Structured Recognition for Generative Models with Explaining Away

Changmin Yu[1,2] **Hugo Soulat**[2] **Neil Burgess**[1] **Maneesh Sahani**[2]
[1]Institute of Cognitive Neuroscience; [2]Gatsby Computational Neuroscience Unit;
UCL, London, United Kingdom
{changmin.yu.19; hugo.soulat.19; n.burgess}@ucl.ac.uk;
maneesh@gatsby.ucl.ac.uk

Abstract

A key goal of unsupervised learning is to go beyond density estimation and sample generation to reveal the structure inherent within observed data. Such structure can be expressed in the pattern of interactions between explanatory latent variables captured through a probabilistic graphical model. Although the learning of structured graphical models has a long history, much recent work in unsupervised modelling has instead emphasised flexible deep-network-based generation, either transforming independent latent generators to model complex data or assuming that distinct observed variables are derived from different latent nodes. Here, we extend amortised variational inference to incorporate structured factors over multiple variables, able to capture the observation-induced posterior dependence between latents that results from "explaining away" and thus allow complex observations to depend on multiple nodes of a structured graph. We show that appropriately parametrised factors can be combined efficiently with variational message passing in rich graphical structures. We instantiate the framework in nonlinear Gaussian Process Factor Analysis, evaluating the structured recognition framework using synthetic data from known generative processes. We fit the GPFA model to high-dimensional neural spike data from the hippocampus of freely moving rodents, where the model successfully identifies latent signals that correlate with behavioural covariates.

1 Introduction

A central challenge of unsupervised learning is to identify and model patterns of statistical dependence in high-dimensional data. One approach to this challenge exploits latent-variable generative models, which capture the statistical structure of data through the modelled influence of the latent variables on observations, and through interactions between the latents themselves. Recent developments in deep learning have enabled deep generative models (DGM), with remarkable success in density estimation and high-fidelity image or text generation [1–5]. However, much of the DGM development has emphasised the expressiveness and accuracy of the latent-to-observation generative process, with the latents themselves often assumed independent *a priori*.

Variational learning requires inferring or approximating the posterior distribution of the latents conditioned on observations [6]. In a variational autoencoder [VAE; 1, 2], inference for a DGM is *amortised* by training a recognition network to return parameters of the variational posterior. The structured variational autoencoder [SVAE; 7] connects the DGM framework with richer models that include structured prior dependence amongst the latent variables. In an SVAE the structured latent prior is combined with a DGM link between each latent and a corresponding observation. Variational inference is amortised using recognition networks that return parameters of a factor associated with each link, which are combined with the structured prior to obtain the full posterior. However, in

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

many generative models of interest, observations depend on more than one latent variable. Such interactions induce joint potentials in the likelihood, coupling latents in the posterior even if they are independent *a priori*, a phenomenon sometimes called "explaining away" [8]. To capture such observation-induced dependence in the DGM setting we develop the *structured recognition VAE* (SRVAE), where recognition potentials that incorporate joint factors induced by the observations are learnt in a (structured) variational autoencoding framework.

We instantiate the SRVAE framework for a range of graphical structures, including non-linear latent Gaussian process (GP) models. In the latter case, we bring together the GP-prior VAE [9, 10], and (sparse) Gaussian process factor analysis (GPFA) [11, 12] models, yielding a novel autoencoding model (SR-nlGPFA). Experiments show that SR-nlGPFA outperforms alternatives that lack structured recognition. We apply SR-nlGPFA to data collected from a population of neurons in the hippocampal complex, and show that the unsupervised approach automatically captures dimensions underlying neural firing that reflect relevant behavioural correlates.

2 Background

2.1 Variational Inference and Structured Variational Autoencoders

Consider a general generative process with latent variables z and observations y,

$$z \sim p(z|\theta), \quad y \sim p(y|z;\gamma)\,,$$

where θ and γ are the parameters of the prior and conditional likelihood distributions, respectively (and we write $\Theta = \theta \cup \gamma$). We assume deterministic θ for simplicity, but distributions over θ (with conjugate hyperpriors) can be incorporated by variational Bayes (VB) [13]. The posterior distribution, $p(z|y,\Theta) \propto p(z|\theta)p(y|z,\gamma)$, is often analytically intractable. In such cases, a common approach is to seek an approximation $q(z)$ constrained to a tractable class $\mathcal{Q}$ by variational inference (VI); that is, by minimising the KL-divergence with respect to the true posterior distribution or, equivalently, maximising a variational free energy [6] $\mathcal{F}_{\text{VI}} = \big\langle \log p(y,z|\Theta) - \log q(z) \big\rangle_{q(z)}$:

$$q_{\text{VI}(\mathcal{Q})}(z|y,\Theta) = \operatorname*{argmin}_{q\in\mathcal{Q}} \mathbf{KL}[q(z)\|p(z|y,\Theta)] = \operatorname*{argmax}_{q\in\mathcal{Q}} \big\langle \log p(y,z|\Theta) - \log q(z) \big\rangle_{q(z)}\,. \tag{1}$$

In the variational autoencoder (VAE) architecture, the parametric optimisation of $q(z)$ implicit in (1) is amortised by a recognition network that takes as input the observed values y and returns parameters of q. The parameters of this recognition network are then trained jointly with the generative model by stochastic optimisation of the free energy [1, 2].

In both standard and amortised VI, the approximate distribution q is often constrained to factor over the latent variables, a so-called *mean-field* constraint. While tractable, such mean-field approximations may be too restrictive to capture the complexity of the true posterior [14, 15]. Many approaches have been proposed for improving the expressiveness of variational approximation [16–19]. Here we consider the structured VAE [SVAE; 7]. Unlike standard VAEs, the SVAE assumes the generative prior distribution to be specified by a structured probabilistic graphical model (PGM), $p(z|\theta) \propto \prod_{c\in C} \psi_c(z_c)$, where $\{\psi_c\}_{c\in C}$ correspond to C clique potentials. In addition, the amortised inference network outputs recognition factors, $r(z|y,\phi)$, that approximate the generative likelihood function rather than the full variational posterior. These recognition factors are combined with the structured prior distribution to obtain the amortised variational posterior,

$$q_{\text{SVAE}}(z|y,\Theta,\phi) = \operatorname*{argmax}_{q\propto r(z|y,\phi)p(z|\theta)} \sum_{y\in\mathcal{Y}} \left\langle \log\left(\frac{p(z|\theta)p(y|z,\gamma)}{q(z)}\right)\right\rangle_{q(z)} \propto r(z|y,\phi^*)p(z|\theta)\,, \tag{2}$$

where ϕ^* minimises the averaged KL over the data observations $\mathcal{Y}$. Johnson et al. [7] considered recognition factors that are local (singleton) evidence potentials, chosen to be conjugate to $p(z|\theta)$. Even in this case, the form of (2) allows the dependency structure established by the prior to be carried over to the variational posterior distribution. Further details of SVAE appear in Appendix A.

2.2 Gaussian Process Factor Analysis

Gaussian Process Factor Analysis (GPFA) is a model used in neural data analysis to infer the dynamic latent structure underlying high-dimensional population spike trains [11, 12]. Standard GPFA assumes

the following generative model:

$$\begin{aligned} \text{latent functions: } & f^k(\cdot) \sim \mathcal{GP}\big(m^k_\theta(\cdot), \kappa^k_\theta(\cdot,\cdot)\big), \text{ for } k = 1, \dots, K\,, \\ \text{affine embeddings: } & h_n(\cdot) = \sum_{k=1}^{K} c_{nk} f^k(\cdot) + d_n, \text{ for } n = 1, \dots, N\,, \\ \text{observations: } & y_n(t) \sim p\big(y_n(t) | g(h_n(t))\big), \text{ for } t = 1, \dots, T\,, \end{aligned} \tag{3}$$

where $m^k_\theta(\cdot)$ and $\kappa^k_\theta(\cdot,\cdot)$ parametrise the GP prior, c_{jk} and d_n define an affine mapping from the latent space to the observation space (or a transform of the observations for general likelihoods), and $g(\cdot)$ is a smooth scalar link function appropriate for the observation distribution. We take $m^k_\theta(\cdot) = 0$ unless stated otherwise. Correlations between observed neurons are captured by dependence on the common set of latents, while temporal correlations in the high-dimensional observations are modelled by the latent temporal correlations of the GPs. The PGM describing the GPFA generative model (with sparse variational approximation, see below) is shown in Fig 1a.

Although maximum-likelihood parameters for GPFA can be found using (variational) expectation-maximisation [20, 11] exact GP inference scales cubically in the number of observation times [21]. Sparse variational GP (svGP) inference based on auxilliary inducing points [22] reduces the time complexity of learning in GPFA [23], and also facilitates efficient extensions to (pointwise) non-conjugate likelihoods [12].

For $k = 1, \dots, K$, we introduce inducing values $\mathbf{u}_k$ representing the evaluations of the latent process f^k at M_k inducing locations, $\mathbf{z}_k$. For simplicity, we assume $M_1 = \cdots = M_K = M$ unless stated otherwise. The GPFA generative model (3) can be augmented to include these auxiliary variables:

$$p(\mathbf{u}_k|\mathbf{z}_k) = \mathcal{N}\big(\mathbf{0}, \mathbf{K}^k_{\mathbf{z}_k\mathbf{z}_k}\big), \quad p(f^k(\cdot)|\mathbf{u}_k) = \mathcal{GP}(\mathfrak{F}_k(\cdot)\mathbf{u}_k, \kappa^k_\theta(\cdot,\cdot) - \mathfrak{F}_k(\cdot)\mathbf{K}^k_{\mathbf{z}_k\mathbf{z}_k}\mathfrak{F}^T_k(\cdot)), \tag{4}$$

where $\mathfrak{F}_k(\cdot) = \kappa^k_\theta(\cdot, \mathbf{z}_k)(\mathbf{K}^k_{\mathbf{z}_k\mathbf{z}_k})^{-1}$ is the linear operator that maps $\mathbf{u}_k$ to $f^k(\cdot)$; $\kappa^k_\theta(\cdot, \mathbf{z}_k)$ is a vector-valued function with $\kappa^k_\theta(x; \mathbf{z}_k) = [\kappa^k_\theta(x, z_{k1}), \kappa^k_\theta(x, z_{k2}), \dots, \kappa^k_\theta(x, z_{kM_k})]$; and $\mathbf{K}^k_{\mathbf{z}_k\mathbf{z}_k}$ is the covariance matrix obtained by evaluation of the kernel function $\kappa^k_\theta(\cdot)$ at the inducing locations $\mathbf{z}_k$.

Introducing a variational distribution over the inducing points and the latent functions, and utilising the generative model (Fig 1a), the free energy of the sparse variational GPFA (svGPFA) can be expressed as following:

$$\mathcal{F}(\Theta, \mathbf{C}, \mathbf{d}, \mathbf{Z}, \phi) = \left\langle \log \frac{\Big(\prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{f}_t; \gamma, \mathbf{C}, \mathbf{d})\Big) p(\mathbf{F}|\mathbf{U}; \theta) p(\mathbf{U}|\mathbf{Z}; \theta)}{q(\mathbf{F}, \mathbf{U}; \phi)} \right\rangle_{q(\mathbf{F}, \mathbf{U}; \phi)}. \tag{5}$$

where $\mathbf{y}_t = [y_n(t)] \in \mathbb{R}^N$, $\mathbf{f}_t = [f^k(t)] \in \mathbb{R}^K$, $\mathbf{F} = [\mathbf{f}_t]^T \in \mathbb{R}^{T\times K}$, $\mathbf{C} = [c_{nk}] \in \mathbb{R}^{N\times K}$, $\mathbf{d} = [d_n] \in \mathbb{R}^n$, $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_K]^T \in \mathbb{R}^{K\times M}$, and $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_K]^T \in \mathbb{R}^{K\times M}$. The svGP approach constrains $q(\mathbf{F}, \mathbf{U}; \phi)$ to the form $q(\mathbf{U})p(\mathbf{F}|\mathbf{U}, \theta)$ with Gaussian $q(\mathbf{U})$ [22]. In the multi-GP case, the distribution is typically also taken to factorise over processes [23], $q(\mathbf{F}, \mathbf{U}) = \prod_{k=1}^{K} p(\mathbf{f}^k|\mathbf{u}_k)q(\mathbf{u}_k)$, with $\mathbf{f}^k = [f^k(1), \dots, f^k(T)] \in \mathbb{R}^t$, $q(\mathbf{u}_k) = \mathcal{N}(\mathbf{m}_k, \mathbf{S}_k)$. Under these constraints on q, the svGPFA free energy simplifies to

$$\mathcal{F}_{\text{svGPFA}} = \sum_t \langle \log p(\mathbf{y}_t|\mathbf{h}_t)\rangle_{q(\mathbf{h}_t)} - \sum_{k=1}^{K} \mathbf{KL}[q(\mathbf{u}_k)\|p(\mathbf{u}_k|\mathbf{z}_k)]\,,$$

where $q(\mathbf{h}_t) = \int d\mathbf{U}\, p(\mathbf{h}_t|\mathbf{U})q(\mathbf{U})$ is itself a GP with mean and kernel functions given by

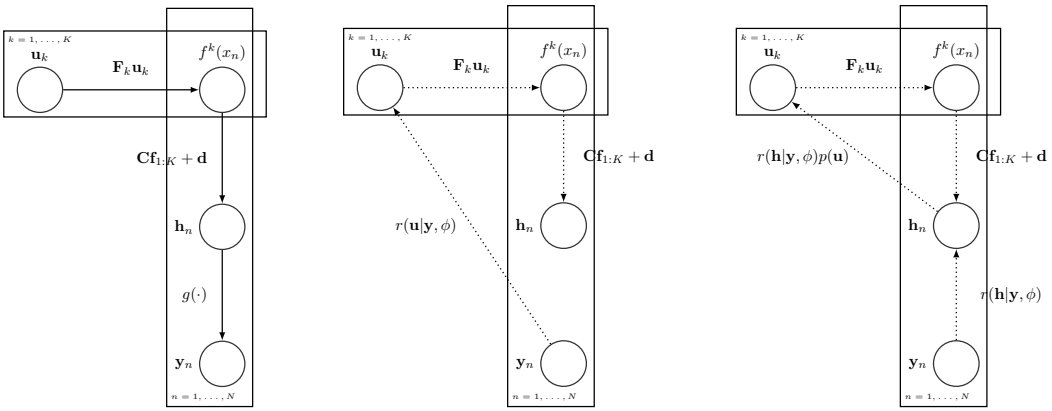
$$m^h_n(t) = \sum_{k=1}^{K} c_{nk}\mathfrak{F}_k(t)\mathbf{m}_k + d_n \text{ and } \nu^h_n(t,t') = \sum_{k=1}^{K} c^2_{nk}(\kappa_k(t,t') + \mathfrak{F}_k(t)(\mathbf{S}_k - \mathbf{K}_{\mathbf{z}_k\mathbf{z}_k})\mathfrak{F}_k(t')^T)\,.$$

Hence the computational cost for sparse variational GP inference reduces from $\mathcal{O}(T^3)$ down to $\mathcal{O}(M^3 + TM^2)$.

