Longitudinal Variational Autoencoder: Supplementary Materials

1 Covariance functions

Squared exponential CF

Let $\boldsymbol{x}^{(r)} = x \in \mathcal{X}_j$ denote an univariate continuous-valued covariate. The squared exponential (SE) CF is defined as

$$k_{\text{se}}(\boldsymbol{x}^{(r)}, \boldsymbol{x}^{(r)\prime}|\theta_{\text{se}}) = \sigma^2_{\text{se}} \exp\left(-\frac{(x-x')^2}{2\ell^2_{\text{se}}}\right), \qquad \theta_{\text{se}} = (\sigma^2_{\text{se}}, \ell_{\text{se}})$$

where σ^2_{se} is the magnitude parameter (also called scale) and $\ell_{\text{se}} \geq 0$ is the length-scale. The magnitude controls the marginal variance of the GP and length-scale controls its smoothness [Rasmussen and Williams, 2006].

Categorical CF

Let $\boldsymbol{x}^{(r)} = x \in \mathcal{X}_j$ denote a categorical or discrete covariate. The categorical CF is defined as:

$$k_{\text{ca}}(\boldsymbol{x}^{(r)}, \boldsymbol{x}^{(r)\prime}) = \begin{cases} 1, & \text{if } x = x' \\ 0, & \text{otherwise} \end{cases} \qquad \theta_{\text{ca}} = \emptyset$$

Binary CF

Let $\boldsymbol{x}^{(r)} = x \in \mathcal{X}_j$ denote an arbitrary univariate covariate. The binary CF is defined as:

$$k_{\text{bi}}(\boldsymbol{x}^{(r)}, \boldsymbol{x}^{(r)\prime}) = \begin{cases} 1, & \text{if } x = x' = 1 \\ 0, & \text{otherwise} \end{cases} \qquad \theta_{\text{bi}} = \emptyset$$

Interaction CF

Let $\boldsymbol{x}^{(r)} = [\boldsymbol{x}^{(a)^T}, \boldsymbol{x}^{(b)^T}]^T$, where $\boldsymbol{x}^{(a)}$ and $\boldsymbol{x}^{(b)}$ are arbitrary sub-vectors of $\boldsymbol{x}$. We define the interaction CF for these subsets as the product of two CFs defined over $\boldsymbol{x}^{(a)}$ and $\boldsymbol{x}^{(b)}$ respectively:

$$k_{\text{in}}(\boldsymbol{x}^{(r)}, \boldsymbol{x}^{(r)\prime}|\theta^{(r)}) = k^{(a)}(\boldsymbol{x}^{(a)}, \boldsymbol{x}^{(a)\prime}|\theta^{(a)}) k^{(b)}(\boldsymbol{x}^{(b)}, \boldsymbol{x}^{(b)\prime}|\theta^{(b)}),$$

where $\theta^{(r)} = \theta^{(a)} \cup \theta^{(b)}$. The interaction CF enables us to combine any combination of univariate SE, categorical, and binary CFs in a product CF. However, in practice we restrict such combinations to include no more than a single SE CF. As Cheng et al. [2019] stated, such a condition affords the SE GP flexibility with random intercept/slope constructions similar to the linear mixed effect modelling framework, without sacrificing interpretability. For example, in longitudinal studies an instance-specific auto-correlated temporal deviation from the population-level temporal mean can be captured by an interaction term between a categorical CF over the instance identifiers and a SE CF over the temporal covariate. Furthermore, Cheng et al. [2019] adapted an approach to handle missing covariates based on an interaction CF containing a missing-ness mask.

2 Efficient KL divergence computation

As the main text states, optimising the variational objective of L-VAE involves the computation of L KL divergences $D_{\mathrm{KL}} = D_{\mathrm{KL}}(\mathcal{N}(\bar{\boldsymbol{\mu}}_l, W_l)||\mathcal{N}(\mathbf{0}, \Sigma_l))$, where $\bar{\boldsymbol{\mu}}_l = [\mu_{\phi,l}(\boldsymbol{y}_1), \dots, \mu_{\phi,l}(\boldsymbol{y}_N)]^T$, $W_l = \mathrm{diag}(\sigma^2_{\phi,l}(\boldsymbol{y}_1), \dots, \sigma^2_{\phi,l}(\boldsymbol{y}_N))$, and $\Sigma_l = \sum_{r=1}^{R} K_{XX}^{(l,r)} + \sigma^2_{zl} I_N$. Henceforth, we drop the index l for notational simplicity. Each of the KL divergences is available in closed form using the well-known expression for the KL divergence between two multivariate normal distributions:

$$D_{\mathrm{KL}} = \frac{1}{2}\left(\mathrm{tr}(\Sigma^{-1}W) + \bar{\boldsymbol{\mu}}^T\Sigma^{-1}\bar{\boldsymbol{\mu}} - N + \log|\Sigma| - \log|W|\right),$$

but its exact computation requires $\mathcal{O}(N^3)$ flops, which makes it impractical when N exceeds a few thousands. In this section, we provide a derivation of a novel strategy to approximately compute this KL divergence at a reduced computational cost.

KL divergence and evidence lower bound We start by exploiting the diagonal structure of W and establish the connection between the upper bound for D_{KL} and the evidence lower bound for the marginal log likelihood (MLL) of a Gaussian process:

$$\begin{aligned} D_{\mathrm{KL}}(\mathcal{N}(\bar{\boldsymbol{\mu}}, W)||\mathcal{N}(\mathbf{0}, \Sigma)) &\triangleq \int_{\boldsymbol{z}} \mathcal{N}(\boldsymbol{z}|\bar{\boldsymbol{\mu}}, W) \log\left(\frac{\mathcal{N}(\boldsymbol{z}|\bar{\boldsymbol{\mu}}, W)}{\mathcal{N}(\boldsymbol{z}|\mathbf{0}, \Sigma)}\right) d\boldsymbol{z} \\ &= -\int_{\boldsymbol{z}} \log\left(\mathcal{N}(\boldsymbol{z}|\mathbf{0}, \Sigma)\right) \mathcal{N}(\boldsymbol{z}|\bar{\boldsymbol{\mu}}, W) d\boldsymbol{z} - \frac{1}{2}\log|(2\pi e)W| \\ &\leq -\int_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{z}; \Sigma)\mathcal{N}(\boldsymbol{z}|\bar{\boldsymbol{\mu}}, W) d\boldsymbol{z} - \frac{1}{2}\log|(2\pi e)W| \qquad (1) \\ &\text{for any function } \mathcal{L}(\boldsymbol{z}; \Sigma) : \mathcal{L}(\boldsymbol{z}; \Sigma) \leq \log\left(\mathcal{N}(\boldsymbol{z}|\mathbf{0}, \Sigma)\right) \ \forall \boldsymbol{z}. \end{aligned}$$

Hence, for any lower bound of GP MLL $\mathcal{L}(\boldsymbol{z}; \Sigma)$, the corresponding expression would provide an upper bound for KL divergence between the considered multivariate normal distributions. However, the lower bound of GP MLL is a much more common and well-studied problem. Please note that in the expression (1) and further throughout this section, we specifically use the *lower bound* and *ELBO* terms for the GP MLL lower bound $\mathcal{L}(\boldsymbol{z}; \Sigma)$, and not for the lower bound of the deep generative model as in the main text.

Variational learning of inducing variables in sparse Gaussian processes One of the most fundamental works on ELBOs for GP MLL was done by Titsias [2009], and is based on the paradigm of low-rank inducing point approximations of GPs. We briefly recap their key results here, and later build upon their derivation by introducing modifications that would suit the specific structure of matrices Σ in our problem setting.

We denote the set of inducing points locations in $\mathcal{X}$ as $S = [\boldsymbol{s}_1^T, \dots, \boldsymbol{s}_M^T]^T$, and the value of the Gaussian process at the inducing locations as $\boldsymbol{u} = [u_1, \dots, u_M]^T$. We recall that $\Sigma = K_{XX} + \sigma_z^2 I_N$ and therefore, $\boldsymbol{z}$ can be represented as the sum of noise-free GP, $\boldsymbol{f} \sim \mathcal{N}(\mathbf{0}, K_{XX})$ and i.i.d. Gaussian noise. The following identities always stand:

$$\begin{aligned} p(\boldsymbol{z}|\boldsymbol{f}) &= \mathcal{N}(\boldsymbol{z}|\boldsymbol{f}, \sigma_{\boldsymbol{z}}^2 I_N) \\ p(\boldsymbol{f}|\boldsymbol{u}) &= \mathcal{N}(\boldsymbol{f}|K_{XS}K_{SS}^{-1}\boldsymbol{u}, \tilde{K}), \\ \tilde{K} &= K_{XX} - K_{XS}K_{SS}^{-1}K_{SX} \\ p(\boldsymbol{u}) &= \mathcal{N}(\boldsymbol{u}|\mathbf{0}, K_{SS}) \\ p(\boldsymbol{z}) &= \int_{\boldsymbol{u}}\int_{\boldsymbol{f}} p(\boldsymbol{z}|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{u})p(\boldsymbol{u})d\boldsymbol{f}d\boldsymbol{u}. \end{aligned} \qquad (2)$$

Applying the Jensen inequality on the conditional log-probability $p(\boldsymbol{z}|\boldsymbol{u})$ leads to:

$$\log p(\boldsymbol{z}|\boldsymbol{u}) = \log \int_{\boldsymbol{f}} p(\boldsymbol{z}|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{u})d\boldsymbol{f} \qquad (3)$$

$$\geq \int_{\boldsymbol{f}} \log\left(p(\boldsymbol{z}|\boldsymbol{f})\right) p(\boldsymbol{f}|\boldsymbol{u})d\boldsymbol{f} = \sum_{i=1}^{N}\left[\log\mathcal{N}(z_i|\mu_i, \sigma_{\boldsymbol{z}}^2) - \frac{\tilde{K}_{ii}}{2\sigma_{\boldsymbol{z}}^2}\right], \qquad (4)$$

where $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_N]^T = K_{XS}K_{SS}^{-1}\boldsymbol{u}$ and $\tilde{K}_{ii}$ denotes the i^{th} diagonal element of $\tilde{K}$. The inequation reduces to identity *iff* $\boldsymbol{u}$ is a sufficient statistic of $\boldsymbol{f}$, so that all elements of $\tilde{K}$ are zero. However, the inequation remains tight and the approximation is justified as long as the $\tilde{K}_{ii}$ elements remain small, which is achieved by setting M to be sufficiently high and optimising the inducing point locations S. After integrating out $\boldsymbol{u}$, this approximation leads to the collapsed representation of the variational evidence lower bound,

$$\mathcal{L}_1(\boldsymbol{z}; \Sigma) \triangleq \log \mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, K_{XS}K_{SS}^{-1}K_{SX} + \sigma_{\boldsymbol{z}}^2 I_N) - \frac{1}{2\sigma_{\boldsymbol{z}}^2}\text{tr}(\tilde{K}). \tag{5}$$

