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

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

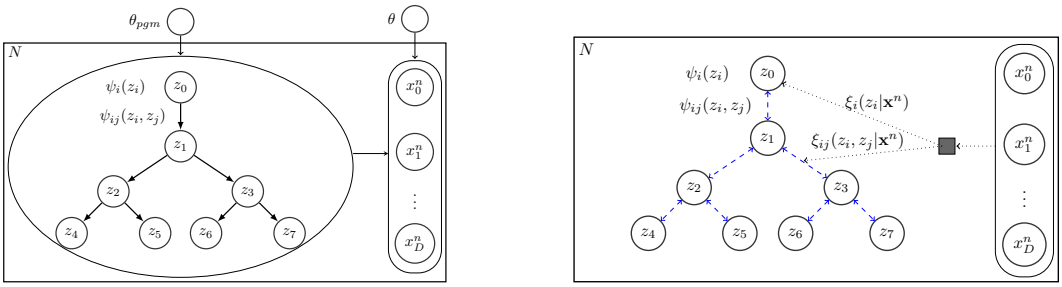
Due to the tree structure (i.e., no loops), belief propagation converges after a single inward-outward pass (similar to forward-backward propagation in HMM models). Upon convergence, we can compute the marginal distributions of the variables and factors as products of messages and factor potentials.

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

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, $m_{i\to j}^{(k)}(x_j) \sim \mathcal{N}(x_j|\mu_{i\to j}^{(k)}, (\lambda_{i\to j}^{(k)})^{-1})$. At each iteration k, the message updates are shown as following.

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

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

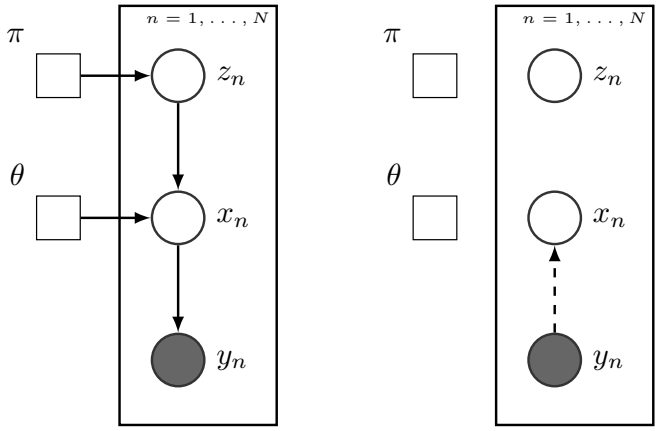


Figure 5: **SRVAEframework implemented with tree-structured latent variable model. (a)** generative model **(b)** inference model, with tree-structured (or a joint factor over all latent variables) recognition potentials.

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.

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

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 ($D = 8$).

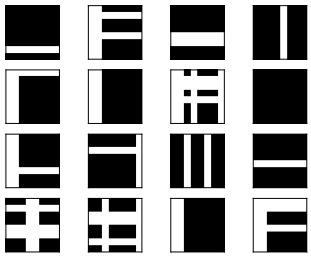
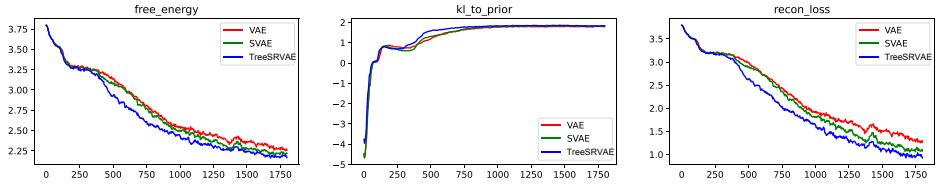
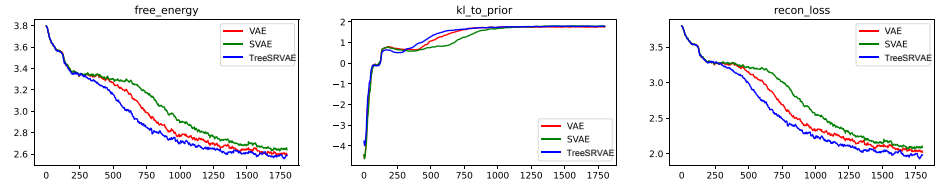


Figure 6: **Samples from the "Bar" dataset**.



(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

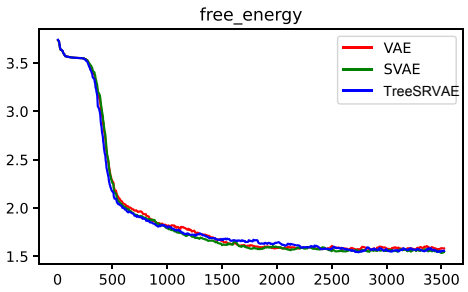


Figure 8: Free energy of the TreeSRVAE, SVAE and VAE over the training process on the "side-dependent" "Bar" dataset (Eq. 36)

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.