3 Structured Recognition and Explaining Away

3.1 Structured Recognition in VAEs

A general non-linear likelihood may induce additional latent dependency structure in the posterior beyond that of the prior (a simple illustrative example appears in Appendix C). Although the SVAE



(a) (b) (c)

Figure 1: **Graphical models for sparse variational GPFA models**. **(a)** Generative model of GPFA model with sparse inducing point approximation; **(b)** Standard GPFA inference with sparse amortised variational approximation (note that $\mathbf{C} = \mathbf{I}$ and $\mathbf{d} = \mathbf{0}$ for SGP-VAE [10]); **(c)** Structured recognition potential enables full-covariance variational inference in SR-nlGPFA model.

approach developed by Johnson et al. [7] composed amortised inference with a structured prior distribution, the recognition models these authors discussed all contributed amortised potentials that factored over latent variables. Such an approach cannot accurately model posteriors with dependence structure that differs from that of the prior PGM.

Here we adopt a recognition network that outputs structured factor potentials over the latents, providing amortised estimates of the additional dependencies induced by "explaining away". We denote the resulting model the *Structured Recognition VAE* (SRVAE). The SRVAE variational approximation takes the form

$$q_{\text{SRVAE}}(z|y, \theta, \phi) \propto \prod_{c \in C} \psi_c(z_c; \theta) \prod_{c \in C_r} \xi_c(z_c|y; \phi)\,, \tag{6}$$

where C_r is the set of recognition factors and the ξ_cs the associated factor potentials. We assume that the ξ_cs are chosen to be conjugate to the prior factors unless otherwise stated. Hence the analytical (approximate, if necessary) form of q_{SRVAE} can be computed with (variational) message passing. In the most general form, we could let $\xi(z|y; \phi)$ be a single joint factor potential over the complete set of latent variables.

The free energy objective takes the same form as the standard autoencoding free energy objective [1, 2], but with the structured amortised variational approximation, q_{SRVAE}, as the variational distribution.

$$\mathcal{F}_{\text{SRVAE}}(\Theta, \phi) = \langle \log p(z|\theta) + \log p(y|z, \gamma) - \log q_{\text{SRVAE}}(z|y, \phi, \theta) \rangle_{q_{\text{SRVAE}}(z|y,\phi,\theta)} \tag{7}$$

The training of the model follows standard VAE-style stochastic optimisation to update the parameters of the recognition and generative networks, as well as (optionally) the parameters of the prior PGM.

The following proposition supports the use of SRVAE framework (the proof appears in Appendix E).

Proposition 3.1. *The SRVAE objective function provides a tighter lower bound to the free energy than the SVAE objective function.*

$$\max_{q} \mathcal{F}_{VI(\mathcal{Q})}(\Theta, q) \geq \max_{\phi} \mathcal{F}_{SRVAE}(\Theta, \phi) \geq \max_{\phi} \mathcal{F}_{SVAE}(\Theta, \phi) \tag{8}$$

Below we describe two instantiations of the SRVAE framework, the first based on a latent Gaussian mixture model, and the second on a latent GPFA model, both with DGM outputs. Structured amortised inference facilitates scalable inference of the posterior latent distribution with full covariance structure, allowing more accurate learning than with factored recognition approaches. However, the SRVAE framework is more general-purposed, and can be combined with many different latent variable models. Further examples appear in Appendix E.

3.2 Structured Recognition with Latent Gaussian Mixture Model

We instantiate SRVAE with a classical latent variable model, the latent Gaussian mixture model (GMM) [24]. We assume the following generative process (Figure 5a):

$$\begin{aligned} z_t|\pi &\sim \text{Categorical}(z|\pi) \\ \mathbf{h}_t &\sim \mathcal{N}(\mathbf{h}|\boldsymbol{\mu}^{(z_t)}, \boldsymbol{\Sigma}^{(z_t)}) \\ \mathbf{y}_t|\mathbf{h}_t &\sim \mathcal{N}(\mathbf{y}|\boldsymbol{\mu}_{\text{NN}}(\mathbf{h}_t), \boldsymbol{\Sigma}_{\text{NN}}(\mathbf{h}_t)) \end{aligned} \tag{9}$$

This model was studied by Johnson et al. [7], using fully factorised recognition potentials, followed by variational message passing[VMP; 25] to obtain the GMM variational posterior (details can be found in Appendix A). However, the coupled conditional model, parametrised by a neural network, induces posterior correlations between the latent variables that do not exist *a priori*. Hence in order to implement the structured amortised inference, we assume a full covariance Gaussian recognition potential on $\mathbf{h}$, and the SRVAE-GMM variational approximation is computed with VMP given the recognition and prior factors.

3.3 Structured Recognition Variational Autoencoding Nonlinear GPFA

The GPFA model (3) incorporates a generalised-linear likelihood, with the link function $g(\cdot)$ acting separately on each affine embedding value h_n. We are now in a position to use the SRVAE framework to extend GPFA to include a DGM likelihood, greatly increasing the expressiveness of the generative model. Consider observations $\{(x_t, \mathbf{y}_t)\}_{t=1}^T$, defining $\mathbf{f}_t \in \mathbb{R}^K$ and $\mathbf{h}_t \in \mathbb{R}^N$ to be the corresponding vectors of latent process values and embeddings at x_t. [In neural applications, inputs x_t are usually taken to be the timestamp t; see (3). Here we consider the more general case of arbitrary inputs.] In the generative model, we retain the affine mapping from $\mathbf{f}_t$ to $\mathbf{h}_t$, replacing the link function by a non-linear multivariate DGM: $\mathbf{y}_t \sim p(\mathbf{y}|g(\mathbf{h}_t, \gamma))$. Although the generative affine mapping could, in principle, be subsumed within the general deep network, the embeddings $\mathbf{h}_t$ will play a valuable role in parametrising structured recognition.

For amortised svGP inference, our goal is to define a variational distribution $q(\mathbf{U}|\mathbf{Y}, \phi)$. Ashman et al. [10] have considered a similar latent GP model called the SGP-VAE. There, the variational distribution over the latents was found by combining the prior GP distribution with an amortised approximation of the likelihood function factored over latent processes [22]:

$$q(\mathbf{F}, \mathbf{U}) = \prod_k p(\mathbf{f}^k|\mathbf{u}_k) q(\mathbf{u}_k), \text{ where } q(\mathbf{u}_k) \propto p(\mathbf{u}_k) \prod_t r(\mathbf{u}_k|\mathbf{y}_t, x_t, \mathbf{z}_k), \tag{10}$$

The graphical model for SGP-VAE inference corresponds to that of Figure 1b, with $\mathbf{C} = \mathbf{I}$ and $\mathbf{d} = \mathbf{0}$. While this approach captures induced correlation amongst the inducing points, it fails to capture correlations *between* latent processes that arise in the posterior through "explaining away", potentially leading to sub-optimal inference and learning.

Our solution is to recast the output of the amortised inference to the likelihood of $\mathbf{h}$ rather than $\mathbf{U}$, and propose the following structured variational distribution.

$$q(\mathbf{F}, \mathbf{U}) = \left[\prod_k p(\mathbf{f}^k|\mathbf{u}_k)\right] q(\mathbf{U}), \text{ where } q(\mathbf{U}) \propto \int d\mathbf{H}\, p(\mathbf{U}) p(\mathbf{H}|\mathbf{U}) \prod_t r(\mathbf{h}_t|\mathbf{y}_t) \tag{11}$$

where $r(\mathbf{h}_t|\mathbf{y}_t) \propto \mathcal{N}(\mathbf{h}_t|\mathbf{m}(\mathbf{y}_t;\phi), \boldsymbol{\Psi}(\mathbf{y}_t;\phi)) = \mathcal{N}\big(\boldsymbol{\mu}_t^h, \boldsymbol{\Psi}_t^h\big)$, with $\boldsymbol{\Psi}_t^h$ diagonal, and $\mathbf{H} = [\mathbf{h}(x_1), \ldots, \mathbf{h}(x_T)]$. Given the linear-Gaussian relationship between $\mathbf{U}$ and $\mathbf{h}$ (Fig 1a), the integral in (11) can be computed in closed form. Furthermore, even though the recognition potential on $\mathbf{h}$, $r(\mathbf{h}|\mathbf{y}_t)$, is assumed to be fully factorised over the dimensions of each $\mathbf{h}_t$, the variational distribution on $\mathbf{U}$ includes coupling between latent processes induced by the affine mixing coefficients. Hence $q(\mathbf{U})$ captures the correlations both between the latent processes and through time (through combination with the GP prior under the structured autoencoding formulation).

Specifically, given the general expression of the GPFA generative model and the sparse approximation with inducing points, the linear Gaussian relationship between $\mathbf{h}$ and $\mathbf{U}$ at any x is given by

$$p(\mathbf{h}|\mathbf{U}, x) = \mathcal{N}\big(\mathbf{C}\mathfrak{F}(x)\mathbf{U} + \mathbf{d}, \mathbf{C}(\mathbf{K}_x - \mathfrak{F}(x)\mathbf{K}_{\mathbf{U}}\mathfrak{F}(x)^T)\mathbf{C}^T\big), \tag{12}$$

where

$$\mathfrak{F}(x) = \begin{bmatrix} \mathfrak{F}_1(x) & & \\ & \ddots & \\ & & \mathfrak{F}_K(x) \end{bmatrix}, \quad \mathbf{K_U} = \begin{bmatrix} \mathbf{K}_{\mathbf{z}_1\mathbf{z}_1} & & \\ & \ddots & \\ & & \mathbf{K}_{\mathbf{z}_K\mathbf{z}_K} \end{bmatrix}$$

with $\mathbf{K}_x = \text{diag}[\kappa_\theta^k(x,x)]$. Combining this result with the fully factorised recognition potential on $\mathbf{h}$ at x_t (11), we obtain the structured variational distribution on $\mathbf{U}$:

$$q(\mathbf{U}) = \mathcal{N}(\mathbf{m_U}, \mathbf{S_U}) \propto p(\mathbf{U}) \prod_t \mathcal{N}(\mathbf{C}\mathfrak{F}(x_t)\mathbf{U}|\boldsymbol{\mu}_t^h, \boldsymbol{\Psi}_t^h), \text{ with} \tag{13}$$

$$\mathbf{S_U}^{-1} = \mathbf{K_U}^{-1} + \sum_t \mathfrak{F}(x_t)^T \mathbf{C}^T (\boldsymbol{\Psi}_t^h)^{-1} \mathbf{C}\mathbf{F}(x_t), \quad \mathbf{m_U} = \mathbf{S_U}\left(\sum_t \mathfrak{F}(x_t)^T \mathbf{C}^T (\boldsymbol{\Psi}_t^h)^{-1}(\boldsymbol{\mu}_t^h - \mathbf{d})\right).$$

This variational posterior on $\mathbf{U}$ leads to a corresponding posterior on $\mathbf{h}$ at each x:

$$q(\mathbf{h}(x)) = \int d\mathbf{U}\, p(\mathbf{h}|\mathbf{U}) q(\mathbf{U}) = \mathcal{N}\big(\mathbf{C}\mathfrak{F}(x)\mathbf{m_U} + \mathbf{d}, \mathbf{C}(\mathbf{K}_n + \mathfrak{F}(x)(\mathbf{S_U} - \mathbf{K_U})\mathfrak{F}(x)^T)\mathbf{C}^T)\big).$$

and we write $q(\mathbf{h}(x_t)) = \mathcal{N}\big(\mathbf{m}_t^h, \mathbf{S}_t^h\big)$. Then the complete (reparametrised) Monte Carlo estimate of the free energy objective given a mini-batch of data, $\{(x_b, \mathbf{y}_b)\}_{b=1}^B$, takes the following expression.

$$\mathcal{F}(\theta, \gamma, \phi, \mathbf{C}, \mathbf{d}, \mathbf{Z}) = \sum_{b=1}^B \frac{1}{S} \sum_{s=1}^S \log p(\mathbf{y}_b | \mathbf{m}_b^h + \mathbf{L}_b^h \epsilon_s) - \mathbf{KL}[q(\mathbf{U}) \| p(\mathbf{U})] \tag{13}$$

where $\epsilon_s \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, θ is the set of kernel parameters, $\mathbf{L}_b^h$ is the lower-triangular Cholesky component of $\mathbf{S}_b^h$ such that $\mathbf{S}_b^h = \mathbf{L}_b^h (\mathbf{L}_b^h)^T$.