Divergence upper bound for longitudinal Gaussian process The free-form bound is known to be tight when M is sufficiently high and the covariance function is sufficiently smooth. However, longitudinal studies, by definition, always contain a categorical covariate corresponding to instances, which makes the covariance function non-continuous. By separating the additive component that corresponds to the interaction between instances and time (or age) from the other additive components, the covariance matrix has the following general form $\Sigma = K_{XX}^{(A)} + \hat{\Sigma}$, where $\hat{\Sigma} = \text{diag}(\hat{\Sigma}_1, \ldots, \hat{\Sigma}_P)$, $\hat{\Sigma}_p = K_{X_pX_p}^{(R)} + \sigma_z^2 I_{n_p}$, and $K_{XX}^{(A)} = \sum_{r=1}^{R-1} K_{XX}^{(r)}$ contains all the other $R-1$ components. In order to keep the $\tilde{K}_{ii}$ tight, the bound from eq. (5) would mandate that $M \geq P$, thus rendering the bound of Titsias [2009] either computationally inefficient once P is large (due to high M) or insufficiently tight. Since the interaction covariance function is essential for accurate longitudinal modelling, we devised a novel free-form divergence upper bound for such a class of GPs. Similar to eq. (2), we exploit the opportunity to represent $\boldsymbol{z}$ as a sum of noise-free GP $\boldsymbol{f}_A \sim \mathcal{N}(\boldsymbol{0}, K_{XX}^{(A)})$ and structured noise $\hat{\boldsymbol{f}} \sim \mathcal{N}(\boldsymbol{0}, \hat{\Sigma})$, and we assign the locations as well as values of inducing points for $\boldsymbol{f}_A$. We denote $\tilde{K}^{(A)} = K_{XX}^{(A)} - K_{XS}^{(A)}{K_{SS}^{(A)}}^{-1}K_{SX}^{(A)}$ and $\tilde{K}_{X_pX_p}^{(A)} = K_{X_pX_p}^{(A)} - K_{X_pS}^{(A)}{K_{SS}^{(A)}}^{-1}K_{SX_p}^{(A)}$, then:

$$\begin{aligned}
\log p(\boldsymbol{z}|\boldsymbol{u}) &= \log \int_{\boldsymbol{f}_A} p(\boldsymbol{z}|\boldsymbol{f}_A)p(\boldsymbol{f}_A|\boldsymbol{u})d\boldsymbol{f}_A \\
&= \log \int_{\boldsymbol{f}_A} \mathcal{N}(\boldsymbol{z}|\boldsymbol{f}_A, \hat{\Sigma})\mathcal{N}(\boldsymbol{f}_A|K_{XS}^{(A)}{K_{SS}^{(A)}}^{-1}\boldsymbol{u}, \tilde{K}^{(A)})d\boldsymbol{f}_A \\
&\geq \int_{\boldsymbol{f}_A} \log\left(\mathcal{N}(\boldsymbol{z}|\boldsymbol{f}_A, \hat{\Sigma})\right)\mathcal{N}(\boldsymbol{f}_A|K_{XS}^{(A)}{K_{SS}^{(A)}}^{-1}\boldsymbol{u}, \tilde{K}^{(A)})d\boldsymbol{f}_A \\
&= \sum_{p=1}^{P}\left[\log \mathcal{N}\left(\boldsymbol{z}_p|\boldsymbol{\mu}_p, \hat{\Sigma}_p\right) - \frac{1}{2}\text{tr}\left(\hat{\Sigma}_p^{-1}\tilde{K}_{X_pX_p}^{(A)}\right)\right],
\end{aligned}$$

where $\boldsymbol{z}_p$ is the sub-vector of $\boldsymbol{z}$ that corresponds to the p^{th} individual, $\boldsymbol{\mu}_p$ is the subvector of $K_{XS}^{(A)}{K_{SS}^{(A)}}^{-1}\boldsymbol{u}$, and $\text{tr}(\cdot)$ denotes the matrix trace operator. After integrating out $\boldsymbol{u}$, we obtain a novel variational evidence lower bound,

$$\mathcal{L}_2(\boldsymbol{z}; \Sigma) \triangleq \log \mathcal{N}\left(\boldsymbol{z}|\boldsymbol{0}, K_{XS}^{(A)}{K_{SS}^{(A)}}^{-1}K_{SX}^{(A)} + \hat{\Sigma}\right) - \frac{1}{2}\sum_{p=1}^{P}\text{tr}\left(\hat{\Sigma}_p^{-1}\tilde{K}_{X_pX_p}^{(A)}\right). \tag{6}$$

We substitute this lower bound in eq. (1), which links the ELBO and upper bound for KLD:

$$\begin{aligned}
D_{\text{KL}} &= D_{\text{KL}}(\mathcal{N}(\bar{\boldsymbol{\mu}}, W||\mathcal{N}(\boldsymbol{0}, \Sigma)) \leq -\int_{\boldsymbol{z}} \mathcal{L}_2(\boldsymbol{z}; \Sigma)\mathcal{N}(\boldsymbol{z}|\bar{\boldsymbol{\mu}}, W)d\boldsymbol{z} - \frac{1}{2}\log|(2\pi e)W| \\
&= -\int_{\boldsymbol{z}} \log \mathcal{N}\left(\boldsymbol{z}|K_{XS}^{(A)}{K_{SS}^{(A)}}^{-1}K_{SX}^{(A)} + \hat{\Sigma}\right)\mathcal{N}(\boldsymbol{z}|\bar{\boldsymbol{\mu}}, W)d\boldsymbol{z} - \frac{1}{2}\log|(2\pi e)W| + \frac{1}{2}\sum_{p=1}^{P}\text{tr}\left(\hat{\Sigma}_p^{-1}\tilde{K}_{X_pX_p}^{(A)}\right) \\
&= D_{\text{KL}}\left(\mathcal{N}(\bar{\boldsymbol{\mu}}, W||\mathcal{N}(\boldsymbol{0}, K_{XS}^{(A)}{K_{SS}^{(A)}}^{-1}K_{SX}^{(A)} + \hat{\Sigma}\right) + \frac{1}{2}\sum_{p=1}^{P}\text{tr}\left(\hat{\Sigma}_p^{-1}\tilde{K}_{X_pX_p}^{(A)}\right) \\
&= \frac{1}{2}\left(\text{tr}(\bar{\Sigma}^{-1}W) + \bar{\boldsymbol{\mu}}^T\bar{\Sigma}^{-1}\bar{\boldsymbol{\mu}} - N + \log|\bar{\Sigma}| - \log|W| + \sum_{p=1}^{P}\text{tr}\left(\hat{\Sigma}_p^{-1}\tilde{K}_{X_pX_p}^{(A)}\right)\right) \triangleq \mathcal{D}_2,
\end{aligned} \tag{7}$$

where $\bar{\Sigma} = K_{XS}^{(A)}{K_{SS}^{(A)}}^{-1}K_{SX}^{(A)} + \hat{\Sigma}$.

Theorem 1. *For any set of inducing points S and $\bar{S}$ where $\bar{S} \subseteq S$, such that* $\text{rank}(K_{SS}) = \text{rank}(K^{(A)}_{\bar{S}\bar{S}})$*, non-strict inequality $\mathcal{L}_1(\boldsymbol{z}; \Sigma, S) \leq \mathcal{L}_2(\boldsymbol{z}; \Sigma, \bar{S})$ holds.*

Proof. Based on the univariate additive GP $f(\boldsymbol{x}) = \sum_{r=1}^{R} f^{(r)}(\boldsymbol{x})$, we construct another GP $g(v_1, v_2, \boldsymbol{x}) = \mathbb{1}(v_1) \sum_{r=1}^{R-1} f^{(r)}(\boldsymbol{x}) + \mathbb{1}(v_2) f^{(R)}(\boldsymbol{x})$, where $\mathbb{1}(v)$ is the indicator function that equals 1 when $v = 1$ and 0 otherwise. Accordingly, we define augmented sets $\hat{X} = [\mathbf{1}_N, \mathbf{1}_N, X]$ and $\hat{S} = [\mathbf{1}_M, \mathbf{1}_M, S]$, where $\mathbf{1}_N$ is a column vector of ones that has length N, which effectively adds the two additional covariates, v_1 and v_2, into X and S. Then, the marginal covariance of $g(v_1, v_2, \boldsymbol{x})$ for $(\hat{X}, \hat{X})$ is K_{XX}, for $(\hat{X}, \hat{S})$ is K_{XS}, and for $(\hat{S}, \hat{S})$ is K_{SS}. Thus, $\mathcal{L}_1(\boldsymbol{z}; \Sigma, S) = \mathcal{L}_1(\boldsymbol{z}; \Sigma, \hat{S})$.

Since the collapsed lower bound, eq. (5), is obtained as a closed-form solution to the optimisation problem of the variational parameters of the distribution over the inducing points, expanding the set of the inducing points can only expand the variational family and, therefore, never decreases the $\mathcal{L}_1(\boldsymbol{z}; \Sigma, \hat{S})$ bound [Titsias, 2009]. We consider an expanded set of inducing points $\tilde{S} = [\hat{S}^T, \hat{S}_A^T, \hat{S}_B^T, \hat{X}_B^T]^T$, where $\hat{S}_A = [\mathbf{1}_M, \mathbf{0}_M, S]$ ($\mathbf{0}_M$ is a column vector of zeros that has length M), $\hat{S}_B = [\mathbf{0}_M, \mathbf{1}_M, S]$, and $\hat{X}_B = [\mathbf{0}_N, \mathbf{1}_N, X]$. For any $\boldsymbol{s} \in \hat{S}$, $g(1, 1, \boldsymbol{s}) = g(1, 0, \boldsymbol{s}) + g(0, 1, \boldsymbol{s})$, so that the values of $g(v_1, v_2, \boldsymbol{x})$ over $\hat{S}$ are linearly dependent on the values over $\hat{S}_A$ and $\hat{S}_B$. Therefore, the variational family remains the same for the reduced set of inducing points $\check{S} = [\hat{S}_A^T, \hat{S}_B^T, \hat{X}_B^T]^T$, and $\mathcal{L}_1(\boldsymbol{z}; \Sigma, \hat{S}) \leq \mathcal{L}_1(\boldsymbol{z}; \Sigma, \tilde{S}) = \mathcal{L}_1(\boldsymbol{z}; \Sigma, \check{S})$.

