mathbf{Q}^{-1}$.

G Forward KL fixed point

Using the moment matching property of the fixed point solution to the forward KL objective, we can write

$$\bar{\boldsymbol{\mu}}_{t+1}^* = \int t(\mathbf{z}_{t+1}) \mathbb{E}_{q_t} [p_{\boldsymbol{\theta}}(\mathbf{z}_{t+1} | \mathbf{z}_t)] d\mathbf{z}_{t+1} \quad (112)$$

$$= \int t(\mathbf{z}_{t+1}) \quad (113)$$

$$\times \left[\int h(\mathbf{z}_t) h(\mathbf{z}_{t+1}) \exp(t(\mathbf{z}_{t+1})^\top \boldsymbol{\lambda}_{\boldsymbol{\theta}}(\mathbf{z}_t) + t(\mathbf{z}_t)^\top \boldsymbol{\lambda}_t - A(\boldsymbol{\lambda}_t) - A(\boldsymbol{\lambda}_{\boldsymbol{\theta}}(\mathbf{z}_t))) d\mathbf{z}_t \right] d\mathbf{z}_{t+1}$$

$$= \int h(\mathbf{z}_t) \exp(t(\mathbf{z}_t)^\top \boldsymbol{\lambda}_t - A(\boldsymbol{\lambda}_t)) \quad (114)$$

$$\times \left[\int t(\mathbf{z}_{t+1}) h(\mathbf{z}_{t+1}) \exp(t(\mathbf{z}_{t+1})^\top \boldsymbol{\lambda}_{\boldsymbol{\theta}}(\mathbf{z}_t) - A(\boldsymbol{\lambda}_{\boldsymbol{\theta}}(\mathbf{z}_t))) d\mathbf{z}_{t+1} \right] d\mathbf{z}_t$$

$$= \mathbb{E}_{q_t} [\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}_t)] \quad (115)$$

This result holds for any setting where stochastic transitions, $p_{t+1|t}$, and the approximate marginal distributions, q_t , are restricted to the same exponential family distribution.

H Linear Gaussian (information form) smoothing

Using the derived relation,

$$\mathbf{P}_t^{-1} = \bar{\mathbf{P}}_t^{-1} + \mathbf{C}^\top \mathbf{R}^{-1} + \mathbf{F}^\top \mathbf{S}_{t+1} \mathbf{F} \quad (116)$$

means that,

$$\mathbf{P}_t^{-1} - \bar{\mathbf{P}}_t^{-1} = \mathbf{C}^\top \mathbf{R}^{-1} \mathbf{C} + \mathbf{F}^\top \mathbf{S}_{t+1} \mathbf{F} \quad (117)$$

so that using the definition of \mathbf{S}_t (rewritten for convenience) then plugging in, we get,

$$\mathbf{S}_t = \mathbf{Q}^{-1} - \mathbf{Q}^{-1} (\mathbf{Q}^{-1} + \mathbf{P}_t^{-1} - \bar{\mathbf{P}}_t^{-1})^{-1} \mathbf{Q}^{-1} \quad (118)$$

$$= \mathbf{Q}^{-1} - \mathbf{Q}^{-1} (\mathbf{Q}^{-1} + \mathbf{C}^\top \mathbf{R}^{-1} \mathbf{C} + \mathbf{F}^\top \mathbf{S}_{t+1} \mathbf{F}) \mathbf{Q}^{-1} \quad (119)$$

which gives a recurrence backward in time for $\mathbf{S}_{1:T}$ starting from $\mathbf{S}_{T+1} = \mathbf{0}$. Now since,

$$\mathbf{h}_t - \bar{\mathbf{h}}_t = \mathbf{F}^\top \mathbf{Q}^{-1} (\mathbf{Q}^{-1} + \mathbf{P}_{t+1}^{-1} - \bar{\mathbf{P}}_{t+1}^{-1})^{-1} (\mathbf{h}_{t+1} - \bar{\mathbf{h}}_{t+1}) \quad (120)$$

and,

$$(\mathbf{Q}^{-1} + \mathbf{C}^\top \mathbf{R}^{-1} \mathbf{C} + \mathbf{F}^\top \mathbf{S}_{t+1} \mathbf{F})^{-1} = \mathbf{Q} - \mathbf{Q} \mathbf{S}_t \mathbf{Q} \quad (121)$$

we have that,

$$\mathbf{F}^\top \mathbf{s}_t = \mathbf{F}^\top \mathbf{Q}^{-1} (\mathbf{Q}^{-1} + \mathbf{C}^\top \mathbf{R}^{-1} \mathbf{C} + \mathbf{F}^\top \mathbf{S}_{t+1} \mathbf{F})^{-1} (\mathbf{C}^\top \mathbf{R}^{-1} \mathbf{y}_{t+1} + \mathbf{F}^\top \mathbf{s}_{t+1}) \quad (122)$$

which simplifies into a recursion for $\mathbf{s}_{1:T}$ backward in time starting from $\mathbf{s}_{T+1} = \mathbf{0}$, given by,

$$\mathbf{s}_t = \mathbf{Q}^{-1} (\mathbf{Q}^{-1} + \mathbf{C}^\top \mathbf{R}^{-1} \mathbf{C} + \mathbf{F}^\top \mathbf{S}_{t+1} \mathbf{F})^{-1} (\mathbf{C}^\top \mathbf{R}^{-1} \mathbf{y}_{t+1} + \mathbf{F}^\top \mathbf{s}_{t+1}) \quad (123)$$

$$= (\mathbf{I} - \mathbf{S}_t \mathbf{Q}) (\mathbf{C}^\top \mathbf{R}^{-1} \mathbf{y}_{t+1} + \mathbf{F}^\top \mathbf{s}_{t+1}) \quad (124)$$

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: we try to clearly state our contributions.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We discuss potential limitations of the work.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We list assumptions about the types of generative models we consider.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.

- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide experimental details in the appendix that include the architectures and training regimes.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide sufficient instructions and algorithmic details.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.

- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: We provide details about splits, optimization, and parameterizations in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [\[Yes\]](#)

Justification: We discuss the neuroscientific implications of the results our model could be used to discover from neural data.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: we provide our computing setup.

Answer: [Yes]

Justification: We provide the specifications of our computing setup.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines?>

Answer: [Yes]

Justification: we adhere to the code of ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss the broader impacts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: the paper poses no such risks

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: we produced all code to run the experiments, we cite real data from where they were procured.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New Assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: the paper does not release new assets

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and Research with Human Subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve crowdsourcing or human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: this paper does not involve human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.