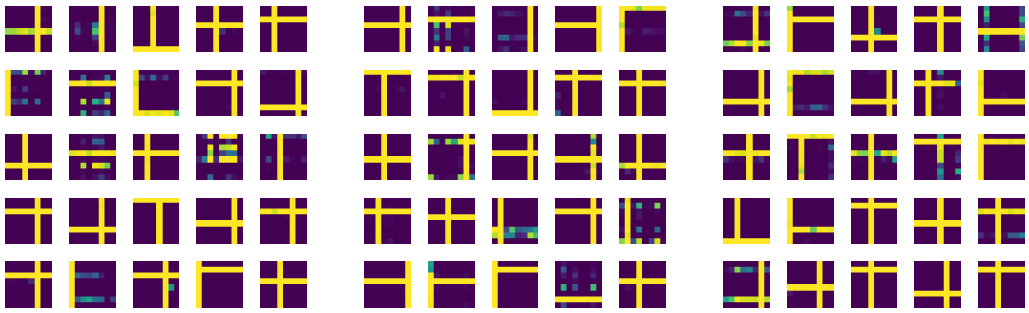
Despite having achieved similar free energy through training (Figure 8), we argue that the structured recognition framework should still be preferred. Namely, we examine the generation quality of the generated samples.

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

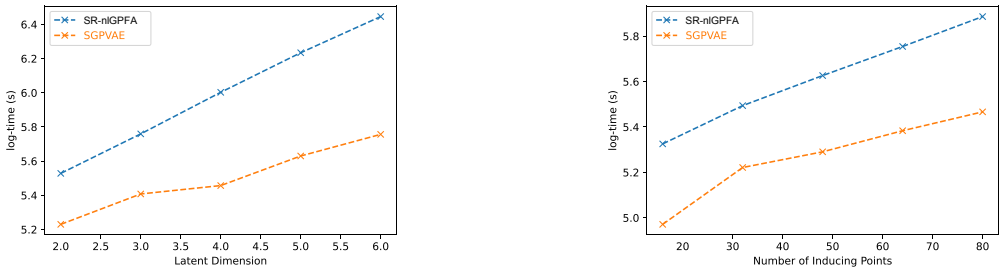
F.3 Full Quantitative Evaluation Comparison

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).

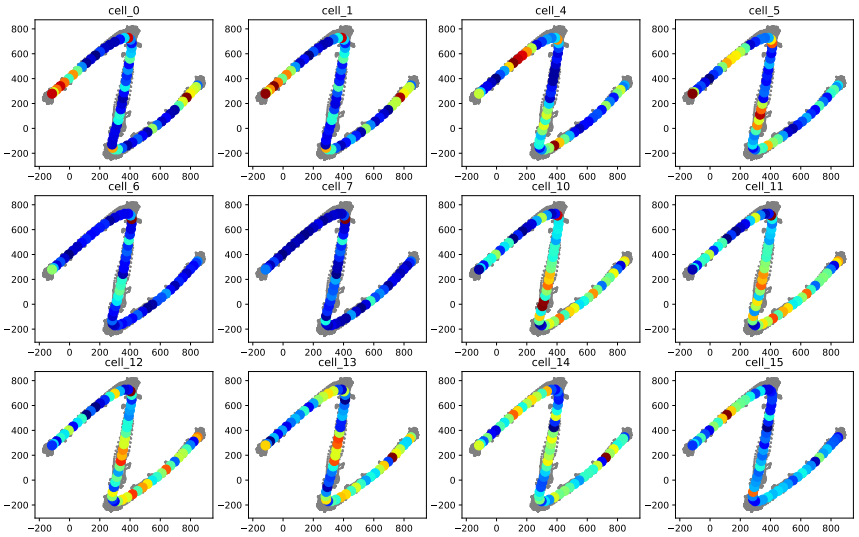


Figure 11: Firing patterns of selected CA1 and mEC neurons along the Z-shaped track (color represents firing rate).

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

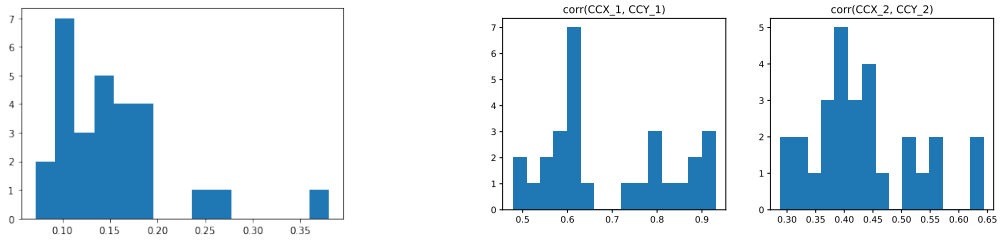
We note that since the recognition network outputs local evidence potentials on $\mathbf{h}$, the computation of $q(\mathbf{U})$ is not constrained to a single set of inducing locations $\mathbf{Z}$, rather we are free to choose the set of inducing points to work with (13). During training, due to the 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 F.1. 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.