We will next write the $\mathcal{L}_1$ bound from equation (5) for $\check{S}$:

$$\mathcal{L}_1(\boldsymbol{z}; \Sigma, \check{S}) = \log \mathcal{N}(\boldsymbol{z}|\mathbf{0}, \check{K}_{X\check{S}} \check{K}^{-1}_{\check{S}\check{S}} \check{K}_{\check{S}X} + \sigma^2_{\boldsymbol{z}} I_N) - \frac{1}{2\sigma^2_{\boldsymbol{z}}} \text{tr}(K_{XX} - \check{K}_{X\check{S}} \check{K}^{-1}_{\check{S}\check{S}} \check{K}_{\check{S}X}),$$

where (from the definition of $\check{S}$)

$$\check{K}_{X\check{S}} = \left[K^{(A)}_{XS}, K^{(R)}_{XS}, K^{(R)}_{XX}\right] \quad \text{and} \quad \check{K}_{\check{S}\check{S}} = \begin{bmatrix} K^{(A)}_{SS} & 0 & 0 \\ 0 & K^{(R)}_{SS} & K^{(R)}_{SX} \\ 0 & K^{(R)}_{XS} & K^{(R)}_{XX} \end{bmatrix}.$$

We recall the 2-by-2 block-matrix inverse formula for the second and third blocks of $\check{K}_{\check{S}\check{S}}$:

$$\begin{bmatrix} K^{(R)}_{SS} & K^{(R)}_{SX} \\ K^{(R)}_{XS} & K^{(R)}_{XX} \end{bmatrix}^{-1} = \begin{bmatrix} Q^{-1} & -Q^{-1} K^{(R)}_{SX} {K^{(R)}_{XX}}^{-1} \\ -{K^{(R)}_{XX}}^{-1} K^{(R)}_{XS} Q^{-1} & {K^{(R)}_{XX}}^{-1} + {K^{(R)}_{XX}}^{-1} K^{(R)}_{XS} Q^{-1} {K^{(R)}_{XX}}^{-1} \end{bmatrix}$$

where $Q = K^{(R)}_{SS} - K^{(R)}_{SX} {K^{(R)}_{XX}}^{-1} K^{(R)}_{XS}$. Then, we substitute this expression into the matrix-product term $\check{K}_{X\check{S}} \check{K}^{-1}_{\check{S}\check{S}} \check{K}_{\check{S}X}$, alongside utilising the zero-blocks in first row/column of $\check{K}_{\check{S}\check{S}}$. Such a manipulation yields a very simple expression, as all the terms involving Q are cancelled out:

$$\begin{aligned} \check{K}_{X\check{S}} \check{K}^{-1}_{\check{S}\check{S}} \check{K}_{\check{S}X} &= K^{(A)}_{XS} {K^{(A)}_{SS}}^{-1} K^{(A)}_{SX} + \left[K^{(R)}_{XS}, K^{(R)}_{XX}\right] \begin{bmatrix} K^{(R)}_{SS} & K^{(R)}_{SX} \\ K^{(R)}_{XS} & K^{(R)}_{XX} \end{bmatrix}^{-1} \left[K^{(R)}_{XS}, K^{(R)}_{XX}\right]^T \\ &= K^{(A)}_{XS} {K^{(A)}_{SS}}^{-1} K^{(A)}_{SX} + K^{(R)}_{XX}. \end{aligned}$$

The $\mathcal{L}_1$ bound for the inducing point set $\check{S}$ can then be written as:

$$\begin{aligned} \mathcal{L}_1(\boldsymbol{z}; \Sigma, \check{S}) &= \log \mathcal{N}\left(\boldsymbol{z}|\mathbf{0}, K^{(A)}_{XS} {K^{(A)}_{SS}}^{-1} K^{(A)}_{SX} + K^{(R)}_{XX} + \sigma^2_{\boldsymbol{z}} I_N\right) - \frac{1}{2\sigma^2_{\boldsymbol{z}}} \text{tr}(K_{XX} - K^{(A)}_{XS} {K^{(A)}_{SS}}^{-1} K^{(A)}_{SX} - K^{(R)}_{XX}) \\ &= \log \mathcal{N}\left(\boldsymbol{z}|\mathbf{0}, K^{(A)}_{XS} {K^{(A)}_{SS}}^{-1} K^{(A)}_{SX} + \hat{\Sigma}\right) - \frac{1}{2\sigma^2_{\boldsymbol{z}}} \text{tr}(K^{(A)}_{XX} - K^{(A)}_{XS} {K^{(A)}_{SS}}^{-1} K^{(A)}_{SX}). \end{aligned} \tag{8}$$

Focusing on the last trace term we have,

$$\begin{aligned}
\mathrm{tr}\left(\hat{\Sigma}^{-1}\tilde{K}_{XX}^{(A)}\right) &= \mathrm{tr}\left((K_{XX}^{(R)}+\sigma_{\boldsymbol{z}}^2 I_N)^{-1}\tilde{K}_{XX}^{(A)}\right) = \mathrm{tr}\left((\sigma_{\boldsymbol{z}}^{-2}I_N - \sigma_{\boldsymbol{z}}^{-4}({K_{XX}^{(R)}}^{-1}+\sigma_{\boldsymbol{z}}^{-2}I_N)^{-1})\tilde{K}_{XX}^{(A)}\right) \\
&= \sigma_{\boldsymbol{z}}^{-2}\mathrm{tr}\left(\tilde{K}_{XX}^{(A)}\right) - \sigma_{\boldsymbol{z}}^{-4}\mathrm{tr}\left(({K_{XX}^{(R)}}^{-1}+\sigma_{\boldsymbol{z}}^{-2}I_N)^{-1}\tilde{K}_{XX}^{(A)}\right) \\
&= \sigma_{\boldsymbol{z}}^{-2}\mathrm{tr}\left(\tilde{K}_{XX}^{(A)}\right) - \sigma_{\boldsymbol{z}}^{-4}\mathrm{tr}\left(\hat{L}\hat{L}^T\tilde{K}_{XX}^{(A)}\right) = \sigma_{\boldsymbol{z}}^{-2}\mathrm{tr}\left(\tilde{K}_{XX}^{(A)}\right) - \sigma_{\boldsymbol{z}}^{-4}\mathrm{tr}\left(\hat{L}^T\tilde{K}_{XX}^{(A)}\hat{L}\right) \\
&\leq \sigma_{\boldsymbol{z}}^{-2}\mathrm{tr}\left(\tilde{K}_{XX}^{(A)}\right) \\
&= \frac{1}{\sigma_{\boldsymbol{z}}^2}\mathrm{tr}(K_{XX}^{(A)} - K_{XS}^{(A)}{K_{SS}^{(A)}}^{-1}K_{SX}^{(A)}),
\end{aligned} \tag{9}$$

where $\hat{L}$ is the Cholesky decomposition of $({K_{XX}^{(R)}}^{-1}+\sigma_{\boldsymbol{z}}^{-2}I_N)^{-1}$. As matrix ${K_{XX}^{(R)}}^{-1}+\sigma_{\boldsymbol{z}}^{-2}I_N$ is always positive definite, its inverse and corresponding Cholesky decomposition always exist. Additionally, we used the matrix trace cyclic rotation property and the fact that the trace of a positive semidefinite matrix is $\hat{L}^T\tilde{K}_{XX}^{(A)}\hat{L}$, which is always non-negative. Combining equations (8) and (9) we have

$$\begin{aligned}
\mathcal{L}_1(\boldsymbol{z};\Sigma,\check{S}) &= \log\mathcal{N}\left(\boldsymbol{z}|\boldsymbol{0},K_{XS}^{(A)}{K_{SS}^{(A)}}^{-1}K_{SX}^{(A)}+\hat{\Sigma}\right) - \frac{1}{2\sigma_{\boldsymbol{z}}^2}\mathrm{tr}(K_{XX}^{(A)} - K_{XS}^{(A)}{K_{SS}^{(A)}}^{-1}K_{SX}^{(A)}) \\
&\leq \log\mathcal{N}\left(\boldsymbol{z}|\boldsymbol{0},K_{XS}^{(A)}{K_{SS}^{(A)}}^{-1}K_{SX}^{(A)}+\hat{\Sigma}\right) - \frac{1}{2}\mathrm{tr}\left(\hat{\Sigma}^{-1}\tilde{K}_{XX}^{(A)}\right) \\
&= \mathcal{L}_2(\boldsymbol{z};\Sigma,S).
\end{aligned}$$

Consequently, the non-strict inequalities chain gives the final result:

$$\mathcal{L}_1(\boldsymbol{z};\Sigma,S) = \mathcal{L}_1(\boldsymbol{z};\Sigma,\hat{S}) \leq \mathcal{L}_1(\boldsymbol{z};\Sigma,\tilde{S}) = \mathcal{L}_1(\boldsymbol{z};\Sigma,\check{S}) \leq \mathcal{L}_2(\boldsymbol{z};\Sigma,S)$$

□

Computational complexity The computational complexity of the bound described in eq. (7), is $\mathcal{O}(\sum_{p=1}^P n_p^3 + NM^2)$ flops, which leads to an approximately similar computational complexity as the Titsias [2009] bound, when $n_p \simeq M \ll N$, but is significantly tighter. Here, we elucidate in detail how to perform the computations in eq. (7) to achieve such complexity. The first term of eq. (7) can be written as:

$$\begin{aligned}
\mathrm{tr}(\bar{\Sigma}^{-1}W) &= \mathrm{tr}(\hat{\Sigma}^{-1}W) - \mathrm{tr}\left(\hat{\Sigma}^{-1}K_{XS}^{(A)}\left[K_{SS}^{(A)}+K_{SX}^{(A)}\hat{\Sigma}^{-1}K_{XS}^{(A)}\right]^{-1}K_{SX}^{(A)}\hat{\Sigma}^{-1}W\right) \\
&= \sum_{p=1}^P\left(\mathrm{diag}(\hat{\Sigma}^{-1})\cdot\mathrm{diag}(W_p)\right) - \mathrm{tr}\left(V^{-1}\left(K_{SX}^{(A)}\hat{\Sigma}^{-1}W\hat{\Sigma}^{-1}K_{XS}^{(A)}\right)\right),
\end{aligned}$$

where $V = K_{SS}^{(A)} + K_{SX}^{(A)}\hat{\Sigma}^{-1}K_{XS}^{(A)}$. We have used the Woodbury matrix identity to obtain the first equality and the cyclic rotation property of the matrix trace to obtain the second equality [Press et al., 2007]. Since $\hat{\Sigma}$ is a block-diagonal matrix, obtaining $\hat{\Sigma}^{-1}$ takes $\mathcal{O}(\sum_{p=1}^P n_p^3)$ flops and obtaining products $K_{SX}^{(A)}\hat{\Sigma}^{-1}K_{XS}^{(A)}$ or $K_{SX}^{(A)}(\hat{\Sigma}^{-1}W\hat{\Sigma}^{-1})K_{XS}^{(A)}$ takes $\mathcal{O}(\sum_{p=1}^P n_p M^2) = \mathcal{O}(NM^2)$ flops. Moreover, inverting V takes $\mathcal{O}(M^3)$ flops.

We can use the Woodbury matrix identity to write the second term as,

$$\bar{\boldsymbol{\mu}}^T\bar{\Sigma}^{-1}\bar{\boldsymbol{\mu}} = \bar{\boldsymbol{\mu}}^T\hat{\Sigma}^{-1}\bar{\boldsymbol{\mu}} - \bar{\boldsymbol{\mu}}^T\hat{\Sigma}^{-1}K_{XS}^{(A)}V^{-1}K_{SX}^{(A)}\hat{\Sigma}^{-1}\bar{\boldsymbol{\mu}}.$$

Obtaining $\hat{\Sigma}^{-1}\bar{\boldsymbol{\mu}}$ takes $\mathcal{O}(\sum_{p=1}^P n_p^2)$ flops and obtaining $K_{SX}^{(A)}\left(\hat{\Sigma}^{-1}\bar{\boldsymbol{\mu}}\right)$ takes $\mathcal{O}(NM^2)$ flops.