We note that we have retained the scalability of amortised inference by choosing an amortised diagonal-Gaussian potential on $\mathbf{h}$. However, the linear-Gaussian relationship between the inducing points $\mathbf{U}$ and $\mathbf{h}(x)$ (12) leads to a full-covariance variational Gaussian approximation for $\mathbf{U}$, allowing amortised inference to capture the observation-induced posterior correlations between latent processes (commonly referred to as the "explaining away" effect). Thus, the retention of the affine GPFA mapping introduces a key extenstion to SGP-VAE. We will refer to the new model as *structured recognition non-linear GPFA* (SR-nlGPFA). The generative model corresponds to that of svGPFA (Figure 1a) but with the nonlinearity $g(\cdot)$ generalised to a flexible form modelled by a neural network. The complete inference procedure for SR-nlGPFA is graphically illustrated in Figure 1c.

The implicit computation of the variational distribution over $\mathbf{U}$ given the recognition potentials on $\mathbf{h}$ also makes it possible to carry out svGP inference with changed inducing locations (13). This is particularly useful in situations that require inference over test datasets of different durations to those seen in training. In particular, it may be possible to learn an amortised inference network using short sub-sequences drawn from a longer dataset, and then infer a posterior over latent GPs for the complete data sequence efficiently by optimising the placement of inducing points along its full length. See further details on the free-form svGP inference step in Appendix F.2.

4 Related work

Structured Deep Generative Models. A number of studies have considered latent graphical structure within the DGM framework [26, 27, 7, 28–30]. Our work builds on and generalises the prominent SVAE proposal of Johnson et al. [7]. Both SVAE and SRVAE combine the structure of a prior PGM with the flexibility of neural network-based recognition. The difference lies in the form of the recognition potentials. These were taken to extend over single latent variables in the earlier study. Here, we consider joint potentials that capture dependence induced by the generative likelihood. In a closely related study, Lin et al. [28] proposed a structured inference network, which approximates the variational distribution as the combination of the recognition potential and a separate structured latent distribution independent of the prior, the independent structured latent distribution can have non-conjugate factors to improve the expressiveness of the model.

svGPFA and Extensions. Since its introduction as a model to identify linear low-dimensional struction in neural population data [11], GPFA has been extended to incorporate non-linear link

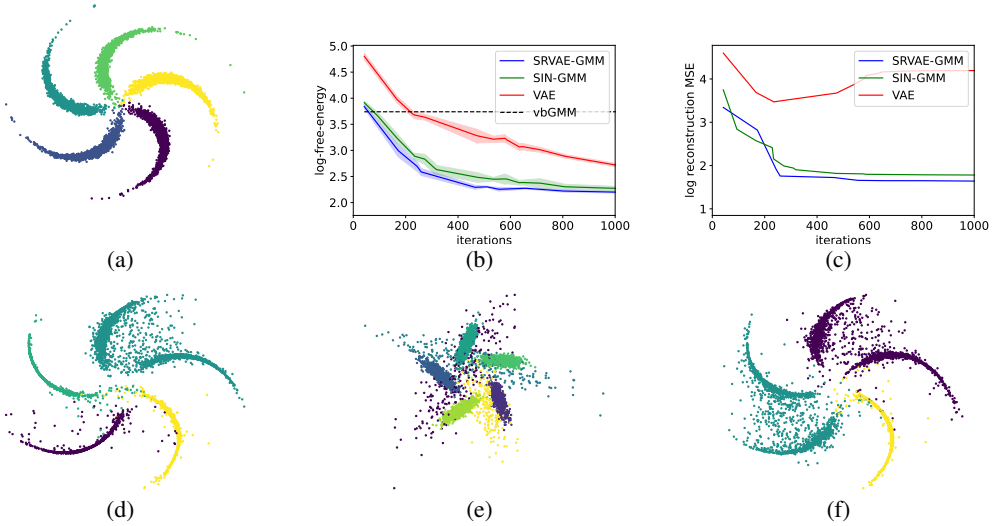


Figure 2: **Empirical Evaluations on Pinwheel Dataset.** (a) Sampled data from the pinwheel dataset; Training curves of the (b) variational free energy objective and (c) testing reconstruction MSE through out training (all in log-scale); Sampled data given trained model of (d) SRVAE-GMM; (e) variational GMM; (f) SIN-GMM [28]. All evaluations are based on the averages over 5 random seeds.

functions and non-conjugate (notably Poisson count or point-process) noise models, and combined with sparse variational inference for efficiency [23, 31, 12, 32]. Related work has considered GP latents in the context of DGMs, particularly using variational autoencoding [9, 33, 10]. In a sense, SR-nlGPFA combines both approaches. It incorporates multiple latent GPs (as in GPFA and some DGM GP models) with an affine mapping that feeds into a nonlinear DGM. However, this structure means that the affine map contributes little to the generative process. Instead it provides a target for amortised inference, which combines with sparse variational GP inference to yield a full structured posterior on inducing point values. As such, the affine map may is better seen as an element of the structured recognition model.

5 Results

We evaluated SRVAE methods for both latent mixtures, and nlGPFA, comparing to relevant baselines on synthetic and real datasets. Empirical results on the instantiations of the SRVAE framework with other latent variable models can be found in Appendix F[1].

5.1 Experiments with SRVAE-GMM

To assess the empirical performance of structured amortisation in a standard latent-variable model, we evaluated SRVAE-GMM on the classic pinwheel dataset (Figure 2a) [7]. We compare SRVAE-GMM with the structured inference network [SIN; 28], which is a more flexible instantiation of the SVAE framework. From Figure 2, we observe that SRVAE-GMM outperforms SIN-GMM in terms of the training variational free energy, reconstruction mean-squared error, and generation fidelity. The models are identical apart from the recognition stage, providing support for the idea that by capturing the "explaining away" posterior factors, structured recognition leads to better training and stronger generation.

[1]Python implementation can be found at `https://github.com/changmin-yu/structured-recognition-neurips-2022`

	Synthetic data		EEG		Population spiking
	SMSE	NLL	SMSE	NLL	SMSE
SR-nlGPFA	$\mathbf{0.31 \pm 0.02}$	$\mathbf{1.69 \pm 0.04}$	$\mathbf{0.27 \pm 0.05}$	$\mathbf{1.81 \pm 0.21}$	$\mathbf{0.47 \pm 0.18}$
SGP-VAE [10]	0.36 ± 0.05	1.76 ± 0.07	0.35 ± 0.05	2.17 ± 0.17	0.55 ± 0.19
Vanilla-VAE [1, 2]	1.05 ± 0.02	13.60 ± 3.58	0.57 ± 0.09	3.45 ± 0.87	3.19 ± 1.98
SVAE-LDS [7]	0.933 ± 0.02	9.37 ± 1.94	3.04 ± 0.38	11.71 ± 2.66	2.52 ± 1.31

Table 1: Quantitative comparison of performance between SR-nlGPFA and baseline models on synthetic dataset, EEG dataset and population spiking dataset (Section 5.2.2) with respect to standarised mean squared error (SMSE) and negative log-likelihood (NLL). Averages over 5 random seeds.

5.2 Experiments with SR-nlGPFA

5.2.1 Synthetic and EEG Dataset

We compared SR-nlGPFA to methods that do not capture the inter-latent posterior correlation, using synthetic data and a small-scale EEG dataset used previously [34, 10]. Unless stated otherwise, we employed an exponentiated quadratic kernel, $\kappa(x, x') = \lambda \exp(-\frac{||x-x'||^2}{\tau^2})$, where λ and τ are the marginal variance and length-scale parameters, respectively.

Baselines Our main baseline model is SGP-VAE [10] (described in Sections 3.3 and 4). We also compared to a "vanilla" VAE with fully factorised Gaussian variational distribution [1, 2], and SVAE with a linear dynamical system (SVAE-LDS) latent prior [7]. All models were implemented with the same recognition and generative network architectures (see Appendix G for implementation details).

Synthetic Dataset We generated data from the GPFA generative model (Eq. 3), with $g(\cdot) = \sigma(\Phi(\cdot))$, where $\Phi(\cdot)$ represents the functional mapping through a fixed, randomly-initialised 2-layer MLP with ReLU hidden non-linearity, and $\sigma(\cdot)$ is the sigmoid function.

EEG Dataset We follow the experimental procedure described by Requeima et al. [34], and consider an EEG measurement dataset spanning 1 second at 256 Hz sample frequency, taken during image viewing [35]. Each datapoint consists of the voltage readings from 7 electrodes positioned on the participant's scalp. Here we report results when all data are observed, contrary to the settings in [34, 10] (we include results and further discussion on partial observability in Appendix F.2).

SR-nlGPFA achieved lower standardised mean squared error (SMSE) and lower test negative log-likelihood (NLL) than the alternative methods on both data sets (Table 1), providing evidence for the benefits of structured recognition. The gains come despite the computational complexity of SR-nlGPFA being of the same order as SGP-VAE (see appendix F.2).

5.2.2 Population Neuronal Firing Data

The firing of place cells in Hippocampal area CA1 and grid cells in the medial Entorhinal Cortex (mEC) is known to be modulated by the animal's location [36, 37], speed and direction of locomotion [38–40], with the mapping between these behavioural covariates and neural activity expressed in non-linear mixed tuning curves. The behavioural signals are continuous and often mutually dependent, and so create temporal and spatial structure in the time series of population activity. We asked whether SR-nlGPFA and related methods would be able to identify and extract this structure without supervision; that is, without direct access to the behavioural covariates.

We used single-cell spiking data from neurons in the hippocampal CA1 and mEC regions of rats recorded during exploration of a Z-shaped track, as reported by Ólafsdóttir et al. [41]. The data comprised 28 experimental sessions, each spanning 10 minutes. Example neural firing patterns of the population are shown in Figure 11 of the Appendix.

For SR-nlGPFA and SGP-VAE, we adopted the GPFA generative model Eq. 3 with DGM non-linearity and Poisson observation likelihood.

$$p(\mathbf{y}(x)|\mathbf{h}(x)) = \prod_{n=1}^{N} \text{Poisson}(y_n(x)|g(\mathbf{h}(x))_n) \tag{14}$$

where x are discrete times, N is the number of recorded neurons, $g(\cdot)$ represents the neural-network mapping from the GPFA features, $\mathbf{h}$, to the rate of Poisson-distributed firing counts for each neuron over contiguous 100 ms bins. SR-nlGPFA obtained a lower SMSE of prediction than SGP-VAE and other methods tested (Table 1).

Computational constraints meant that the SR-nlGPFA model was trained using short batches of data and 64 inducing points per batch. Latent trajectory estimates derived from such batches will not necessarily be continuous at the boundaries between them. Thus, once the model was fit, we performed svGP inference over complete sessions with increased numbers of inducing points (see Section 3.3 and Appendix F.3.2).

To relate these recovered latent time-series to behavioural covariates, we performed two-dimensional Canonical Correlation Analysis [CCA; 42]. We present results for one session here, emphasising qualitative effects. Further qualitative and quantitative results across sessions can be found in Appendix F.3.2.

Figure 3a shows a heatmap of the correlation coefficients between the canonical correlates of the posterior means ($CCX\{1, 2\}$) and the individual behavioural covariates (distance from one end of the track, speed, direction of travel, head direction, and 'unfolded' position along a full lap of the track). Many correlations are high, indicating the low-dimensional manifold parametrised by the conjunctive set of behavioural correlates can be accurately captured by the posterior latent variables learned with SR-nlGPFA solely from neural spikes. Please refer to Appendix Figure 13 for numerical values of the correlations (and for other sessions).

To see whether the learned latent structure contained decodable information about behavioural covariates, we extracted the direction-modulated neurons predicted by the trained model (details in Appendix F.3.2). Figure 3b compares the direction modulation of the model-predicted neurons against that of the rest of the neurons, where direction modulation is defined as the correlation between raw single-cell spike counts and the direction values. Neurons predicted by the model to be direction modulated exhibited significantly greater direction modulation than the other neurons (p-value=5.56×10^{-4}) [43]. Figure 3c shows the firing profile for travel in each direction for the neuron with the strongest predicted direction modulation. The pattern of firing fields exhibits clear directional dependence. Similar comparisons hold for spatial and speed modulation, indicating that SR-nlGPFA is able, in a purely unsupervised fashion, to learn a latent space that contains linearly decodable information about the behavioural covariates associated with individual neuronal firing, even for neurons whose activity exhibits conjunctive coding.

Figure 3d shows the two canonical correlates obtained from the latent trajectories as a function of the animal's spatial location. These reflect both location and direction of movement (colours), indicating the latent dimensions learned by SR-nlGPFA disentangle direction from spatial location in a simple linear projection, despite the conjunctive coding mechanism exhibited in the CA1 neurons. A similar plot derived from SGP-VAE (Figure 14b), shows less clear disentanglement.

Figure 3e shows the temporal evolution of the latent CCs for SR-nlGPFA and SGP-VAE, as well as the spatial location of the rat. Both $CCX1$ and $CCX2$ of SR-nlGPFA consistently track the dynamics of the distance (with perfect phase alignment and half-cycle offset, respectively), whereas neither $CCX1$ or $CCX2$ of SGP-VAE could accurately reflect the distance. Moreover, given the additional svGP inference step over the entire trajectory (with increased number of inducing points), latent signals of SR-nlGPFA exhibit significantly greater smoothness than that of SGP-VAE.

6 Discussion

We have developed a framework of structured recognition for variational autoencoders, which can be viewed as a strict generalisation of the SVAE [7], capturing posterior correlations induced by the "explaining away" effect. Correspondingly, the SRVAE framework yields a tighter bound on the log-likelihood than the SVAE free energy objective.