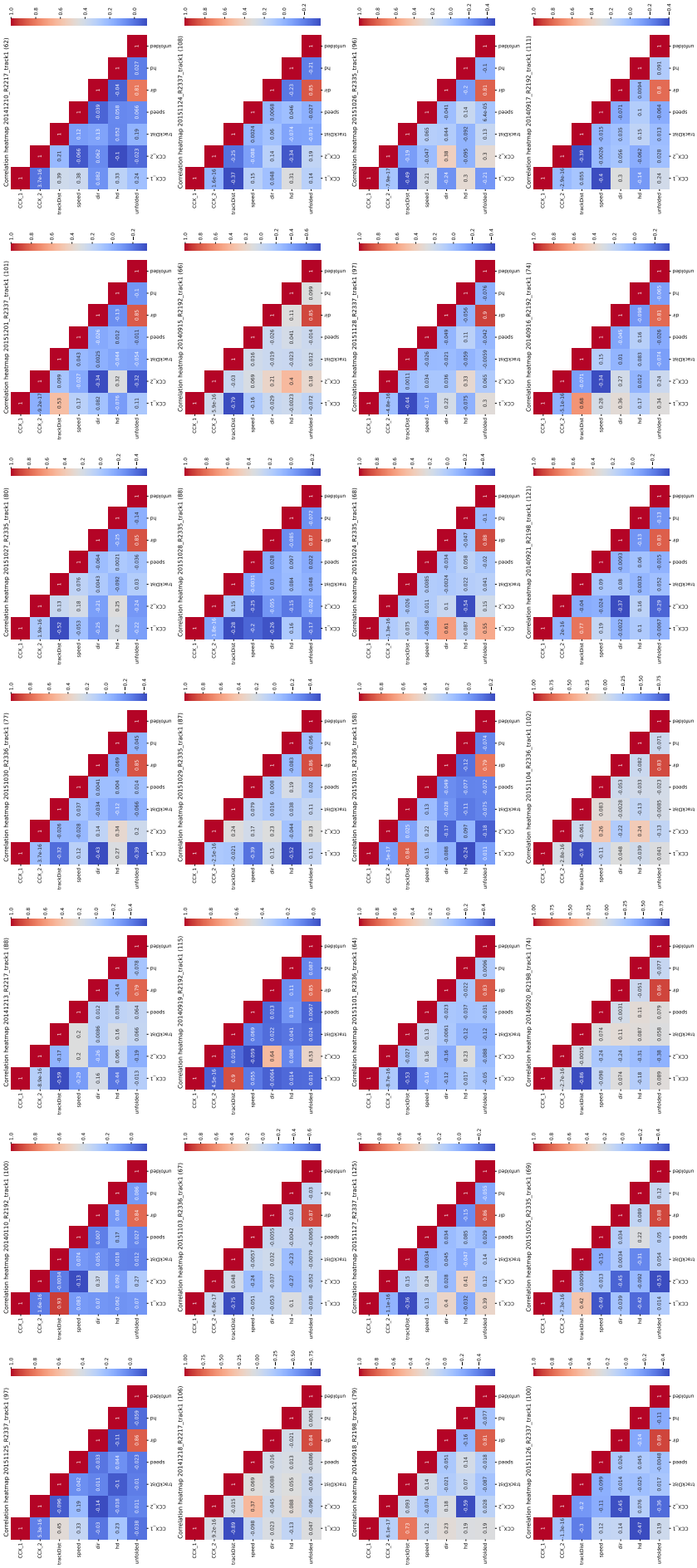
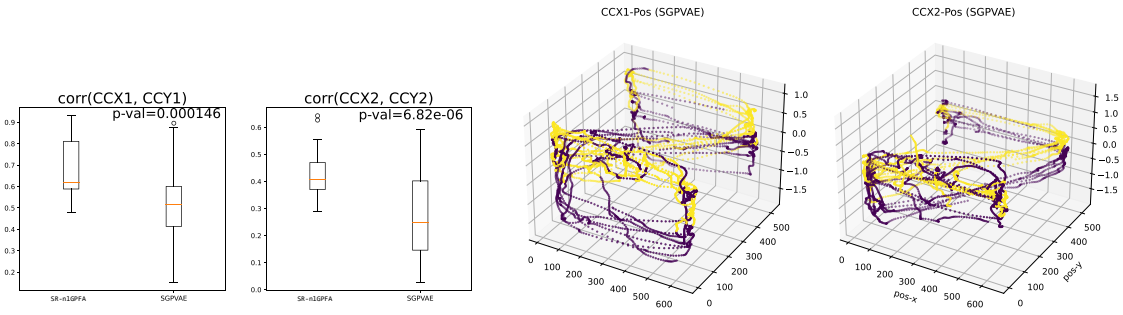


Figure 13: Heatmaps of correlation coefficients between $CCX\{1, 2\}$ of posterior latent mean given SR-nlGPFA and the behavioural covariates.



(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).