For the forth term we use the generalised determinant lemma so that,

$$|\bar{\Sigma}| = |\hat{\Sigma}||K_{SS}^{(A)}|^{-1}|V|.$$

These determinant computations take $\mathcal{O}(\sum_{p=1}^P n_p^3)$, $\mathcal{O}(M^3)$, and $\mathcal{O}(M^3)$, respectively.

The fifth term is trivially $\mathcal{O}(N)$. We can again use the cyclic rotation property of the matrix trace to write the last term as:

$$\begin{aligned}\mathrm{tr}\left(\hat{\Sigma}_p^{-1}\tilde{K}^{(A)}_{X_pX_p}\right) &= \mathrm{tr}\left(\hat{\Sigma}_p^{-1}K^{(A)}_{X_pX_p}\right) - \mathrm{tr}\left(\hat{\Sigma}_p^{-1}K^{(A)}_{X_pS}{K^{(A)}_{SS}}^{-1}K^{(A)}_{SX_p}\right)\\ &= \mathrm{tr}\left(\hat{\Sigma}_p^{-1}K^{(A)}_{X_pX_p}\right) - \mathrm{tr}\left(\left(K^{(A)}_{SX_p}\hat{\Sigma}_p^{-1}K^{(A)}_{X_pS}\right){K^{(A)}_{SS}}^{-1}\right).\end{aligned}$$

The first trace takes $\mathcal{O}(n_p^2)$ to compute once the $\hat{\Sigma}^{-1}$ is available. It takes $\mathcal{O}(n_pM^2)$ to compute the product in parenthesis in the second trace, and $\mathcal{O}(M^2)$ for the trace itself once ${K^{(A)}_{SS}}^{-1}$ is available. Since we need to compute these traces for $p = 1, \ldots, P$, the overall complexity of the last term in eq. (7) is $\mathcal{O}(\sum_{p=1}^P n_p^2 + NM^2)$. Combining the terms we get the final time complexity:

$$\begin{aligned}\text{Complexity} &= \mathcal{O}(\sum_{p=1}^P n_p^3) + \mathcal{O}(NM^2) + \mathcal{O}(M^3) + \mathcal{O}(\sum_{p=1}^P n_p^2) + \mathcal{O}(NM^2)\\ &\quad + \mathcal{O}(\sum_{p=1}^P n_p^3) + \mathcal{O}(M^3) + \mathcal{O}(M^3) + \mathcal{O}(N) + \mathcal{O}(\sum_{p=1}^P n_p^2 + NM^2)\\ &= \mathcal{O}(\sum_{p=1}^P n_p^3 + NM^2). \qquad (10)\end{aligned}$$

3 Stochastic Variational Inference for longitudinal Gaussian process

A fundamental drawback of the methods described in the previous section is that they require using the full training dataset to compute the loss and perform a gradient step. This can be an issue for many problems with large data, such as sequences of images, electronic health records, etc. A common machine learning technique to tackle such problems is based on training the model using mini-batches. The mini-batch approach makes use of unbiased stochastic estimates of the loss and its gradients, which are computed based on a subset of the data. The subsets are chosen such that all training data points are used within an epoch. In this section, we first recall earlier work on how to adjust the bound of eq. (5) to become compatible with stochastic variational inference and mini-batching. Then, we modify this bound to account for the specifics of the GP covariance structure in L-VAE.

In contrast to Titsias [2009], Hensman et al. [2013] proposed to avoid analytical marginalisation of inducing values $\boldsymbol{u}$ in eq. (3). Instead, Hensman et al. [2013] proposed to explicitly keep track of its distribution, which is assumed to be Gaussian $\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{m}, H)$. Then, the authors derived an alternative evidence lower bound:

$$\begin{aligned}\log p(\boldsymbol{z}) &= \int_{\boldsymbol{u}}\int_{\boldsymbol{f}} p(\boldsymbol{z}|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{u})p(\boldsymbol{u})d\boldsymbol{f}d\boldsymbol{u} = \log\int_{\boldsymbol{u}} p(\boldsymbol{z}|\boldsymbol{u})\frac{p(\boldsymbol{u})}{q(\boldsymbol{u})}q(\boldsymbol{u})d\boldsymbol{u} \geq \int_{\boldsymbol{u}}\log p(\boldsymbol{z}|\boldsymbol{u})q(\boldsymbol{u})d\boldsymbol{u} - D_{KL}(q(\boldsymbol{u})||p(\boldsymbol{u}))\\ &\geq \int_{\boldsymbol{u}}\log\mathcal{N}(\boldsymbol{z}|K_{XS}K_{SS}^{-1}\boldsymbol{u}, \sigma_z^2 I_N)\mathcal{N}(\boldsymbol{u}|\boldsymbol{m}, H)d\boldsymbol{u} - \frac{1}{2\sigma_z^2}\mathrm{tr}(\tilde{K}) - D_{KL}(\mathcal{N}(\boldsymbol{m}, H)||\mathcal{N}(\boldsymbol{0}, K_{SS}))\\ &= \log\mathcal{N}(\boldsymbol{z}|K_{XS}K_{SS}^{-1}\boldsymbol{m}, \sigma_z^2 I_N) - \frac{1}{2\sigma_z^2}\mathrm{tr}(HK_{SS}^{-1}K_{SX}K_{XS}K_{SS}^{-1}) - \frac{1}{2\sigma_z^2}\mathrm{tr}(\tilde{K}) - D_{KL}(\mathcal{N}(\boldsymbol{m}, H)||\mathcal{N}(\boldsymbol{0}, K_{SS}))\\ &\triangleq \mathcal{L}_3. \qquad (11)\end{aligned}$$

Substituting this bound in eq. (1) yields:

$$
\begin{aligned}
D_{\mathrm{KL}}(\mathcal{N}(\bar{\boldsymbol{\mu}}, W)||\mathcal{N}(\mathbf{0}, \Sigma)) &\leq -\int_{\boldsymbol{z}} \mathcal{L}_3 \mathcal{N}(\boldsymbol{z}|\bar{\boldsymbol{\mu}}, W) d\boldsymbol{z} - \frac{1}{2}\log|(2\pi e)W| \\
&= -\int_{\boldsymbol{z}} \log \mathcal{N}(\boldsymbol{z}|K_{XS}K_{SS}^{-1}\boldsymbol{m}, \sigma_z^2 I_N)\mathcal{N}(\boldsymbol{z}|\bar{\boldsymbol{\mu}}, W) d\boldsymbol{z} + \frac{1}{2\sigma_z^2}\mathrm{tr}(\tilde{K}) \\
&\quad + \frac{1}{2\sigma_z^2}\mathrm{tr}(HK_{SS}^{-1}K_{SX}K_{XS}K_{SS}^{-1}) + D_{KL}(\mathcal{N}(\boldsymbol{m}, H)||\mathcal{N}(\mathbf{0}, K_{SS})) - \frac{1}{2}\log|(2\pi e)W| \\
&= \frac{1}{2}(K_{XS}K_{SS}^{-1}\boldsymbol{m} - \bar{\boldsymbol{\mu}})^T(\sigma_z^2 I_N)^{-1}(K_{XS}K_{SS}^{-1}\boldsymbol{m} - \bar{\boldsymbol{\mu}}) + \frac{1}{2}\mathrm{tr}((\sigma_z^2 I_N)^{-1}W) + \frac{N}{2}\log\sigma_z^2 \\
&\quad + \frac{N}{2}\log 2\pi + \frac{1}{2\sigma_z^2}\mathrm{tr}(\tilde{K}) + \frac{1}{2\sigma_z^2}\mathrm{tr}(HK_{SS}^{-1}K_{SX}K_{XS}K_{SS}^{-1}) \\
&\quad + D_{KL}(\mathcal{N}(\boldsymbol{m}, H)||\mathcal{N}(\mathbf{0}, K_{SS})) - \frac{1}{2}\log|(2\pi e)W| \\
&= \frac{1}{2}\Bigg(\sigma_z^{-2}\sum_{i=1}^{N}(K_{\boldsymbol{x}_i S}K_{SS}^{-1}\boldsymbol{m} - \bar{\mu}_i)^2 + \sigma_z^{-2}\sum_{i=1}^{N}\sigma_\phi^2(\boldsymbol{y}_i) + N\log\sigma_z^2 + \sigma_z^{-2}\sum_{i=1}^{N}\tilde{K}_{ii} \\
&\quad + \sigma_z^{-2}\sum_{i=1}^{N}\mathrm{tr}\left(\left(K_{SS}^{-1}HK_{SS}^{-1}\right)\left(K_{S\boldsymbol{x}_i}K_{\boldsymbol{x}_i S}\right)\right) - \sum_{i=1}^{N}\log\sigma_\phi^2(\boldsymbol{y}_i) - N\Bigg) \\
&\quad + D_{KL}(\mathcal{N}(\boldsymbol{m}, H)||\mathcal{N}(\mathbf{0}, K_{SS})) \\
&\triangleq \mathcal{D}_3
\end{aligned} \tag{12}
$$

Each term, except the last one, is additive over $i = 1, \ldots, N$. Therefore, replacing the sum over all $i = 1, \ldots, N$ with a batch-normalised partial sum over a subset of indices, $\mathcal{I} \subset \{1, \ldots, N\}$ of size $|\mathcal{I}| = \hat{N}$:

$$
\begin{aligned}
\hat{\mathcal{D}}_3 = \frac{1}{2}\frac{N}{\hat{N}}\sum_{i\in\mathcal{I}}\Big(\sigma_z^{-2}(K_{\boldsymbol{x}_i S}K_{SS}^{-1}\boldsymbol{m} - \bar{\boldsymbol{\mu}}_i)^2 + \sigma_z^{-2}\sigma_\phi^2(\boldsymbol{y}_i) + \sigma_z^{-2}\tilde{K}_{ii} + \sigma_z^{-2}\mathrm{tr}\left(\left(K_{SS}^{-1}HK_{SS}^{-1}\right)\left(K_{S\boldsymbol{x}_i}K_{\boldsymbol{x}_i S}\right)\right) - \log\sigma_\phi^2(\boldsymbol{y}_i)\Big) \\
+ \frac{N}{2}\log\sigma_z^2 - \frac{N}{2} + D_{KL}(\mathcal{N}(\boldsymbol{m}, H)||\mathcal{N}(\mathbf{0}, K_{SS})),
\end{aligned} \tag{13}
$$

is an unbiased estimate of the KL divergence upper bound $E_{\mathcal{I}\sim\mathfrak{S}\{1,\ldots,N\}}(\hat{\mathcal{D}}_3) = \mathcal{D}_3 \geq D_{\mathrm{KL}}(\mathcal{N}(\bar{\boldsymbol{\mu}}, W)||\mathcal{N}(\mathbf{0}, \Sigma))$. Here, $\mathfrak{S}\{1, \ldots, N\}$ is a uniform distribution over elements of arbitrary fixed partitions of set $\{1, \ldots, N\}$. This property enables us to use the mini-batching technique for the approximate computation of the KL divergence term of L-VAE and its gradients.