Applying structured recognition and non-linear variational autoencoding to the GPFA framework, we introduce SR-nlGPFA, facilitating scalable free-form variational Gaussian approximation. SR-nlGPFA supports a fully non-linear, non-conjugate extension to standard GPFA [11, 23, 12], which is particularly well-suited to settings where posterior inference must be scaled to additional data. We show that SR-nlGPFA outperforms the baseline on all presented tasks, both quantitatively and

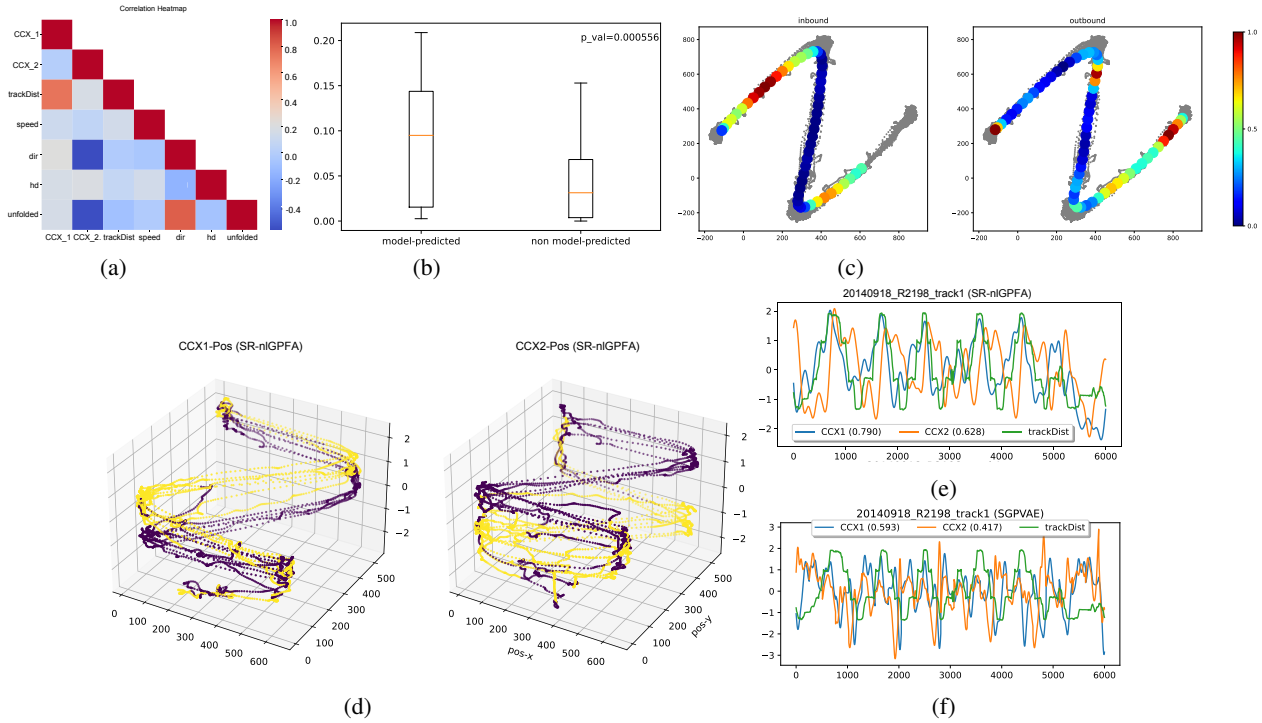


Figure 3: **Analysis of SR-nlGPFA posterior latent structure**. **(a)** Heatmap of the correlation between the CCs and the behavioural covariates for the selected session; **(b)** Directional-modulation comparison between the model-predicted direction-modulated neurons against other neurons; **(c)** Exemplary firing fields of the model-predicted direction-modulated neuron during inbound v.s. outbound movements; **(d)** Plot of $CCX\{1,2\}$ of SR-nlGPFA posterior latents against spatial location, color indicates direction of movement (yellow: inbound, magenta: outbound); **(e-f)** Temporal evolution of (standardised values of) $CCX\{1,2\}$ and spatial location for SR-nlGPFA and SGP-VAE.

qualitatively. Studying population spiking data of hippocampal neurons, we show that SR-nlGPFA is able to identify latent signals that exhibit strong correlation with the behavioural covariates that are well-known to be conjunctively encoded by the recorded neurons, but without direct access to these covariates. While here we focused on temporal correlation exhibited in time-series data, we note that the sparse amortised variational approximation presented here can also be straightforwardly applied to other GP-latent models, such as GP-LVM [44, 45]. While we have considered spiking data alone, these could be substituted by or combined with EEG, calcium imaging, or indeed behavioural data recorded during the same session. Such joint analysis of latent structures given different recordings within the same session could potentially facilitate new findings with respect to, e.g., replay, phase coding, etc.

7 Acknowledgement

We thank the anonymous reviewers for helpful comments and discussions, and Zilong Ji and Dan Bush for helpful discussions regarding the experiments with the neural data. This work is funded by the UKRI, the Gatsby Charitable Foundation, the Simons Foundation and the Wellcome Trust.

References

[1] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[2] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[4] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. In *International Conference on Machine Learning*, pages 1462–1471. PMLR, 2015.

[5] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. *arXiv preprint arXiv:1606.05328*, 2016.

[6] Martin J Wainwright and Michael Irwin Jordan. *Graphical models, exponential families, and variational inference*. Now Publishers Inc, 2008.

[7] Matthew J Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. Composing graphical models with neural networks for structured representations and fast inference. *Advances in neural information processing systems*, 29:2946–2954, 2016.

[8] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.

[9] Francesco Paolo Casale, Adrian Dalca, Luca Saglietti, Jennifer Listgarten, and Nicolo Fusi. Gaussian process prior variational autoencoders. *Advances in neural information processing systems*, 31, 2018.

[10] Matthew Ashman, Jonathan So, Will Tebbutt, Vincent Fortuin, Michael Pearce, and Richard E Turner. Sparse Gaussian process variational autoencoders. *arXiv preprint arXiv:2010.10177*, 2020.

[11] Byron M Yu, John P Cunningham, Gopal Santhanam, Stephen Ryu, Krishna V Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Advances in neural information processing systems*, 21, 2008.

[12] Lea Duncker and Maneesh Sahani. Temporal alignment and latent Gaussian process factor inference in population spike trains. *Advances in Neural Information Processing Systems*, 31, 2018.

[13] Hagai Attias. Inferring parameters and structure of latent variable models by variational bayes. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 21–30, 1999.

[14] Lawrence Saul and Michael Jordan. Exploiting tractable substructures in intractable networks. *Advances in neural information processing systems*, 8, 1995.

[15] Matthew D Hoffman and David M Blei. Structured stochastic variational inference. In *Artificial Intelligence and Statistics*, pages 361–369, 2015.

[16] David M Blei and Michael I Jordan. Variational inference for dirichlet process mixtures. *Bayesian analysis*, 1(1):121–143, 2006.

[17] Dustin Tran, Rajesh Ranganath, and David M Blei. The variational gaussian process. *arXiv preprint arXiv:1511.06499*, 2015.

[18] Yingzhen Li, Richard E Turner, and Qiang Liu. Approximate inference with amortised mcmc. *arXiv preprint arXiv:1702.08343*, 2017.

[19] Dong Bok Lee, Dongchan Min, Seanie Lee, and Sung Ju Hwang. Meta-gmvae: Mixture of gaussian vae for unsupervised meta-learning. In *International Conference on Learning Representations*, 2020.

[20] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[21] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning.* MIT Press, 2005.

[22] Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Artificial intelligence and statistics*, pages 567–574. PMLR, 2009.

[23] Vincent Adam, James Hensman, and Maneesh Sahani. Scalable transformed additive signal decomposition by non-conjugate gaussian process inference. In *2016 IEEE 26th international workshop on machine learning for signal processing (MLSP)*, pages 1–6. IEEE, 2016.

[24] Christopher M Bishop. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[25] John Winn, Christopher M Bishop, and Tommi Jaakkola. Variational message passing. *Journal of Machine Learning Research*, 6(4), 2005.

[26] Rahul G Krishnan, Uri Shalit, and David Sontag. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

[27] Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

[28] Wu Lin, Nicolas Hubacher, and Mohammad Emtiyaz Khan. Variational message passing with structured inference networks. *arXiv preprint arXiv:1803.05589*, 2018.

[29] Chongxuan Li, Max Welling, Jun Zhu, and Bo Zhang. Graphical generative adversarial networks. *Advances in neural information processing systems*, 31, 2018.

[30] Max Paulus, Dami Choi, Daniel Tarlow, Andreas Krause, and Chris J Maddison. Gradient estimation with stochastic softmax tricks. *Advances in Neural Information Processing Systems*, 33:5691–5704, 2020.

[31] Yuan Zhao and Il Memming Park. Variational latent Gaussian process for recovering single-trial dynamics from population spike trains. *Neural Computation*, 29(5):1293–1316, 2017.

[32] Stephen Keeley, David Zoltowski, Yiyi Yu, Spencer Smith, and Jonathan Pillow. Efficient non-conjugate gaussian process factor models for spike count data using polynomial approximations. In *International Conference on Machine Learning*, pages 5177–5186. PMLR, 2020.

[33] Alex Campbell and Pietro Liò. tvgp-vae: Tensor-variate gaussian process prior variational autoencoder. *arXiv preprint arXiv:2006.04788*, 2020.

[34] James Requeima, William Tebbutt, Wessel Bruinsma, and Richard E Turner. The gaussian process autoregressive regression model (gpar). In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1860–1869. PMLR, 2019.

[35] Xiao Lei Zhang, Henri Begleiter, Bernice Porjesz, Wenyu Wang, and Ann Litke. Event related potentials during object recognition tasks. *Brain research bulletin*, 38(6):531–538, 1995.

[36] John O'keefe and Lynn Nadel. *The hippocampus as a cognitive map*. Oxford university press, 1978.

[37] Edvard I Moser, Emilio Kropff, and May-Britt Moser. Place cells, grid cells, and the brain's spatial representation system. *Annu. Rev. Neurosci.*, 31:69–89, 2008.

[38] Bruce L McNaughton, Carol A Barnes, and JJEBR O'Keefe. The contributions of position, direction, and velocity to single unit activity in the hippocampus of freely-moving rats. *Experimental brain research*, 52(1):41–49, 1983.

[39] Francesca Sargolini, Marianne Fyhn, Torkel Hafting, Bruce L McNaughton, Menno P Witter, May-Britt Moser, and Edvard I Moser. Conjunctive representation of position, direction, and velocity in entorhinal cortex. *Science*, 312(5774):758–762, 2006.

[40] Emilio Kropff, James E Carmichael, May-Britt Moser, and Edvard I Moser. Speed cells in the medial entorhinal cortex. *Nature*, 523(7561):419–424, 2015.

[41] H Freyja Ólafsdóttir, Francis Carpenter, and Caswell Barry. Coordinated grid and place cell replay during rest. *Nature neuroscience*, 19(6):792–794, 2016.

[42] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

[43] Francesca Cacucci, Colin Lever, Thomas J Wills, Neil Burgess, and John O'Keefe. Theta-modulated place-by-direction cells in the hippocampal formation in the rat. *Journal of Neuroscience*, 24(38):8265–8277, 2004.

[44] Neil Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. *Advances in neural information processing systems*, 16, 2003.

[45] Anqi Wu, Nicholas A Roy, Stephen Keeley, and Jonathan W Pillow. Gaussian process based nonlinear latent structure discovery in multivariate spike train data. *Advances in neural information processing systems*, 30, 2017.

[46] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[47] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[48] Garoe Dorta, Sara Vicente, Lourdes Agapito, Neill DF Campbell, and Ivor Simpson. Training vaes under structured residuals. *arXiv preprint arXiv:1804.01050*, 2018.

[49] Tom Rainforth, Adam Kosiorek, Tuan Anh Le, Chris Maddison, Maximilian Igl, Frank Wood, and Yee Whye Teh. Tighter variational bounds are not necessarily better. In *International Conference on Machine Learning*, pages 4277–4285. PMLR, 2018.

[50] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[51] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[52] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[53] Edward H Nieh, Manuel Schottdorf, Nicolas W Freeman, Ryan J Low, Sam Lewallen, Sue Ann Koay, Lucas Pinto, Jeffrey L Gauthier, Carlos D Brody, and David W Tank. Geometry of abstract learned knowledge in the hippocampus. *Nature*, 595(7865):80–84, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Details on Structured VAE

The variational free energy for VAE takes a similar formulation as the non-amortised version:

$$\mathcal{F}(\theta, \phi) = \left\langle \log p(y, z|\Theta) - \log q(z|y, \phi) \right\rangle_{q(z|y,\phi)} \tag{15}$$

where we utilise a variational distribution parametrised by a recognition network, $q(z|y, \phi)$. The parametric assumptions in standard VAE formulation limits the expressiveness of the latent space modelling, hence leading sub-optimal training and variational inference. In order to improve the flexibility of both the generative modelling and the variational approximation, Johnson et al. [7] proposed structured VAE, combining PGM-parametrised latent prior distribution with amortised inference such that the flexibility of neural network modelling and the conditional dependence structure can be integrated to derive the variational distribution.

Specifically, the structured autoencoding framework considers the following free energy:

$$\mathcal{F}(\Theta, \lambda) = \left\langle \log \frac{p(y|z, \gamma)p(z|\theta_{\text{PGM}})}{q(z|\lambda)} \right\rangle_{q(z|\lambda)} \tag{16}$$

It is clear to see that the partial optimisation on $q(z|\lambda)$ yields the following (partially) optimal variational approximation:

$$q(z|\lambda^*) \propto p(y|z, \gamma)p(z|\theta_{\text{PGM}}) \tag{17}$$

Due to the apparent intractability, a recognition network is used to generate amortised approximation of the generative likelihood:

$$q(z|\lambda^*) \propto p(y|z, \gamma)p(z|\theta_{\text{PGM}}) \approx r(z|y, \phi)p(z|\theta_{\text{PGM}}) \tag{18}$$

where $r(z|y, \phi)$ is the recognition potential parametrised by ϕ. With conjugate mean-field recognition potential, the resulting $q(z|\lambda^*)$ can hence be computed analytically via VMP, and thus contains the factored structure inherent in the prior PGM, whilst exhibits flexibility due to the neural network parametrised recognition potential. The generative and recognition parameters can be trained following standard stochastic optimisation by maximising the *surrogate* variational free energy:

$$\mathcal{F}_{\text{SVAE}}(\Theta, \phi) = \left\langle \log \frac{p(y|z, \gamma)p(z|\theta_{\text{PGM}})}{q(z|\lambda^*)} \right\rangle_{q(z|\lambda^*)} \tag{19}$$

Note that in the original SVAE formulation, Johnson et al. [7] consider hyperpriors on θ_{PGM}, and are updated given natural gradient descent. Here we assume deterministic θ_{PGM} for simplicity (except for the SRVAE-GMM model), and the structured amortisation framework can be easily extended to adapt to the variational Bayes setting.