Similar to our criticism of the ELBO $\mathcal{L}_1$, the $\mathcal{L}_3$ is not well suited to the typical properties of GP covariance structures used for longitudinal modelling. Following the same notation as in the previous section, we introduce a modification to eq. (11):

$$
\begin{aligned}
\log p(\boldsymbol{z}) &= \int_{\boldsymbol{u}}\int_{\boldsymbol{f}} p(\boldsymbol{z}|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{u})p(\boldsymbol{u})d\boldsymbol{f}d\boldsymbol{u} = \log\int_{\boldsymbol{u}} p(\boldsymbol{z}|\boldsymbol{u})\frac{p(\boldsymbol{u})}{q(\boldsymbol{u})}q(\boldsymbol{u})d\boldsymbol{u} \geq \int_{\boldsymbol{u}}\log p(\boldsymbol{z}|\boldsymbol{u})q(\boldsymbol{u})d\boldsymbol{u} - D_{KL}(q(\boldsymbol{u})||p(\boldsymbol{u})) \\
&\geq \int_{\boldsymbol{u}}\log\mathcal{N}(\boldsymbol{z}|K_{XS}^{(A)}{K_{SS}^{(A)}}^{-1}\boldsymbol{u}, \hat{\Sigma})\mathcal{N}(\boldsymbol{u}|\boldsymbol{m}, H)d\boldsymbol{u} - \frac{1}{2}\sum_{p=1}^{P}\mathrm{tr}\left(\hat{\Sigma}_p^{-1}\tilde{K}_{X_pX_p}^{(A)}\right) - D_{KL}(\mathcal{N}(\boldsymbol{m}, H)||\mathcal{N}(\mathbf{0}, K_{SS}^{(A)})) \\
&= \log\mathcal{N}(\boldsymbol{z}|K_{XS}^{(A)}{K_{SS}^{(A)}}^{-1}\boldsymbol{m}, \hat{\Sigma}) - \frac{1}{2}\mathrm{tr}(H{K_{SS}^{(A)}}^{-1}K_{SX}^{(A)}\hat{\Sigma}^{-1}K_{XS}^{(A)}{K_{SS}^{(A)}}^{-1}) - \frac{1}{2}\sum_{p=1}^{P}\mathrm{tr}\left(\hat{\Sigma}_p^{-1}\tilde{K}_{X_pX_p}^{(A)}\right) \\
&\quad - D_{KL}(\mathcal{N}(\boldsymbol{m}, H)||\mathcal{N}(\mathbf{0}, K_{SS}^{(A)})) \\
&\triangleq \mathcal{L}_4
\end{aligned} \tag{14}
$$

Substituting this novel ELBO in eq. (1) yields:

$$
\begin{aligned}
D_{\mathrm{KL}}(\mathcal{N}(\bar{\boldsymbol{\mu}}, W)||\mathcal{N}(\mathbf{0}, \Sigma)) \leq & -\int_{\boldsymbol{z}} \mathcal{L}_4 \mathcal{N}(\boldsymbol{z}|\bar{\boldsymbol{\mu}}, W) d\boldsymbol{z} - \frac{1}{2}\log|(2\pi e)W| \\
= & -\int_{\boldsymbol{z}} \log \mathcal{N}(\boldsymbol{z}|K_{XS}^{(A)} {K_{SS}^{(A)}}^{-1}\boldsymbol{m}, \hat{\Sigma}) \mathcal{N}(\boldsymbol{z}|\bar{\boldsymbol{\mu}}, W) d\boldsymbol{z} + \frac{1}{2}\sum_{p=1}^{P} \mathrm{tr}\left(\hat{\Sigma}_p^{-1}\tilde{K}_{X_pX_p}^{(A)}\right) \\
& + \frac{1}{2}\mathrm{tr}(H {K_{SS}^{(A)}}^{-1} K_{SX}^{(A)} \hat{\Sigma}^{-1} K_{XS}^{(A)} {K_{SS}^{(A)}}^{-1}) + D_{KL}(\mathcal{N}(\boldsymbol{m}, H)||\mathcal{N}(\mathbf{0}, K_{SS}^{(A)})) \\
& - \frac{1}{2}\log|(2\pi e)W| \\
= & \frac{1}{2}(K_{XS}^{(A)} {K_{SS}^{(A)}}^{-1}\boldsymbol{m} - \bar{\boldsymbol{\mu}})^T \hat{\Sigma}^{-1}(K_{XS}^{(A)} {K_{SS}^{(A)}}^{-1}\boldsymbol{m} - \bar{\boldsymbol{\mu}}) + \frac{1}{2}\mathrm{tr}(\hat{\Sigma}^{-1}W) + \frac{1}{2}\log|\hat{\Sigma}| \\
& + \frac{N}{2}\log 2\pi + \frac{1}{2}\sum_{p=1}^{P} \mathrm{tr}\left(\hat{\Sigma}_p^{-1}\tilde{K}_{X_pX_p}^{(A)}\right) + \frac{1}{2}\mathrm{tr}(H {K_{SS}^{(A)}}^{-1} K_{SX}^{(A)} \hat{\Sigma}^{-1} K_{XS}^{(A)} {K_{SS}^{(A)}}^{-1}) \\
& + D_{KL}(\mathcal{N}(\boldsymbol{m}, H)||\mathcal{N}(\mathbf{0}, K_{SS}^{(A)})) - \frac{1}{2}\log|(2\pi e)W| \\
= & \frac{1}{2}\Bigg(\sum_{p=1}^{P}(K_{X_pS}^{(A)} {K_{SS}^{(A)}}^{-1}\boldsymbol{m} - \hat{\boldsymbol{\mu}}_p)^T \hat{\Sigma}_p^{-1}(K_{X_pS}^{(A)} {K_{SS}^{(A)}}^{-1}\boldsymbol{m} - \hat{\boldsymbol{\mu}}_p) \\
& + \sum_{p=1}^{P}\sum_{i=1}^{n_p}(\hat{\Sigma}_p^{-1})_{ii}\sigma_\phi^2(\boldsymbol{y}_{\mathcal{I}_{pi}}) + \sum_{p=1}^{P}\log|\hat{\Sigma}_p| + \frac{1}{2}\sum_{p=1}^{P}\mathrm{tr}\left(\hat{\Sigma}_p^{-1}\tilde{K}_{X_pX_p}^{(A)}\right) \\
& + \sum_{p=1}^{P}\mathrm{tr}\left(\left({K_{SS}^{(A)}}^{-1} H {K_{SS}^{(A)}}^{-1}\right)\left(K_{SX_p}^{(A)}\hat{\Sigma}_p^{-1}K_{X_pS}^{(A)}\right)\right) - \sum_{i=1}^{N}\log\sigma_\phi^2(\boldsymbol{y}_i) - N\Bigg) \\
& + D_{KL}(\mathcal{N}(\boldsymbol{m}, H)||\mathcal{N}(\mathbf{0}, K_{SS}^{(A)})) \\
\triangleq & \ \mathcal{D}_4,
\end{aligned} \tag{15}
$$

where $\mathcal{I}_{pi}$ is the index of the i^{th} sample for the p^{th} patient and $\hat{\boldsymbol{\mu}}_p = [\bar{\mu}_{\mathcal{I}_{p1}}, \ldots, \bar{\mu}_{\mathcal{I}_{pn_p}}]^T$ is a sub-vector of $\bar{\boldsymbol{\mu}}$ that corresponds to the p^{th} patient. Each term, except for the last one, is additive over $p = 1, \ldots, P$. Therefore, replacing the sum over all $p = 1, \ldots, P$ with a batch-normalised partial sum over a subset of indices $\mathcal{P} \subset \{1, \ldots, P\}$ of size $|\mathcal{P}| = \hat{P}$:

$$
\begin{aligned}
\hat{\mathcal{D}}_4 = & \frac{1}{2}\frac{P}{\hat{P}}\sum_{p\in\mathcal{P}}\Bigg((K_{X_pS}^{(A)} {K_{SS}^{(A)}}^{-1}\boldsymbol{m} - \hat{\boldsymbol{\mu}}_p)^T \hat{\Sigma}_p^{-1}(K_{X_pS}^{(A)} {K_{SS}^{(A)}}^{-1}\boldsymbol{m} - \hat{\boldsymbol{\mu}}_p) + \sum_{i=1}^{n_p}(\hat{\Sigma}_p^{-1})_{ii}\sigma_\phi^2(\boldsymbol{y}_{\mathcal{I}_{pi}}) + \log|\hat{\Sigma}_p| \\
& + \mathrm{tr}\left(\hat{\Sigma}_p^{-1}\tilde{K}_{X_pX_p}^{(A)}\right) + \mathrm{tr}\left(\left({K_{SS}^{(A)}}^{-1} H {K_{SS}^{(A)}}^{-1}\right)\left(K_{SX_p}^{(A)}\hat{\Sigma}_p^{-1}K_{X_pS}^{(A)}\right)\right) - \sum_{i=1}^{n_p}\log\sigma_\phi^2(\boldsymbol{y}_{\mathcal{I}_{pi}})\Bigg) \\
& - \frac{N}{2} + D_{KL}(\mathcal{N}(\boldsymbol{m}, H)||\mathcal{N}(\mathbf{0}, K_{SS}^{(A)})),
\end{aligned} \tag{16}
$$

is an unbiased estimate of the KL divergence upper bound $E_{\mathcal{P}\sim\mathfrak{S}\{1,\ldots,P\}}(\hat{\mathcal{D}}_4) = \mathcal{D}_4 \geq D_{\mathrm{KL}}(\mathcal{N}(\bar{\boldsymbol{\mu}}, W)||\mathcal{N}(\mathbf{0}, \Sigma))$. This property enables us to use the mini-batching technique for a more precise approximate computation of the KL divergence term of L-VAE and its gradients, by approximately splitting equal number of patients to each batch.

The learning of variational parameters $\boldsymbol{m}$ and H can be done either by explicitly parameterising and updating them within the overall optimisation scheme, or by using the natural gradients approach similar to Hensman et al. [2013]. Similar to the claims of the original paper, using the natural gradients can mitigate the challenge of finding a robust unconstrained parameterisation for the variational parameters. The gradients of KL divergence

unbiased estimate $\hat{\mathcal{D}}_4$ w.r.t. the variational parameters have the following form:

$$\frac{\partial \hat{\mathcal{D}}_4}{\partial \boldsymbol{m}} = -\sum_{p\in\mathcal{P}} {K_{SS}^{(A)}}^{-1} K_{SX_p}^{(A)} \hat{\Sigma}_p^{-1} \hat{\boldsymbol{\mu}}_p + \left(\sum_{p\in\mathcal{P}} {K_{SS}^{(A)}}^{-1} K_{SX_p}^{(A)} \hat{\Sigma}_p^{-1} K_{X_pS}^{(A)} {K_{SS}^{(A)}}^{-1} + {K_{SS}^{(A)}}^{-1} \right) \boldsymbol{m} \tag{17}$$

$$\frac{\partial \hat{\mathcal{D}}_4}{\partial H} = -\frac{1}{2} H^{-1} + \frac{1}{2} \sum_{p\in\mathcal{P}} {K_{SS}^{(A)}}^{-1} K_{SX_p}^{(A)} \hat{\Sigma}_p^{-1} K_{X_pS}^{(A)} {K_{SS}^{(A)}}^{-1} + \frac{1}{2} {K_{SS}^{(A)}}^{-1} \tag{18}$$

Making use of the chain rule, we can write the following:

$$\frac{\partial \mathcal{D}_4}{\partial \boldsymbol{\eta}} = \frac{\partial \mathcal{D}_4}{\partial [m, H]} \frac{\partial [m, H]}{\partial \boldsymbol{\eta}} = \begin{bmatrix} \frac{\partial \mathcal{D}_4}{\partial m} & \frac{\partial \mathcal{D}_4}{\partial H} \end{bmatrix} \begin{bmatrix} \frac{\partial m}{\partial \eta_1} & \frac{\partial m}{\partial \eta_2} \\ \frac{\partial H}{\partial \eta_1} & \frac{\partial H}{\partial \eta_2} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathcal{D}_4}{\partial m} & \frac{\partial \mathcal{D}_4}{\partial H} \end{bmatrix} \begin{bmatrix} I & 0 \\ -2m & I \end{bmatrix}$$

Then, using the update rule and following notation similar to Hensman et al. [2013], we get:

$$\begin{aligned}
\boldsymbol{\theta}_{2(t+1)} &= \boldsymbol{\theta}_{2(t)} - l \frac{\partial \hat{\mathcal{D}}_4}{\partial \eta_2} \\
\boldsymbol{\theta}_{2(t+1)} &= \boldsymbol{\theta}_{2(t)} - l \left(\frac{\partial \hat{\mathcal{D}}_4}{\partial \boldsymbol{m}} \frac{\partial \boldsymbol{m}}{\partial \eta_2} + \frac{\partial \hat{\mathcal{D}}_4}{\partial H} \frac{\partial H}{\partial \eta_2} \right) \\
\boldsymbol{\theta}_{2(t+1)} &= \boldsymbol{\theta}_{2(t)} - l \frac{\partial \hat{\mathcal{D}}_4}{\partial H} \\
-\frac{1}{2} H_{(t+1)}^{-1} &= -\frac{1}{2} H_{(t)}^{-1} - l \frac{\partial \hat{\mathcal{D}}_4}{\partial H} \\
H_{(t+1)} &= \left(H_{(t)}^{-1} + 2l \frac{\partial \hat{\mathcal{D}}_4}{\partial H} \right)^{-1}
\end{aligned} \tag{19}$$

and

$$\begin{aligned}
\boldsymbol{\theta}_{1(t+1)} &= \boldsymbol{\theta}_{1(t)} - l \frac{\partial \hat{\mathcal{D}}_4}{\partial \eta_1} \\
\boldsymbol{\theta}_{1(t+1)} &= \boldsymbol{\theta}_{1(t)} - l \left(\frac{\partial \hat{\mathcal{D}}_4}{\partial \boldsymbol{m}} \frac{\partial \boldsymbol{m}}{\partial \eta_1} + \frac{\partial \hat{\mathcal{D}}_4}{\partial H} \frac{\partial H}{\partial \eta_1} \right) \\
\boldsymbol{\theta}_{1(t+1)} &= \boldsymbol{\theta}_{1(t)} - l \left(\frac{\partial \hat{\mathcal{D}}_4}{\partial \boldsymbol{m}} - 2 \frac{\partial \hat{\mathcal{D}}_4}{\partial H} \boldsymbol{m} \right) \\
H_{(t+1)}^{-1} \boldsymbol{m}_{(t+1)} &= H_{(t)}^{-1} \boldsymbol{m}_{(t)} - l \left(\frac{\partial \hat{\mathcal{D}}_4}{\partial \boldsymbol{m}} - 2 \frac{\partial \hat{\mathcal{D}}_4}{\partial H} \boldsymbol{m}_{(t)} \right) \\
\boldsymbol{m}_{(t+1)} &= H_{(t+1)} \left(H_{(t)}^{-1} \boldsymbol{m}_{(t)} - l \left(\frac{\partial \hat{\mathcal{D}}_4}{\partial \boldsymbol{m}} - 2 \frac{\partial \hat{\mathcal{D}}_4}{\partial H} \boldsymbol{m}_{(t)} \right) \right)
\end{aligned} \tag{20}$$

4 Predictive distribution

The problem of obtaining the predictive distribution for the high-dimensional, out-of-sample data $\boldsymbol{y}_*$ given covariates $\boldsymbol{x}_*$ with L-VAE can be split into two parts: obtaining the predictive distribution of the latent representation $\boldsymbol{z}_*$ and propagating the obtained distribution through the probabilistic decoder $p(y_*|z_*)$. Given the training samples Y, covariate information X, the learnt parameters of the generative model $\omega = \{\psi, \theta\}$, and the inference

model ϕ, the predictive distribution follows:

$$\begin{aligned} p_\omega(\boldsymbol{y}_*|\boldsymbol{x}_*, Y, X) &= \int_{\boldsymbol{z}_*} p_\omega(\boldsymbol{y}_*|\boldsymbol{z}_*, \boldsymbol{x}_*, Y, X) p_\omega(\boldsymbol{z}_*|\boldsymbol{x}_*, Y, X) d\boldsymbol{z}_* \\ &= \int_{\boldsymbol{z}_*, Z} \underbrace{p_\psi(\boldsymbol{y}_*|\boldsymbol{z}_*)}_{\text{decode GP prediction}} \underbrace{p_\theta(\boldsymbol{z}_*|\boldsymbol{x}_*, Z, X)}_{\text{GP predictive posterior}} \underbrace{p_\omega(Z|Y, X)}_{\text{posterior of } Z} d\boldsymbol{z}_* dZ \\ &\approx \int_{\boldsymbol{z}_*, Z} \underbrace{p_\psi(\boldsymbol{y}_*|\boldsymbol{z}_*)}_{\text{decode GP prediction}} \underbrace{p_\theta(\boldsymbol{z}_*|\boldsymbol{x}_*, Z, X)}_{\text{GP predictive posterior}} \underbrace{q_\phi(Z|Y, X)}_{\text{encode training samples}} \quad d\boldsymbol{z}_* dZ. \end{aligned} \tag{21}$$

The true posterior $p_\omega(Z|Y, X)$ is intractable and, similar to the model inference or learning problem, it is replaced with the variational approximation defined by the inference model $q_\phi(Z|Y, X)$. Given such a substitution, the (approximate) predictive GP distribution for the latent representation is available in closed form,

$$\hat{p}_\omega(\boldsymbol{z}_*|\boldsymbol{x}_*, Y, X) = \int_Z p_\theta(\boldsymbol{z}_*|\boldsymbol{x}_*, Z, X) q_\phi(Z|Y, X) dZ = N(\boldsymbol{z}_*|\boldsymbol{\mu}_*, \Sigma_*).$$

Since, in the current work, we only consider multi-output GPs with diagonal cross-covariance functions and the output of probabilistic encoder $p_\psi(\boldsymbol{y}_n|\boldsymbol{z}_n)$ is restricted to be a multivariate normal distribution with diagonal covariance matrix, the predictive distribution in the latent space also factorises across the latent dimensions $N(\boldsymbol{z}_*|\boldsymbol{\mu}_*, \Sigma_*) = \prod_{l=1}^{L} N(z_{*l}|\mu_{*l}, \sigma^2_{*l})$ with:

$$\begin{aligned} \mu_{*l} &= K^{(l)}_{\boldsymbol{x}_* X} \Sigma_l^{-1} \bar{\boldsymbol{\mu}}_l \\ \sigma^2_{*l} &= k^{(l)}_{\boldsymbol{x}_* \boldsymbol{x}_*} - K^{(l)}_{\boldsymbol{x}_* X} \Sigma_l^{-1} K^{(l)}_{X \boldsymbol{x}_*} + K^{(l)}_{\boldsymbol{x}_* X} \Sigma_l^{-1} W_l \Sigma_l^{-1} K^{(l)}_{X \boldsymbol{x}_*} + \sigma^2_{zl}, \end{aligned}$$

where $k^{(l)}_{\boldsymbol{x}_* \boldsymbol{x}_*} = K^{(l)}_{\boldsymbol{x}_* \boldsymbol{x}_*} = \sum_{r=1}^{R} K^{(l,r)}_{\boldsymbol{x}_* \boldsymbol{x}_*}(\theta_l^{(r)})$ and $K^{(l)}_{\boldsymbol{x}_* X} = K^{(l)T}_{X \boldsymbol{x}_*} = \sum_{r=1}^{R} K^{(l,r)}_{\boldsymbol{x}_* X}(\theta_l^{(r)})$ (see eq. (4) in the main text), and $\Sigma_l = \sum_{r=1}^{R} K^{(l,r)}_{XX}(\theta_l^{(r)}) + \sigma^2_{zl} I$, $W_l = \text{diag}(\sigma^2_{\phi,l}(\boldsymbol{y}_1), \ldots, \sigma^2_{\phi,l}(\boldsymbol{y}_N))$ and $\bar{\boldsymbol{\mu}}_l = [\mu_{\phi,l}(\boldsymbol{y}_1), \ldots, \mu_{\phi,l}(\boldsymbol{y}_N)]^T$ (as in eq. (8) in the main text). Incorporating this property with eq. (21) leads to,

$$p_\omega(\boldsymbol{y}_*|\boldsymbol{x}_*, Y, X) \approx \int_{\boldsymbol{z}_*} \prod_{d=1}^{D} \mathcal{N}\left(y_{*d}|g_{\psi,d}(\boldsymbol{z}_*), \sigma^2_{yd}\right) \prod_{l=1}^{L} \mathcal{N}(z_{*l}|\mu_{*l}, \sigma^2_{*l}) d\boldsymbol{z}_*.$$