B More Instantiations of Structured Recognition Framework with Latent Variable Models

Here we provide two additional instantiations of the SRVAE framework. The corresponding empirical evaluations can be found in Appendix F.

B.1 Tree-Structured Latent PGM

We instantiate the SRVAE framework with discrete tree-structured PGM (Figure 4a), which we term as TreeSRVAE. A general tree-structured PGM takes the following density function.

$$p(z) = \frac{1}{Z} \prod_i \psi_i(z_i) \prod_{(i,j)\in E} \psi_{ij}(z_i, z_j) \tag{20}$$

Non-linear generative likelihood functions introduces joint factors that does not exist a prior, known as “explaining away" (Section 3). Below we provide two potential solutions

The optimal variational distribution should have the same PGM structure as the posterior, which often contains joint factors over many (possibly all) latent variables. However, as the number of states grows exponentially with the number of latent variables, in practice it is usually not scalable

to have amortised inference output a joint factor that approximates the exact posterior joint factor potentials. We hence seek scalable alternatives. Here we design the recognition network to output the amortised factor potential to be of the same structure as the prior PGM; in this case, a tree-structured distribution with the same set of nodes and pairwise connections (Figure 4b).

$$r(z|y;\phi) = \prod_i \xi_i(z_i) \prod_{(i,j)\in E} \xi_{ij}(z_i, z_j)$$

Note that we do not require the recognition network to output a "proper" density function, but only the factored potentials up to some normalising constant (singleton and pairwise in TreeSRVAE). By utilising a amortised potential of the same structure of the prior distribution, we could easily perform the partial optimisation step of combining the recognition output with the prior distribution, by performing updates only on the existing factored structures.

$$\begin{aligned} &q^*(z|x,\theta,\phi) \propto \prod_i \psi'_i(z_i) \prod_{i,j} \psi'_{ij}(z_i, z_j) \\ &\text{where } \psi'_i(z_i) = \psi_i(z_i)\xi_i(z_i), \forall i, \quad \text{and } \psi'_{i,j}(z_i, z_j) = \psi_{i,j}(z_i, z_j)\xi_{i,j}(z_i, z_j), \forall (i,j) \end{aligned} \tag{21}$$

The training procedure follows the general VAE formulation, i.e., we need to generate reparametrised samples from the variational distribution to compute the Monte Carlo estimate of the free energy objective. Thus far we only have the density function up to the normalising constant. Given the tree-structured posterior latent variables, the normalising constant can be computed exactly with Belief Propogation (BP; [6]). At each iteration k, the message propagation between variable i and factor a are given by

$$\begin{aligned} \mu^{(k)}_{i\to a}(z_i) &= \prod_{c\in\mathcal{N}(i)\setminus a} \mu^{(k-1)}_{c\to i}(z_i) \\ \mu^{(k)}_{a\to i}(z_i) &= \sum_{\mathbf{x}_a\setminus x_i} \psi_a(z_a) \prod_{j\in\mathcal{N}(a)\setminus i} \mu^{(k-1)}_{j\to a}(z_j) \end{aligned} \tag{22}$$

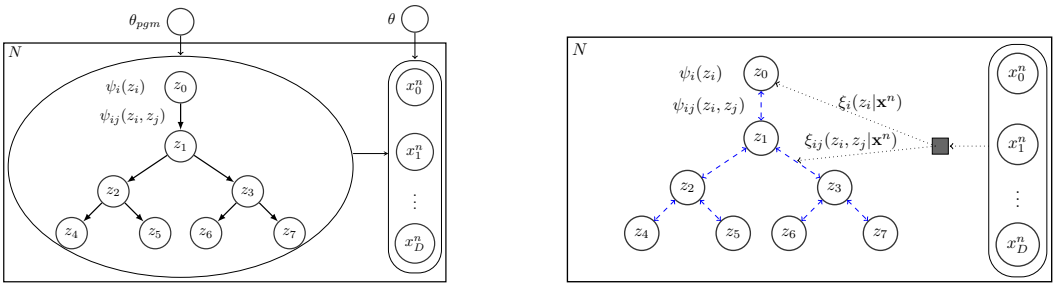
Due to the tree structure (i.e., no loops), belief propagation converges after a single inward-outward pass (similar to forward-backward propagation in HMM models). Upon convergence, we can compute the marginal distributions of the variables and factors as products of messages and factor potentials.

$$\begin{aligned} b_i(z_i) &\overset{+c}{=} \prod_{c\in\mathcal{N}(i)} \mu^*_{c\to i}(z_i) \\ b_a(z_a) &\overset{+c}{=} \psi_a(z_a) \prod_{i\in\mathcal{N}(a)} \mu^*_{i\to a}(z_i) \end{aligned} \tag{23}$$

The singleton and pairwise marginal beliefs can be used to sample from the relevant components of $q^*(z|x,\theta,\phi)$. Due to the tree-structure, the order of sampling the latent variables is irrelevant. That is, we can initiate ancestral sampling from an arbitrary node i in the tree, computing the conditional distribution $q^*(z_j|z_i)$ from the relevant marginals. To sample categorical latent variables we employ the Gumbel-Softmax relaxation [46, 47] so that gradients of the expection can be back-propagated through the samples.

The free energy then takes the following form.

$$\mathcal{F}(\phi,\theta,\gamma) = \mathbb{E}_{q^*(z|x,\theta,\phi)}[\log p(y|z,\gamma)] - \mathrm{KL}[q^*(z|x,\theta,\phi)||p(z,\theta)]$$



(a) (b)

Figure 4: **SRVAE framework implemented with tree-structured latent variable model. (a)** generative model **(b)** inference model, with tree-structured (or a joint factor over all latent variables) recognition potentials.

Given the tree structure, we can compute the KL-divergence analytically with the singleton and pairwise beliefs computed with BP.

$$
\begin{aligned}
\mathrm{KL}[q(\mathbf{z})||p(\mathbf{z})] &= \sum_{\mathbf{z}} q(\mathbf{z}) \log q(\mathbf{z}) - \sum_{\mathbf{z}} q(\mathbf{z}) \log p(\mathbf{z}) \\
&= \sum_{\mathbf{z}} q(\mathbf{z}) \log \left(\frac{\prod_{(i,j)} q_{ij}(z_i, z_j)}{\prod_i q_i(z_i)^{(d_i-1)}} \right) - \sum_{\mathbf{z}} q(\mathbf{z}) \log \left(\frac{\prod_{(i,j)} p_{ij}(z_i, z_j)}{\prod_i p_i(z_i)^{(d_i-1)}} \right) \\
&= \sum_{i,j} \sum_{\mathbf{z}} q(\mathbf{z}) \log q(z_i, z_j) - \sum_i (d_i - 1) \sum_{\mathbf{z}} q(\mathbf{z}) \log q(z_i) \\
&\quad - \sum_{i,j} \sum_{\mathbf{z}} q(\mathbf{z}) \log p(z_i, z_j) + \sum_i (d_i - 1) \sum_{\mathbf{z}} q(\mathbf{z}) \log p(z_i) \\
&= \sum_{i,j} \sum_{z_i, z_j} q(z_i, z_j) \log q(z_i, z_j) - \sum_i (d_i - 1) \sum_{z_i} q(z_i) \log q(z_i) \\
&\quad - \sum_{i,j} \sum_{z_i, z_j} q(z_i, z_j) \log p(z_i, z_j) + \sum_i (d_i - 1) \sum_{z_i} q(z_i) \log p(z_i) \\
&= \sum_{i,j} \mathrm{KL}[q(z_i, z_j)||p(z_i, z_j)] - \sum_i (d_i - 1) \mathrm{KL}[q(z_i)||p(z_i)]
\end{aligned}
\tag{24}
$$

We note that it is also possible to use other dependency structures of the amortised factor potentials, such as a joint factor potential over all latent variables, or to employ latent structure discovery techniques to infer the latent dependency structure as well as the variational parameters [13].

B.1.1 Gaussian Factors in Tree-Structured Latent PGM

We note that treeSRVAE can also be implemented with continuous latent variables, and here we provide a simple example with Gaussian-distributed latent variables for illustration (note that the joint distribution is also Gaussian). When working with continuous latent variables, summation is replaced with integration in the message passing steps (Eq. 22). Since Gaussian distributions are fully parametrised by the natural parameters (same holds for other exponential family distributions), we are able to perform variational message passing by only propagating the messages of the sufficient statistics [25]. Assume the latent prior distribution specified by a Gaussian MRF has a density function of the following format.

$$
\begin{aligned}
p(\mathbf{z}) &= \frac{1}{Z} \exp\left(-\frac{1}{2} \mathbf{z}^T \mathbf{A} \mathbf{z} + \mathbf{h}^T \mathbf{z} \right) \\
&= \frac{1}{Z} \prod_i \exp\left(-\frac{1}{2} A_{ii}^2 z_i^2 + b_i z_i \right) \prod_{ij} \exp\left(-\frac{1}{2} A_{ij} z_i z_j \right) \\
&= \frac{1}{Z} \prod_i \psi_i(z_i) \prod_{ij} \psi_{ij}(z_i, z_j)
\end{aligned}
\tag{25}
$$

where $\mathbf{A}$ is the precision matrix and $\mathbf{h} = \mathbf{A}\boldsymbol{\mu}$, with $\boldsymbol{\mu}$ being the mean parameter of the joint Gaussian distribution. Note that $\mathbf{A}$ is also known as the information matrix, which specifies the dependency structure of the latent distribution. Assume each message at time k is parametrised by a Gaussian distribution, $m^{(k)}_{i\to j}(x_j) \sim \mathcal{N}(x_j|\mu^{(k)}_{i\to j}, (\lambda^{(k)}_{i\to j})^{-1})$. At each iteration k, the message updates are shown as following.

$$
\begin{aligned}
m^{(k)}_{i\to j}(z_j) &= \int dz_i \psi_{ij}(z_i, z_j)\psi_i(z_i) \prod_{k\in\mathcal{N}(i)\setminus j} m^{(k-1)}_{k\to i}(z_i) = \mathcal{N}(z_j; \mu^{(k)}_{i\to j}, \lambda^{(k)}_{i\to j}) \\
\text{where } \lambda^{(k)}_{i\to j} &= \frac{A^2_{ij}}{\lambda^{(k)}_{i\setminus j}}; \quad \mu^{(k)}_{i\to j} = \frac{A_{ij}\mu^{(k)}_{i\setminus j}}{\lambda^{(k)}_{i\to j}} \\
\text{and } \lambda^{(k)}_{i\setminus j} &= A_{ii} + \sum_{l\in\text{nbr}(i)} \lambda^{(k)}_{l\to i}; \quad \mu^{(k)}_{i\setminus j} = \frac{A_{ii}b_i + \sum_{l\in\text{nbr}(i)} \lambda^{(k)}_{l\to i}\mu^{(k)}_{l\to i}}{\lambda^{(k)}_{i\setminus j}}
\end{aligned}
\tag{26}
$$

Then the singleton and pairwise beliefs (used for sampling and computing the KL divergence) can be computed as following.

$$
\begin{aligned}
b(z_i) &= \psi_i(z_i) \prod_{j\in\text{nbr}(i)} m_{j\to i}(z_i) = \mathcal{N}(z_i|\mu_i, \lambda_i) \\
\text{where } \lambda_i &= A_{ii} + \sum_{j\in\text{nbr}(i)} \lambda_{j\to i}; \quad \mu_i = \frac{A_{ii}b_i + \sum_{j\in\text{nbr}(i)} \lambda_{j\to i}\mu_{j\to i}}{\lambda_i} \\
b(z_i, z_j) &= \psi_{ij}(z_i, z_j)\psi_i(z_i)\psi_j(z_j) \prod_{u\in\text{nbr}(i)} m_{u\to i}(x_i) \prod_{v\in\text{nbr}(j)} m_{v\to j}(x_j) = \mathcal{N}([z_i, z_j]^T|\mathbf{m}_{ij}, \Lambda_{ij}) \\
\text{where } \Lambda_{ij} &= \begin{bmatrix} A_{ii} + \lambda'_{i\setminus j} & \frac{1}{2}A_{ij}, & \frac{1}{2}A_{ji} & A_{jj} + \lambda'_{j\setminus i} \end{bmatrix}, \quad \mathbf{m}_{ij} = \begin{bmatrix} b_i + \lambda'_{i\setminus j}\mu'_{i\setminus j} \\ b_j + \lambda'_{j\setminus i}\mu'_{j\setminus i} \end{bmatrix}, \\
\lambda'_{i\setminus j} &= \sum_{u\in\text{nbr}(i)\setminus j} \lambda_{u\to j}\mu'_{i\setminus j} = \frac{\sum_{u\in\text{nbr}(i)\setminus j} \lambda_{u\to j}\mu_{u\to j}}{\lambda'_{i\setminus j}}
\end{aligned}
\tag{27}
$$

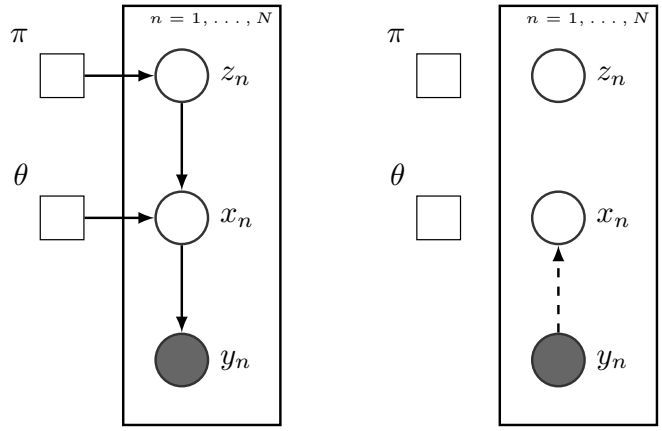
Given the prior tree structure (i.e., the information matrix, A), the recognition network outputs the sufficient statistics (mean and precision matrix) of a Gaussian distribution with the same structure as the tree-based prior distribution, $\mathcal{N}(\mathbf{z}|\mu', (A')^{-1})$. Note that instead of parametrising a full-rank precision matrix, which requires $\mathcal{O}(D^2)$ outputs (D is the number of latent variables), by constraining the dependency structure as the prior distribution, the recognition network only need to output $3D-1$ scalars (for each batch), where $2D-1$ contribute towards the non-zero entries in the precision matrix, and D contribute towards the linear term. Such type of parametrisation is similar to the modelling of off-diagonal elements in the covariance matrix with low-rank decomposition, but we note that the off-diagonal entries in the covariance matrix do not convey information about the dependency structure of the latent variables, hence our framework provides stronger interpretability. Dorta et al. [48] proposes a direct low-rank parametrisation of the precision matrix in a VAE setting, which again lacks interpretation in terms of the latent dependency structure. Moreover, we note that they require the matrix inversion to convert the precision matrix into the covariance matrix for inference and sampling, which requires $\mathcal{O}(D^3)$ complexity, whereas we apply the BP, which on tree-structured PGMs only require $\mathcal{O}(D)$ iterations to converge.