5 Scalable predictive distribution

Computing the predictive distribution, as described above, requires performing cubic operations over the $N \times N$ matrices, which makes it practically infeasible for problems with large training data. In this section, we exploit the same inducing points paradigm that was used for $\mathcal{D}_2$ in eq. (7) to alleviate the cubic complexity. For simplicity of notation, we omit the latent dimension index l in the rest of the section. We use the same notation as in eq. (6) and set $U = K^{(A)}_{SS} + K^{(A)}_{SX} \hat{\Sigma}^{-1} K^{(A)}_{XS}$. Let $X_* = [\boldsymbol{x}_{1*}, \ldots, \boldsymbol{x}_{N'*}]$ denote a collection of N' test data points from P' many subjects. Then,

$$\begin{aligned} \bar{\boldsymbol{\mu}}_* = K_{X_* X} \Sigma^{-1} \bar{\boldsymbol{\mu}} &= \left(K^{(A)}_{X_* S} {K^{(A)}_{SS}}^{-1} K^{(A)}_{SX} + K^{(R)}_{X_* X}\right) \left(K^{(A)}_{XS} {K^{(A)}_{SS}}^{-1} K^{(A)}_{SX} + \hat{\Sigma}\right)^{-1} \bar{\boldsymbol{\mu}} \\ &= \left(K^{(A)}_{X_* S} {K^{(A)}_{SS}}^{-1} K^{(A)}_{SX} + K^{(R)}_{X_* X}\right) \left(\hat{\Sigma}^{-1} \bar{\boldsymbol{\mu}} - \hat{\Sigma}^{-1} K^{(A)}_{XS} U^{-1} K^{(A)}_{SX} \hat{\Sigma}^{-1} \bar{\boldsymbol{\mu}}\right) \end{aligned} \tag{22}$$

In practice, the efficient computations are executed in the following order:

1. Compute $\hat{\Sigma}^{-1} \bar{\boldsymbol{\mu}}$; exploiting the block-diagonal property of $\hat{\Sigma}^{-1}$.
2. Compute $K^{(A)}_{XS} U^{-1} K^{(A)}_{SX} (\hat{\Sigma}^{-1} \bar{\boldsymbol{\mu}})$ using the low-rank properties of all matrices involved.
3. Compute $\tilde{\boldsymbol{\mu}} = \hat{\Sigma}^{-1} \bar{\boldsymbol{\mu}} - \hat{\Sigma}^{-1} K^{(A)}_{XS} U^{-1} K^{(A)}_{SX} \hat{\Sigma}^{-1} \bar{\boldsymbol{\mu}}$.
4. Compute $K^{(A)}_{X_* S} {K^{(A)}_{SS}}^{-1} K^{(A)}_{SX} \tilde{\boldsymbol{\mu}}$ using the low-rank properties of all matrices involved.

5. Compute $K^{(R)}_{X_*X}\tilde{\boldsymbol{\mu}}$. This relies on the fact that the $K^{(R)}_{X_*X}$ matrix will be very sparse, which can be exploited either via sparse matrix operations or simply by cycling over the sets of rows that correspond to each subject and performing the multiplication only for the non-zero part of $K^{(R)}_{X_*X}$ (if none, this product will be zero for the given subject).

6. Finally, sum up $\bar{\boldsymbol{\mu}}_* = K^{(A)}_{X_*S}{K^{(A)}_{SS}}^{-1}K^{(A)}_{SX}\tilde{\boldsymbol{\mu}} + K^{(R)}_{X_*X}\tilde{\boldsymbol{\mu}}$

In practice, the predictive distribution will be subject to the inducing point locations S and is guaranteed to be exact if and only if $\boldsymbol{u}$ is a sufficient statistic of $\boldsymbol{f}_A$.

The predictive mean can also be expressed in terms of the variational parameters:

$$K^{(A)}_{X_*S}{K^{(A)}_{SS}}^{-1}\boldsymbol{m} + K^{(R)}_{X_*X}\hat{\Sigma}^{-1}(\bar{\boldsymbol{\mu}} - K^{(A)}_{XS}{K^{(A)}_{SS}}^{-1}\boldsymbol{m})$$

Also, the predictive covariance can be expressed as follows:

$$\left(K^{(A)}_{X_*S} - K^{(R)}_{X_*X}\hat{\Sigma}^{-1}K^{(A)}_{XS}\right){K^{(A)}_{SS}}^{-1}H{K^{(A)}_{SS}}^{-1}\left(K^{(A)}_{X_*S} - K^{(R)}_{X_*X}\hat{\Sigma}^{-1}K^{(A)}_{XS}\right)^T + \sigma_z^2 I_{N'} + K^{(R)}_{X_*X_*} - K^{(R)}_{X_*X}\hat{\Sigma}^{-1}K^{(R)}_{XX_*}$$

6 Downstream classification task in the healthcare data experiment

Given the test set, our objective is to predict the patient *mortality*. Since our proposed model is generative in nature, to allow such an objective, we shall include the *mortality* covariate in the additive GP prior. The downstream classification task is done by computing the probability that $mortality = 0$ and $mortality = 1$. Given the learnt L-VAE model with fixed parameters ϕ, ψ, θ, we can obtain the probability for the binary mortality event for each patient (X_*, Y_*) in the test set as,

$$P(\text{mortality} = 0) = \frac{\exp(\mathcal{L}(\phi, \psi, \theta; Y_*, X_*, \text{mortality} = 0))}{\sum_{i=0}^{1}\exp(\mathcal{L}(\phi, \psi, \theta; Y_*, X_*, \text{mortality} = i))},$$
$$P(\text{mortality} = 1) = 1 - P(\text{mortality} = 0).$$

Furthermore, we also introduce a time to mortality covariate (or *mortalityTime*) that is based on the survival time. This covariate is relevant for individuals whose $mortality = 1$ and is treated as missing for the individuals that survive. However, in the testing phase, the exact mortality time is not known as our objective is to perform classification based on *mortality*. To overcome this problem, we estimate the distribution of the *mortalityTime* covariate based on the values of the covariate in the training set and, in the test, we compute the expectation of the ELBO described in eq. (5) of the main manuscript w.r.t. to that distribution.

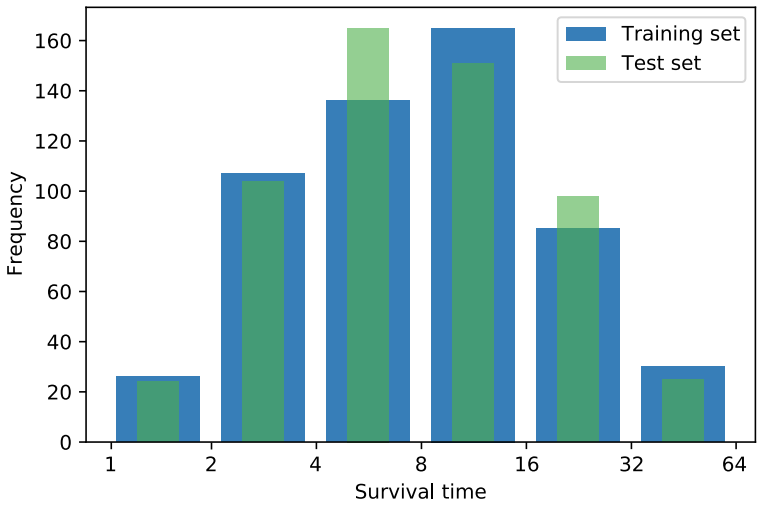

Figure 1: Histogram of survival times in training and test set

We approximate the expectation with a finite weighted average. Concretely, we first allocate the *mortalityTime* covariate values in the training set into B bins based on a logarithmic scale. Let α_i be the bin count and t_i be the average value of the *mortalityTime* in the i^{th} bin. The proportion of *mortalityTime* values in bin i is $w_i = \alpha_i/(\sum_{i=1}^{B} \alpha_i)$. The weighted ELBO is then computed as:

$$\mathcal{L}(\phi, \psi, \theta; Y, X, \text{mortality} = 1) = \sum_{i=1}^{B} w_i \cdot \mathcal{L}(\phi, \psi, \theta; Y, X, \text{mortality} = 1, t_i),$$

where the ELBO terms are now explicitly conditioned by the *mortalityTime*, t_i. We use $B = 6$ in our analysis. The histograms of the survival times in the training and test data are shown in Fig. 1.

We follow the data preprocessing steps described in Luo et al. [2018], and standardise the measurements of the 35 different attributes.

7 Optimisation and practical considerations

We make use of a suitable stochastic optimisation technique to minimise the ELBO in eq. (4) of the main manuscript. The parameters that we need to optimise include the neural network weights (ϕ and ψ) and kernel parameters (θ) of the multi-output additive GPs. In particular, the optimisation is done using the Adam optimiser [Kingma and Ba, 2015], which is an adaptive learning rate method that maintains an exponentially decaying average of past gradients as well as squared gradients. In case of mini-batch training, the Adam steps are conducted interchangeably with natural gradient-based updates of the variational parameters. For the inference implementation, we make use of PyTorch [Paszke et al., 2019] which allows the computation of derivatives using automatic differentiation.

In all the experiments, we first pre-train the neural networks with a standard normal distribution as the prior on the latent space (standard VAE [Kingma and Welling, 2014]) for 1000 epochs. This is followed by training the L-VAE model using the pre-trained encoder and decoder networks as initial values for ϕ and ψ, respectively. While training the L-VAE model, we monitor the loss on the validation (independent) dataset as a performance metric. Similar to the strategy of early stopping, we save the weights of the model that has the best performance on our defined metric. These model weights are chosen to perform predictions and other downstream tasks. However, to handle the possibility of a local minimum, we do not specify a stopping criterion and continue the training procedure till a predefined number of epochs has been performed. In the Rotated MNIST and Health MNIST experiments, L-VAE is trained for a maximum of 2000 epochs. Moreover, in the Healthcare data experiment, L-VAE is trained for a maximum of 1000 epochs.

8 Supplementary tables

Table 1 describes the neural network architecture used for the Rotated MNIST experiment. The hyperparameter choices are similar to Casale et al. [2018]. The architecture used for the Health MNIST experiment is described in table 2. We have tried to replicate the hyperparameter choices from Fortuin et al. [2020] for this experiment. For the Physionet Challenge 2012 dataset, we did not make use of a convolutional neural network (CNN) as was done for the Rotated MNIST as well as Health MNIST experiments because CNNs are more appropriate for image based (visual) data where the regional correlation (receptive field) of the measured values is important [Goodfellow et al., 2016]. Table 3 describes the architecture for the multi layered perceptron (MLP) that we used for the Physionet Challenge 2012 dataset. It is similar to the architecture used in Fortuin et al. [2020].

	Hyperparameter	Value
Inference network	Dimensionality of input	28×28
	Number of convolution layers	3
	Number of filters per convolution layer	72
	Kernel size	3×3
	Stride	2
	Number of feedforward layers	1
	Width of feedforward layers	128
	Dimensionality of latent space	L
	Activation function of layers	ELU
Generative network	Dimensionality of input	L
	Number of transposed convolution layers	3
	Number of filters per transposed convolution layer	72
	Kernel size	3×3
	Stride	1
	Number of feedforward layers	1
	Width of feedforward layers	128
	Activation function of layers	ELU

Table 1: Neural network architectures used in the Rotated MNIST dataset.

	Hyperparameter	Value
Inference network	Dimensionality of input	36×36
	Number of convolution layers	2
	Number of filters per convolution layer	144
	Kernel size	3×3
	Stride	2
	Pooling	Max pooling
	Pooling kernel size	2×2
	Pooling stride	2
	Number of feedforward layers	2
	Width of feedforward layers	300, 30
	Dimensionality of latent space	L
	Activation function of layers	RELU
Generative network	Dimensionality of input	L
	Number of transposed convolution layers	2
	Number of filters per transposed convolution layer	256
	Kernel size	4×4
	Stride	2
	Number of feedforward layers	2
	Width of feedforward layers	30, 300
	Activation function of layers	RELU

Table 2: Neural network architectures used in the Health MNIST dataset.