B.2 Gaussian Mixture Model Latent PGM

The PGMs for the generative and inference models of SRVAE-GMM is shown in Figure 5.

C Joint Factor Potentials Induced by Non-Linear Likelihood ("Explaining Away")

Here we show that even a simple non-linear likelihood function can induce non-trivial posterior correlations that do not exist *a priori*. Consider a Bernoulli generative likelihood function, with the



(a) (b)

Figure 5: **SRVAEframework implemented with tree-structured latent variable model. (a)** generative model **(b)** inference model, with tree-structured (or a joint factor over all latent variables) recognition potentials.

logits being an affine transformation of z.

$$p(y|z,\theta) = \mathcal{B}(y;\sigma(Wz+b)) = \prod_i \mathcal{B}(y_i;\sigma(\sum_j W_{ij}z_j + b_i)) \tag{28}$$

where $\theta = \{W, b\}$.

Given this simple generative model, and the tree-structured latent distribution (Eq. 20), we can derive the true posterior distribution analytically.

$$\begin{aligned}
\log p(z|x) &\overset{+c}{=} \log p(z) + \log p(y|z,\theta) \\
&= \sum_i \log \psi_i(z_i) + \sum_{i,j} \log \psi_{i,j}(z_i, z_j) + \\
&\quad \sum_j y_j \log\left(\frac{1}{1+e^{-\sum_i W_{ji}z_i + b_j}}\right) + (1-y_j)\log\left(\frac{e^{-\sum_i W_{ji}z_i+b_j}}{1+e^{-\sum_i W_{ji}z_i+b_j}}\right) \\
&= \sum_i \log \psi_i(z_i) + \sum_{i,j} \log \psi_{i,j}(z_i, z_j) + \\
&\quad \sum_j y_j \left(\sum_i W_{ji}z_i + b_j\right) + \sum_j \log\left(\frac{e^{-\sum_i W_{ji}z_i+b_j}}{1+e^{-\sum_i W_{ji}z_i+b_j}}\right) \\
&\overset{+c}{=} \sum_i \log \phi_i(z_i) + \sum_{i,j} \log \phi_{i,j}(z_i, z_j) + \phi_{\mathbf{z}}(\mathbf{z})
\end{aligned} \tag{29}$$

where $\phi_i(z_i) = \psi_i(z_i)\exp(z_i \sum_j W_{ji}y_j), \phi_{ij}(z_i, z_j) = \psi_{ij}(z_i, z_j), \forall i, j,$

$$\phi_{\mathbf{z}}(\mathbf{z}) = \sum_j \log\left(\frac{e^{-\sum_i W_{ji}z_i+b_j}}{1+e^{-\sum_i W_{ji}z_i+b_j}}\right)$$

We observe that the true posterior distribution involves the singleton and pairwise potentials that preserve the structure of the prior distribution, and at the same time the normalising constant of the conditional likelihood function introduces a joint factor over all latent variables, $\phi_{\mathbf{z}}(\mathbf{z})$, which cannot be captured by the fully factorised amortised potential, but can be partially captured by the tree-structured potentials.

D Further Details of SR-nlGPFA

The derivation of the variational free energy objective for svGPFA (and SR-nlGPFA; Eq. 5) is shown as following Titsias [22].

$$
\begin{aligned}
&\log p(\mathbf{y}) \\
&= \log \iint d\mathbf{f}\, d\mathbf{u}_{1:K}\, p(\mathbf{y}, \mathbf{f}(\cdot), \mathbf{u}_{1:K}) \\
&= \log \iint d\mathbf{f}\, d\mathbf{u}_{1:K}\, p(\mathbf{y}|\mathbf{f}(\cdot)) \prod_k p(f_k(\cdot)|\mathbf{u}_k) p(\mathbf{u}_k|\mathbf{z}_k) \\
&\geq \iint d\mathbf{f}\, d\mathbf{u}_{1:K}\, q(\mathbf{u}_{1:K}, \mathbf{f}(\cdot)) \log \left[\frac{p(\mathbf{y}|\mathbf{f}(\cdot)) \prod_k p(f_k(\cdot)|\mathbf{u}_k) p(\mathbf{u}_k|\mathbf{z}_k)}{q(\mathbf{u}_{1:K}, \mathbf{f})} \right] \\
&= \iint d\mathbf{f}\, d\mathbf{u}_{1:K}\, q(\mathbf{u}_{1:K}, \mathbf{f}(\cdot)) \log \left[\frac{p(\mathbf{y}|\mathbf{f}(\cdot)) \prod_k p(f_k(\cdot)|\mathbf{u}_k) p(\mathbf{u}_k|\mathbf{z}_k)}{[\prod_k p(f_k(\cdot)|\mathbf{u}_k)]\, q(\mathbf{u})} \right] \\
&= \mathbb{E}_{q(\mathbf{h}(\cdot))} \left[\log p(\mathbf{y}|\mathbf{h}(\cdot)) \right] - \int d\mathbf{u}_{1:K}\, q(\mathbf{u}) \log \frac{q(\mathbf{u})}{\prod_k p(\mathbf{u}_k|\mathbf{z}_k)} \\
&= \mathbb{E}_{q(\mathbf{h}(\cdot))} \left[\log p(\mathbf{y}|\mathbf{h}(\cdot)) \right] + \mathcal{H}[q] + \sum_k \mathbb{E}_{q(\mathbf{u}_k)} [\log p(\mathbf{u}_k|\mathbf{z}_k)]
\end{aligned}
\tag{30}
$$

where $\mathcal{H}[q]$ is the entropy of the variational distribution $q(\mathbf{u})$.

E Proof of Proposition 3.1

We re-state the proposition.

Proposition E.1. *Consider the following latent structured prior.*

$$p(z; \theta^0) = \frac{1}{Z} \prod_{c \in C} \psi_c(z_c)$$

where we assume θ^0 is the set of trainable deterministic prior parameter. Consider the free energy objective.

$$\mathcal{F}[q(z)] = \mathbb{E}_{q(z)} \left[\log \frac{p(z|\theta^0) p(x|z, \theta)}{q(z)} \right]$$

Both the SVAE objective and the AEA-objective take the following expression.

$$\mathcal{F}(\theta, \phi) = \mathbb{E}_{q^*(z)} \left[\log \frac{p(z|\theta^0) p(x|z, \theta)}{q^*(z)} \right],$$

For SVAE, $q^*(z)$ is derived through partial optimisation given conjugate amortised inference outputs.

$$q^*(z) = \underset{q^(z)}{\operatorname{argmax}}\, \mathbb{E}_{q(z)} \left[\log \frac{p(z|\theta^0) \prod_i l_c(z_i|x, \phi)}{q(z)} \right],$$

where $\prod_i l_i(z_i|x, \phi)$ represents the approximate local evidence potentials (corresponding to the product of singleton potentials).

For SRVAE, instead of being fully factorised, the recognition network outputs structured factor potentials, which in principle, could contain factors of arbitrary set of latent variables (e.g., a joint factor for all latent variables).

$$q^*(z) = \underset{q^(z)}{\operatorname{argmax}}\, \mathbb{E}_{q(z)} \left[\log \frac{p(z|\theta^0) \prod_{c' \in C^*} r_c(z_c|x; \phi)}{q(z)} \right],$$

Then the SRVAE objective function provides tighter lower bound to the free energy than the SVAE objective function.

$$\max_{q(z)} \mathcal{F}[q(z)] \geq \max_{\phi} \mathcal{F}_{\textit{SRVAE}}(\theta, \phi) \geq \max_{\phi} \mathcal{F}_{\textit{SVAE}}(\theta, \phi)$$

We note that throughout the paper we assume both the SRVAE and SVAE models employ identical generative models, hence we have the stronger results that the inequality for all θ. For SR-nlGPFA, despite introducing the additional affine transformation (parametrised by $\mathbf{C}$ and $\mathbf{d}$) in the generative process, the linear operation can be completely subsumed into the neural-network generative model, hence leading to the identical generative model as SGP-VAE.

Proof. The first inequality is trivial. For the second inequality, it is easy to see that the set of fully factorised amortised potentials, $L = \{l(z)|l(z) \propto \prod_i l_i(z_i|x;\phi)\}$, is a strict subset of the set of all structured amortised potentials, $H = \{h(z|x;\phi)|h(z|x;\phi) \propto \prod_{c\in C^*} r_c(z_c|x;\phi)\}$, where C^* is the set of factors in the amortised approximation to the likelihood. Hence for arbitrary $p(x|z,\theta)$, we have that,

$$\operatorname*{argmin}_{h\in H} \mathbf{KL}[h(z|x,\phi)\|p(x|z,\theta)] \leq \operatorname*{argmin}_{l\in L} \mathbf{KL}\left[\prod_{c\in C} l_c(z_c|x,\phi)\middle\| p(x|z,\theta)\right]$$

with equality if and only if $p(x|z,\theta) \in L$. Hence it is trivial to derive that,

$$\max_{\phi} \mathcal{F}_{\text{SRVAE}}(\theta,\phi) = \mathbb{E}_{q^*_{\text{AEA}}(z)}\left[\log \frac{p(z|\theta^0)p(x|z,\theta)}{q^*(_{\text{AEA}}z)}\right] \geq \mathbb{E}_{q^*_{\text{SVAE}}(z)}\left[\log \frac{p(z|\theta^0)p(x|z,\theta)}{q^*(_{\text{SVAE}}z)}\right] = \max_{\phi} \mathcal{F}_{\text{SVAE}}(\theta,\phi), \forall\theta$$

with equality if and only if $p(x|z,\theta) \in L$. □

We note that having a tighter free energy lower bound does not necessarily lead to stronger generative modelling [49]. However, by exploring an alternative formulation of the free energy objective (relative to Eq. 1):

$$\mathcal{F}[q] = \log p(y) - \mathbf{KL}[q(z)\|p(z|y)] \Rightarrow \max_q \mathcal{F}[q] = \min_q \mathbf{KL}[q(z)\|p(z|y)] \tag{31}$$

we see that a tighter free energy bound leads to smaller KL-divergence between the variational approximation and the ground-truth posterior distribution, hence achieving more accurate posterior inference. This is indeed reflected in our empirical evaluation of SR-nlGPFA on the spiking dataset (that the posterior latent processes is significantly more coherent with the underlying behavioural covariates than that of SGP-VAE [10]), and also reflected by the better generation quality in the bar dataset (F.1).

F Further Experimental Results

F.1 TreeSRVAE on "Bar" Dataset

We first evaluate tree-structured recognition potential on a synthetic "Bar" dataset, consists of square-grid observations with horizontal and vertical bars.

$$\begin{aligned} p(\mathbf{z}|\theta) &= \mathcal{B}(z_0|p_0)\prod_{i>0}\mathcal{B}(z_i|\text{pa}(z_i)), \\ p(\mathbf{y}|\mathbf{z},\mathbf{W},\mathbf{b}) &= \prod_{d=1}^{D^2}\mathcal{B}(y_d|\sigma(\mathbf{W}\mathbf{z}+\mathbf{d})_d) \end{aligned} \tag{32}$$

where $\text{pa}(z)$ denote the parent node of node z, $\sigma(\cdot) = \frac{1}{1+\exp(\cdot)}$ is the sigmoid function, and $\mathcal{B}(p)$ denotes the Bernoulli distribution with rate p. Each binary z_i denotes the activation of one bar in the square-grid (e.g., $z_1 = 1$ indicates the presence of the first horizontal bar). The generative parameters $\mathbf{W}$ and $\mathbf{b}$ are defined as

$$W_{ij} = \begin{cases} 2\times\omega_i & \text{if pixel i is on the bar j} \\ 0 & \text{otherwise} \end{cases}, \quad b_i = -\omega_i, \text{ for } i = 1,\ldots,D^2, \tag{33}$$

where $\{\omega_i\}_{i=1}^{D^2}$ are the temperature parameters that control the degree of randomness in the data generation. Note that in current implementations we set $\omega_i = \omega$ for all i. Exemplary samples from the "Bar" dataset if shown in Figure 6 ($D = 8$).

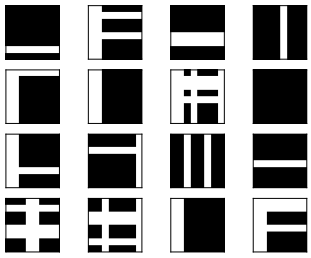
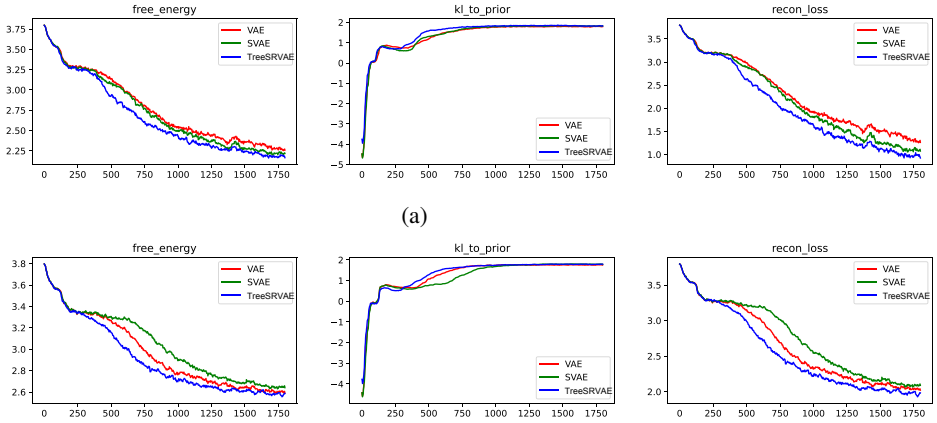


Figure 6: **Samples from the "Bar" dataset**.



(a)

(b)

Figure 7: Evaluation of TreeSRVAE on the bar test experiment. **From Top to Bottom**: Bar-test experiment with (a) sampling temperature of 4; (b) sampling temperature of 10. **From left to right:** free energy; reconstruction loss (log-scale); KL-divergence (with respect to the prior distribution).

Note that here we only perform variational inference, instead of variational Bayes optimisation, hence we assume the parameters of the prior distribution (Eq. 20) to be deterministic quantities (which can still be optimised via gradient descent, but do not follow any natural-gradient updates).

Given the tree-structured prior distribution (Eq. 32), we have that the singleton and pairwise potentials take the following expression.

$$\psi_i(z_i) = \begin{cases} \mathcal{B}(z_0|p^0) & \text{if } i = 0; \\ 1 & \text{if } i > 0; \end{cases} \quad \psi_{ij}(z_i, z_j) = \begin{cases} \mathcal{B}(z_i|p^i_{z_j}) & \text{if } j = \text{pa}(i); \\ 1 & \text{otherwise ;} \end{cases} \tag{34}$$

We use the Gumbel-Softmax trick for relaxed reparametrised sampling of the discrete latent variables. However, we can also apply a hard-transformation to the relaxed samples to generate the binary-valued latent samples as following.

$$z_{hard} = 0.5 \times (\text{sign}(z - 0.5) + 1) \tag{35}$$

In practice we find using the hard-sampling trick improves performance for all models considered (VAE, SVAE, TreeSRVAE) on the "Bar" dataset.

We set the prior distribution to be a "uninformative" prior, where $p^0 = 0.5$ and p^i_{ij} for all (i, j) (Eq. 34).

In Figure 7 we show the training curves of the free energy (Eq. 7), reconstruction loss ($\log p(x|z)$) and KL-divergence (with respect to the prior distribution, $\text{KL}[q^*(z, \phi)||p(z, \theta_{\text{pgm}})]$) for the "Bar" dataset generated with varying sampling temperature (ω). , where the observation is over an 8×8 square grid, hence the latent variables are $z \in \{0, 1\}^{16}$. We evaluate on three models: i) the latent variable follows a fully factorised Bernoulli distribution, and the recognition network outputs mean-field inference, which is equivalent to standard VAE with Bernoulli latents [46]; ii) the latent variable follows a

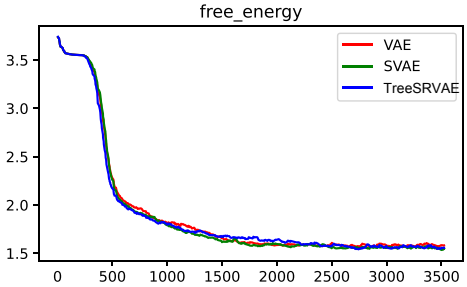


Figure 8: Free energy of the TreeSRVAE, SVAE and VAE over the training process on the "side-dependent" "Bar" dataset (Eq. 36)

distribution characterised by a tree-structured probabilistic graphical model, and the recognition network outputs fully factorised potentials, which is equivalent to the original SVAE setup with tree-structured Bernoulli latent [7]; iii) the latent variable follows a distribution characterised by a tree-structured probabilistic graphical model, and the recognition network outputs tree-structured factor potentials (containing both the singleton and pairwise factor potentials), which is the TreeSRVAE. Note that we use the "hard-transformation" trick for all models presented in Figure 7. From Figure 7 we see that TreeSRVAE consistently outperforms both VAE and SVAE on the "Bar" dataset with varying level of sampling noise, in terms of the overall free energy (both the sample efficiency and asymptotic performance), hence providing a tighter lower bound to the true likelihood.

What if the underlying latent structure deviates largely from being tree-structured? We consider the following "side-dependent" prior distribution ($p(\mathbf{y}|\mathbf{z}, \mathbf{W}, \mathbf{b})$ remain the same).

$$p(\mathbf{z}|\theta) \propto \text{Cat}(z_{1:D}|p_{1:D})\text{Cat}(z_{D+1:2\times D}|p_{D+1:2\times D}) \tag{36}$$

where $\text{Cat}(\mathbf{p})$ denotes the categorical distribution with probabilities $\mathbf{p}$. Intuitively, sampling from such prior distribution means that we will observe one and only one bar on each side of the square. Clearly, such prior distribution cannot be modelled by any tree-structured distribution.

We apply the same set of models to the data from the new generative model, and the results are shown in Figure 8. We observe that the performances for all three methods are mostly similar. Hence in situations where there exists a large mismatch between the structures of the amortised potentials and the true posterior distribution, the additional structure present in the amortised inference will not interrupt learning, but instead allow the learning process be at least on the same level of performance as standard VAE and SVAE in terms of free energy.

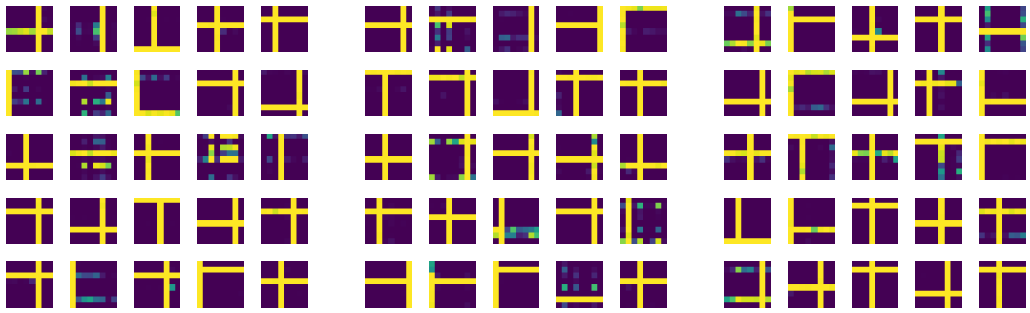
Despite having achieved similar free energy through training (Figure 8), we argue that the structured recognition framework should still be preferred. Namely, we examine the generation quality of the generated samples.

Firstly, visual inspection of the randomly samples from the three trained models in Figure 9 indicates TreeSRVAE generates more "cross" patterns than SVAE and VAE. The qualitative indicates TreeSRVAE has learned a better generative model than the other two models.

Now for the quantitative comparison. Given the relatively small latent state space (16 binary latents, leading to $2^{16} = 65536$ possible latent configurations). We exhaustively generate samples from all possible latent configurations given the three models. For each generated sample, we compute the most similar "cross" pattern (in terms of squared error) with respect to the sample, and their squared distance. We then compute the averaged smallest squared error given the generated samples of all possible 2^{16} latent configurations for the three models. The statistics is reported in Table 2. Namely, for each $z \in \mathbb{R}^{16}$,

$$\begin{aligned} \hat{y} &= p(y|z, \gamma) \\ y^* &= \underset{y \in \mathcal{Y}}{\text{argmax}} \, ||y - \hat{y}||^2 \\ d_z &= ||y^* - \hat{y}||^2 \end{aligned} \tag{37}$$

where $\mathcal{Y}$ denotes the set of all possible cross patterns (64 different patterns in an 8×8 square-grid. Hence we compare $\tilde{d} = \frac{1}{|\mathcal{Z}|} \sum_{z \in \mathcal{Z}} d_z$, where $\mathcal{Z}$ is the set of all possible latent configurations.



(a) (b) (c)

Figure 9: **Generation from trained models on the "bar" dataset with "side-depdent" prior.** (a) VAE (mean-field latent prior and mean-field recognition inference); (b) SVAE (tree-structured latent prior and mean-field recognition inference); (c) TreeSRVAE (tree-structured latent prior and tree-structured recognition inference). Note that all samples are not cherry-picked.

Hence we have shown that, despite having achieved similar free energy, TreeSRVAE enables stronger generation both qualitatively and quantitatively.

	VAE	SVAE	TreeSRVAE
$\tilde{d}$	1.297	0.559	**0.351**

Table 2: Comparison of $\tilde{d}$ for VAE, SVAE, and TreeSRVAE.

F.1.1 Tree-Structured Latent PGM with Gaussian Factors

We note that the tree-structured recognition network generalises beyond discrete latents. Namely, we developed a Gaussian-TreeSRVAE model (details in appendix B.1.1) and applied to the MNIST dataset [50]. Table 3 show that Gaussian-TreeSRVAE outperforms the two baseline models.

	VAE	SVAE	Gaussian-TreeSRVAE
Free energy	89.54 ± 0.21	91.52 ± 0.41	$\mathbf{88.39 \pm 0.55}$

Table 3: Free energy of VAE, SVAE and Gaussian-TreeSRVAE trained on MNIST dataset (given 100 training epochs).

F.2 SR-nlGPFA

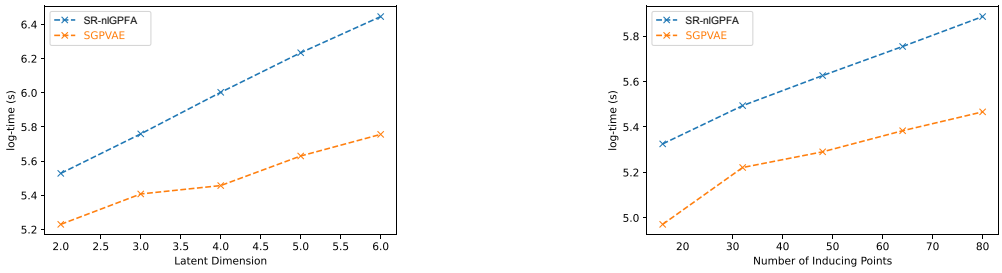
F.3 Full Quantitative Evaluation Comparison

The complete evaluation comparison between SR-nlGPFA and the selected baselines in terms of both the SMSE and the negative log-likeihood (NLL) is shown in Table 1.

F.3.1 Complexity Analysis of SR-nlGPFA

We note that SR-nlGPFA, despite allowing the inference of full-covariance structure over all latent processes, does not incur higher-order complexity than standard SVAEs with GP-latents (e.g., SGP-VAE [10]), hence allowing scalable application. The major difference between SR-nlGPFA and SGP-VAE lie in the inference step, where SGP-VAE requires only inverting the covariance matrices for the latent-specific inducing points, and SR-nlGPFA needs to numerically invert the covariance matrices over all inducing points across latent dimensions. Hence the complexities for such operation is $\mathcal{O}(K^3\bar{M}^3)$ and $\mathcal{O}(K\bar{M}^3)$ for SR-nlGPFA and SGP-VAE, respectively. Hence SR-nlGPFA incurs additional computations by a factor of K^2. However, we note that we usually seek a low-dimensional GPFA latents that characterise the manifold upon which the high-dimensional neural trajectories lie within, hence K is usually chosen to be small (e.g., $K = 4$ for the single-cell spiking dataset

from Section 5.2.2 and $K = 2$ for the EEG dataset considered in Section 5.2.1). Hence overall speaking, we expect considerably small increase in computation time. We show the comparison between the running times with varying latent dimensions and number of inducing points for SR-nlGPFA and SGP-VAE in Figure 10, which is in accordance with our hypothesis that small additional computational cost is incurred with SR-nlGPFA.



(a) (b)

Figure 10: Computation time (log-scale) comparison between SR-nlGPFA and SGP-VAE with varying latent dimension (a) and number of inducing points per latent dimension (b).

F.3.2 Further Analysis of SR-nlGPFA on Neural Population Spiking Dataset

The firing fields of some exemplary neurons on the Z-shaped track in shown in Figure 11. Visual inspections show that the majority of the recorded cells show spatial modulation, and resembling the firing patterns of place cells and (potentially) grid cells, amongst the recorded neurons there also exists ones whose firing patterns resemble that of interneurons (e.g., third neurons from the left on the third row).