	Hyperparameter	Value
Inference network	Dimensionality of input	35
	Number of feedforward layers	2
	Number of elements in each feedforward layer	128, 64
	Dimensionality of latent space	L
	Activation function of layers	RELU
Generative network	Dimensionality of input	L
	Number of feedforward layers	2
	Number of elements in each feedforward layer	64, 128
	Activation function of layers	RELU

Table 3: Neural network architectures used in the Physionet Challenge 2012 dataset.

9 Supplementary images

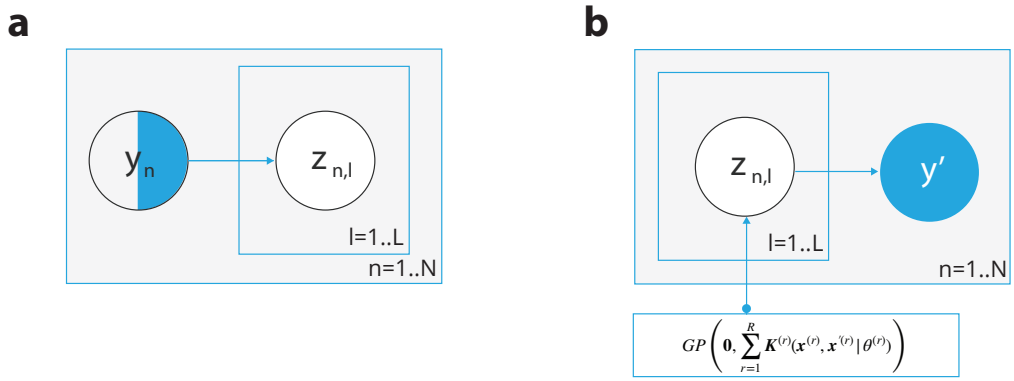

Figure 2: Plate diagram of the model. The shaded circle refers to an observed variable, the partially shaded circle refers to a partially observed variable (due to missing values), and the un-shaded circle refers to an un-observed variable. **(a)** Represents the inference (or encoder) model and **(b)** Represents the generative (or decoder) model.

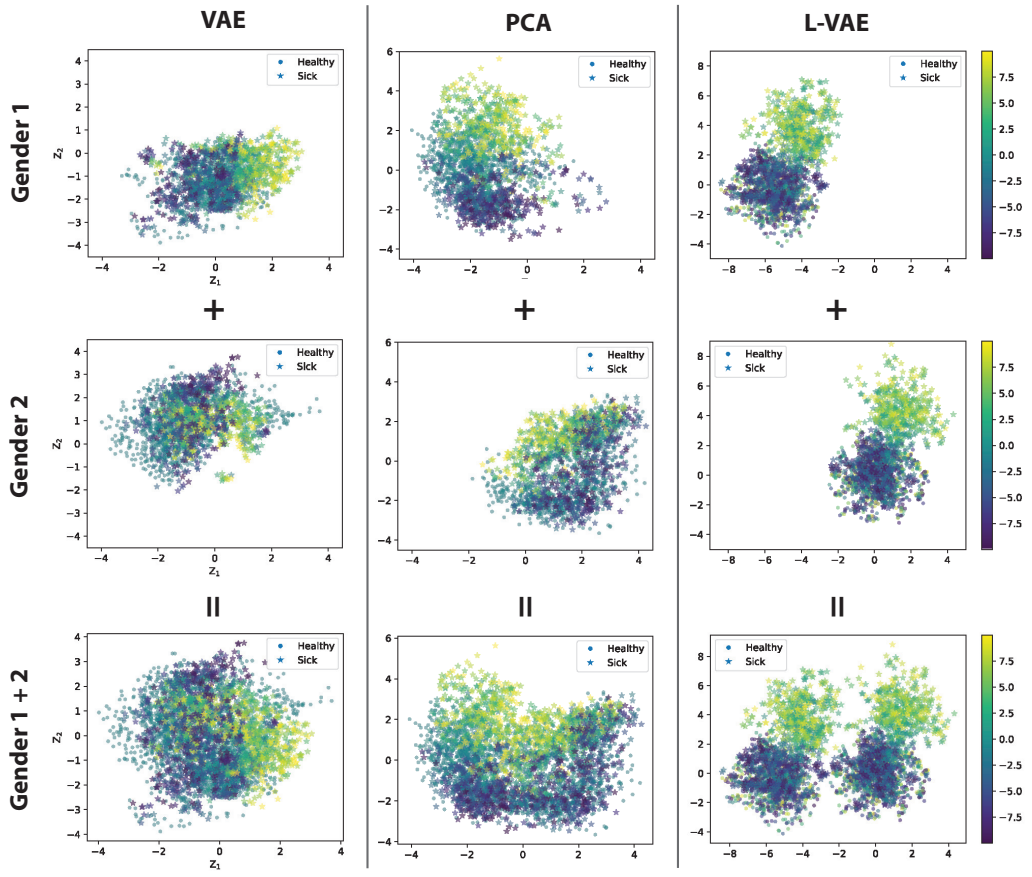

Figure 3: Comparison of the resulting latent space using VAE, PCA, and L-VAE on the Health MNIST dataset. The L-VAE model is fit using $\boldsymbol{f}_{\text{ca}}(\text{id}) + \boldsymbol{f}_{\text{se}}(\text{age}) + \boldsymbol{f}_{\text{ca}\times\text{se}}(\text{id} \times \text{age}) + \boldsymbol{f}_{\text{ca}\times\text{se}}(\text{sex} \times \text{age}) + \boldsymbol{f}_{\text{ca}\times\text{se}}(\text{diseasePresence} \times \text{diseaseAge})$ as the multi-output additive GP prior. The number of latent dimensions is set to 2. The points are coloured according to the diseaseAge as shown in the colour bar.

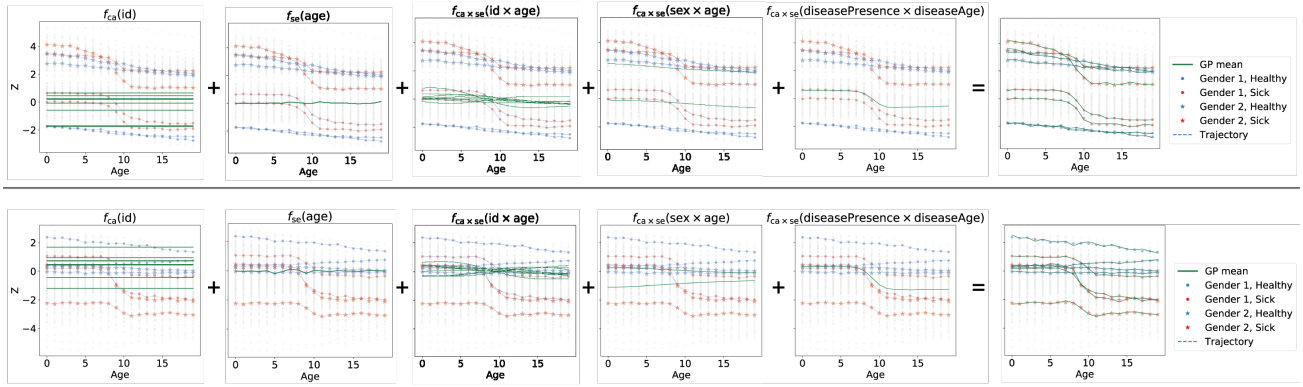

Figure 4: GP model fittings of L-VAE in the latent space with dimension 2 on the Health MNIST dataset.

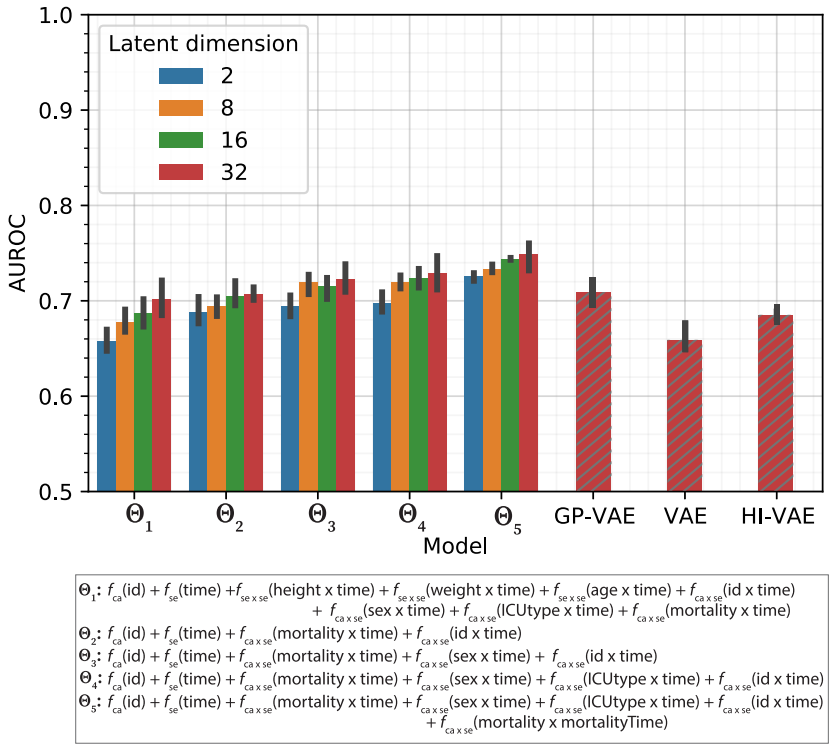

Figure 5: AUROC scores for the patient mortality prediction task on the test set of the Physionet Challenge 2012 dataset. This is an extension to Fig. 4 in the main manuscript. In this figure, we can observe L-VAE's performance with different latent dimensions as well as all the patient-specific general auxiliary covariates. Higher AUROC score is better.

References

F. P. Casale, A. V. Dalca, L. Saglietti, J. Listgarten, and N. Fusi. Gaussian process prior variational autoencoders. In *Advances in Neural Information Processing Systems, NeurIPS*, 2018.

L. Cheng, S. Ramchandran, T. Vatanen, N. Lietzén, R. Lahesmaa, A. Vehtari, and H. Lähdesmäki. An additive gaussian process regression model for interpretable non-parametric analysis of longitudinal data. *Nature Communications*, 2019.

V. Fortuin, D. Baranchuk, G. Rätsch, and S. Mandt. GP-VAE: deep probabilistic time series imputation. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS*. PMLR, 2020.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, UAI*. AUAI Press, 2013.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR*, 2015.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR*, 2014.

Y. Luo, X. Cai, Y. Zhang, J. Xu, and X. Yuan. Multivariate time series imputation with generative adversarial networks. In *Advances in Neural Information Processing Systems, NeurIPS*, 2018.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems, NeurIPS*, 2019.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, USA, 3 edition, 2007.

C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

M. K. Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, AISTATS*, 2009.