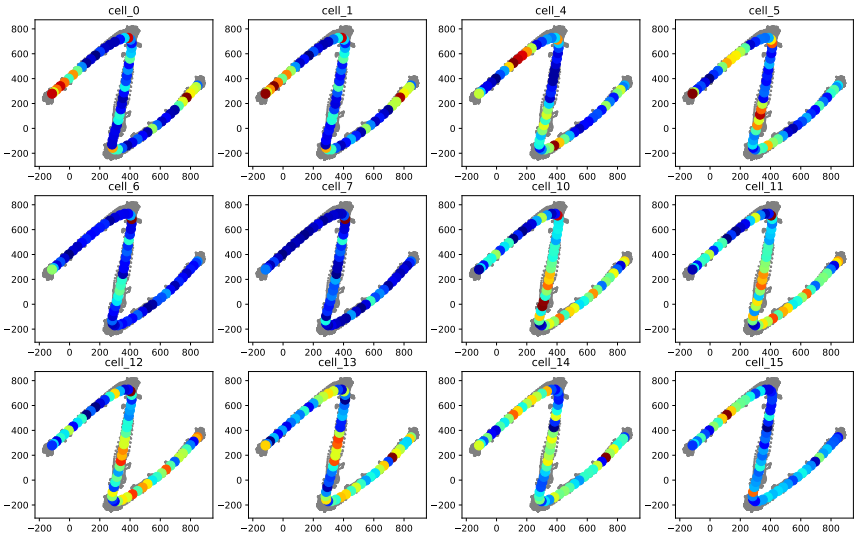


Figure 11: Firing patterns of selected CA1 and mEC neurons along the Z-shaped track (color represents firing rate).

Extraction of behavioural-covariate-modulated neurons predicted by the trained SR-nlGPFA model.

Given the behavioural covariate of interests, we wish to predict the neurons whose firing pattern exhibit strongest modulation with respect to the behavioural covariate using the trained SR-nlGPFA model (in a completely unsupervised fashion).

Specifically, given the projection matrices from the CCA fitting, we could qualitatively identify the latent processes that are most correlated with the target behavioural covariate. Given the generative likelihood, we could compute the gradient of the output Poisson rate parameters for each neuron with respect to the selected latent variable (the Jacobian matrix) using AutoDiff as found in most deep learning libraries [51]. We define the *relevance score* of a latent process, f, with respect to the firing of neuron s, $\Delta_f(s)$ as the average squared-norm of the gradient of the associated mapping (given the likelihood function, Eq. 14).

$$\Delta_f(s) = \int_{t=0}^{T} dt || \frac{\partial \lambda_{\text{NN}}(\mathbf{f}(t))_s}{\partial f} ||^2 \approx \frac{1}{N} \sum_{n=1}^{N} || \frac{\partial \lambda_{\text{NN}}(\mathbf{f}(x_n))_s}{\partial f} ||^2 \tag{38}$$

We categorise the top 20% of the neurons with the highest (or lowest if negatively correlated) relevance score as the model-predicted neurons that are modulated by the target behavioural covariate.

Additional svGP Inference Step with Changed Inducing Locations

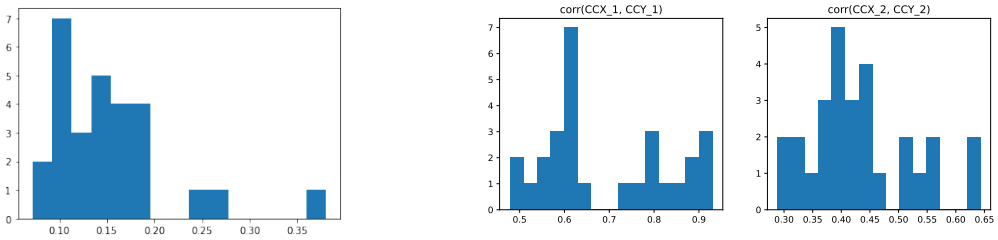
We note that since the recognition network outputs local evidence potentials on $\mathbf{h}$, the computation of $q(\mathbf{U})$ is not constrained to a single set of inducing locations $\mathbf{Z}$, rather we are free to choose the set of inducing points to work with (13). During training, due to the computational constraints of stochastic mini-batch training, it is impractical to use a large number of inducing points. However, the small number of inducing points leads to temporal chunking artifacts in the variational posterior distribution of the latent processes ($q(\mathbf{f})$) over the entire sequence. Hence SR-nlGPFA enables an additional svGP inference step given the trained recognition network and an expanded set of inducing points, leading to smoother temporal interpolation, hence alleviating the temporal chunking effects.

Quantitative Analysis of SR-nlGPFA across Experimental Sessions

Here we quantitatively evaluate various aspects of the posterior latent GPs given the trained models. We train independent models of SR-nlGPFA with the same architectures and training procedures (see appendix G for implementation details) on the single-cell population spiking datasets from each of the 28 experimental sessions considered. Firstly, we examine if full-covariance structure is a necessary assumption. We compute the ratio between the magnitudes of the off-diagonal entries and that of all entries in the posterior covariance matrix ($\mathbf{S}_n^f$, for all n, Section 5.2.2), $\sum_n \frac{\sum_{i,j,i\neq j} |\mathbf{S}_{n,ij}^f|}{\sum_{i,j} |\mathbf{S}_{n,ij}^f|}$. From Figure 12a we observe that the posterior latent covariance matrices of all sessions have non-trivial off-diagonal elements, indicating the existence of posterior correlations induced by "explaining away" and the necessity of posterior inference with full covariance structure for capturing the induced correlations.

In order to assess the quality of the learned posterior latents, we perform two-dimensional Canonical Correlation Analysis (CCA, [42]) on the learned posterior means and the behavioural covariates. In Figure 12b we observe that the correlations between the extract canonical correlates (CC) of the posterior means and the behavioural covariates exhibits high correlation over all sessions, hence indicating the extracted posterior latent dimensions captures large proportion of the information in the behavioural variables.

We further show the correlations between the $CCX\{1,2\}$ (CCs of the latent posterior means) and each of the behavioural covariates for all sessions in Figure 13. We observe strong correlations between one or both of the CCXs with the (unfolded) distance along the track quantity, showing that the extracted latent variables given the population spiking data is strongly indicative of the spatial location of the rat along the Z-shaped track, and corresponds nicely with prior knowledge that the majority of the recorded cells are located in the CA1 region of hippocampus and are well-known to exhibit significant spatial modulation [36]. Moreover, in most sessions we observe high correlations (in terms of magnitude) between the CCXs and the direction of travelling, again conforming with experimental evidence that CA1 place cells firing are modulated by direction along the linear track (the Z-shaped track environment is usually interpreted as a 1D linear track rather than a 2D environment [36]). We additionally observe that the CCXs of the learned latent variables in most sessions show strong correlation with respect to speed of travelling, which also has nice experimental correspondence [38, 40].



(a) (b)

Figure 12: Quantitative analysis of SR-nlGPFA across experimental sessions. (a) Histogram of the ratio of the magnitude of the off-diagonal entries relative to that of all entries in the covariance matrix of $q(\mathbf{f})$; (b) Histogram of correlation coefficients between $CCX\{1,2\}$ and $CCY\{1,2\}$.

Comparison with SGP-VAE

By applying SGP-VAE to the data from the same session studied in Section 5.2.2, we perform similar CCA analysis on the learned latent variables of SGP-VAE. We observe that the correlations between $CCX\{1,2\}$ and $CCY\{1,2\}$ of SR-nlGPFA are significantly larger than that of SGP-VAE (with standard t-test, p-values shown in the figure; Figure 14a).

We additionally examine the same session we looked at in Section 5.2.2, and plot the $CCX\{1,2\}$ of the learned latent variables of SGP-VAE against the x- and y-positions. By comparing to the similar plot with SR-nlGPFA (Figure 3d), we observe that the resulting plot does not show clear clustered structure with respect to spatial and direction covariates compared to that of SR-nlGPFA, hence illustrating that SR-nlGPFA learns qualitatively better latent dimensions than SGP-VAE with respect to the underlying behavioural covariates.

G Implementation Details

"Bar" Dataset

The details of the generation of the "Bar" dataset can be found in appendix **??**. We use the same neural network architectures for all models considered (VAE, SVAE, TreeSRVAE): both the recognition and generative networks are MLPs with two hidden layers of 50 hidden units, with ReLU non-linear activation function, and the latent dimension is 16. All models and sessions are trained with Adam optimiser with learning rate of 5×10^{-4} over 3500 epochs with a batch-size of 256 [52].

Neural Population Spiking Dataset

We use the same neural network architectures for both SR-nlGPFA and SGP-VAE: both the recognition and generative networks are MLPs with two hidden layers with 256 hidden units, with ReLU activatieon function. The latent dimensions for both model is 6, which we choose via cross-validation. Note that the latent dimension of 6 corresponds to the results from other existing unsupervised latent feature extraction works on CA1 neuron firing patterns (e.g., Nieh et al. [53] reported $5-7$ is optimal for the latent dimension with their manifold inference model). The dimension of the neurla feature (h) in the GPFA generative model (Eq. 3) is 20. Both models are trained with Adam optimiser with learning rate of 1×10^{-4} over 400 epochs with a batch-size of 128 [52]. The number of inducing points for each latent dimension is 64 for both models.

For the additional SVGP inference step after training with SR-nlGPFA (SGP-VAE does not allow the additional SVGP inference step as discussed in Section 5.2.2), we set the number of inducing points to be 1000 for all latent dimensions to smooth out the temporal chunking artifacts caused by the small number of inducing points in training due to scalability concerns with stochastic optimisation.

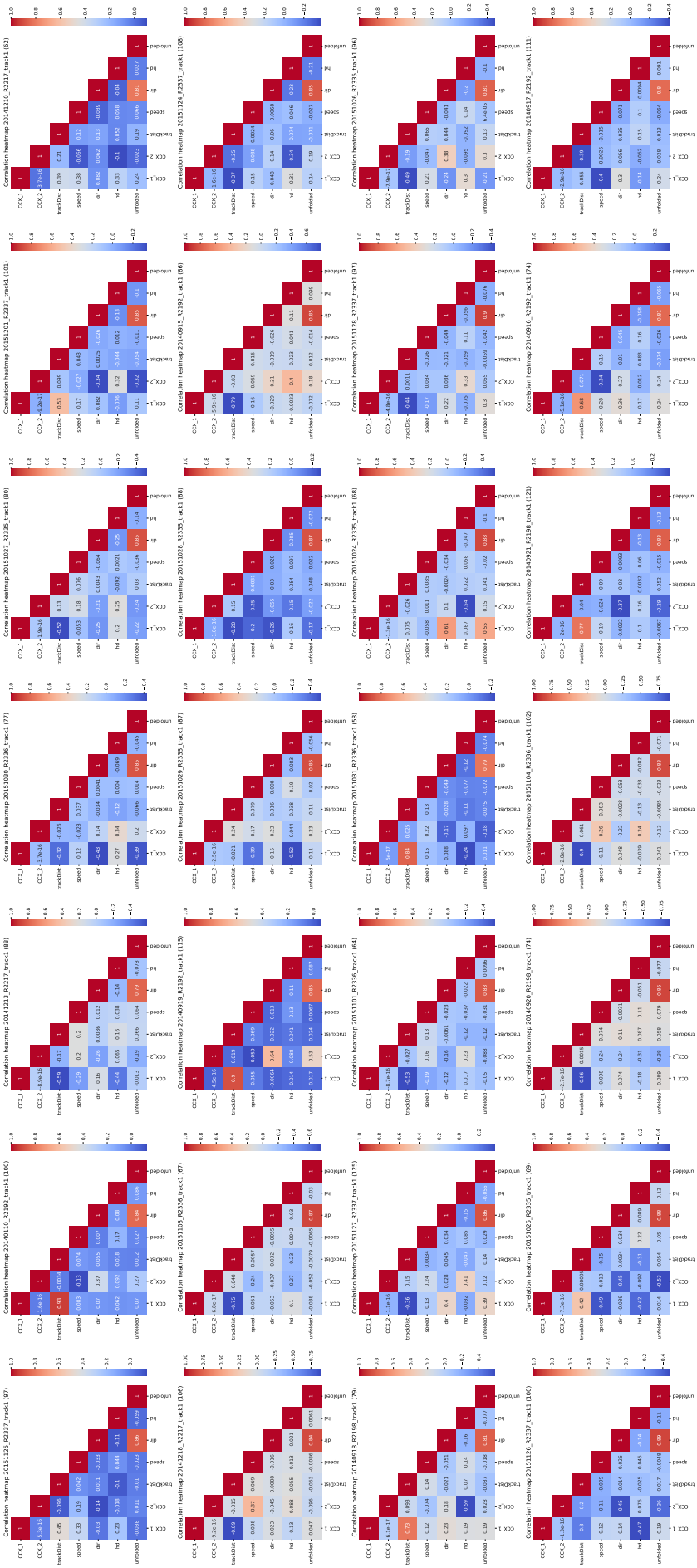
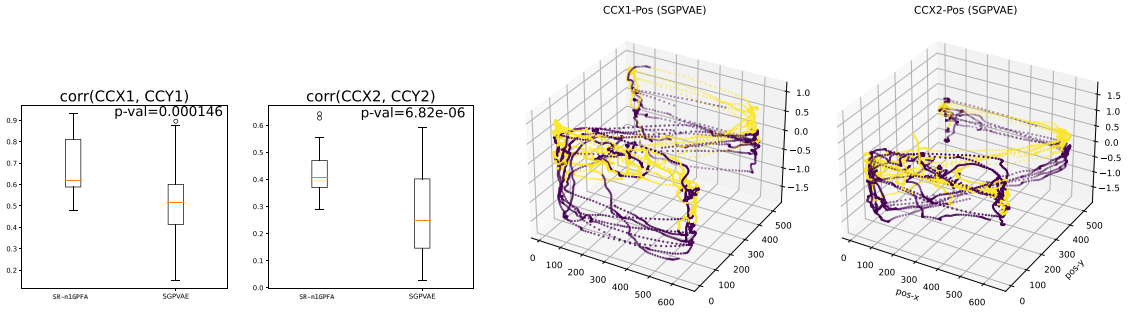


Figure 13: Heatmaps of correlation coefficients between $CCX\{1,2\}$ of posterior latent mean given SR-nlGPFA and the behavioural covariates.



(a) (b)

Figure 14: Comparison with SGP-VAE. (a) Comparison of $corr(CCX1, CCY1)$ (left) and $corr(CCX2, CCY2)$ (right) between SR-nlGPFA and SGP-VAE; (b) posterior mean of the latent process learned by SGP-VAE against x- and y-location of the rat, color indicating direction of travelling (yellow: inbound, magenta: outbound).