Latent variable model for high-dimensional point process with structured missingness

Maksim Sinelnikov [1] **Manuel Haussmann** [1 2] **Harri Lähdesmäki** [1]

Abstract

Longitudinal data are important in numerous fields, such as healthcare, sociology, and seismology, but real-world datasets present notable challenges for practitioners because they can be high-dimensional, contain structured missingness patterns, and measurement time points can be governed by an unknown stochastic process. While various solutions have been suggested, the majority of them have been designed to account for only one of these challenges. In this work, we propose a flexible and efficient latent-variable model that is capable of addressing all these limitations. Our approach utilizes Gaussian processes to capture temporal correlations between samples and their associated missingness masks as well as to model the underlying point process. We construct our model as a variational autoencoder together with deep neural network parameterised encoder and decoder models and develop a scalable amortised variational inference approach for efficient model training. We demonstrate competitive performance using both simulated and real datasets.

1. Introduction

Longitudinal data arise in many domains such as healthcare, sociology and seismology (Liu, 2015). These datasets consist of repeated measurements of unique instances, e.g. patients, collected over time. However, real-world applications pose several challenges for practitioners: measurements are typically high-dimensional and contain non-trivial missingness patterns, and time points of the observations are not deterministic but rather arise from an unknown stochastic process. These challenges are characteristic of many real biomedical datasets, such as electronic health records.

Variational autoencoders (VAEs) have become a popular approach to model high-dimensional data (Kingma & Welling, 2014; Rezende et al., 2014). However, a notable limitation of standard VAEs is their assumption that the latent variables factorize across samples, hence ignoring correlations between observations and making the models inappropriate for temporal and longitudinal datasets. Several recent works (Casale et al., 2018; Fortuin et al., 2020; Ramchandran et al., 2021) have addressed this issue by incorporating Gaussian process (GP) priors for these latent variables, creating a probabilistic model that is capable of modelling arbitrary correlations between latent encodings.

The simplest form of missingness is missing completely at random (MCAR), i.e., the missingness pattern is independent of the observed and unobserved data. While it is generally feasible to handle MCAR in most latent-variable models, more complex patterns of structured missingness (Rubin, 1976; Mitra et al., 2023) require additional modeling capacities. Extending VAEs to be able to model various structured missingness patterns has recently become the focus of several papers (Collier et al., 2020; Ipsen et al., 2021; Ghalebikesabi et al., 2021). However, these methods still lack the ability to model correlations among observations or missingness masks, thus limiting their applicability to temporal data.

While VAEs have been applied to various biomedical datasets, the existing methods cannot consider observation time points as random variables. Instead, they have to treat time as a deterministic covariate and, therefore, loose useful information embedded in its stochastic nature. A separate line of research has proposed methods to model unknown temporal point processes primarily using GP-based methods (Lloyd et al., 2015; Liu & Hauskrecht, 2019). Overall, the field lacks versatile modeling methods that would allow modeling high-dimensional, marked point-processes that may be corrupted by structured missingness.

Contributions. In this work, we propose a novel deep latent variable model (DLVM), that is specifically designed to capture structured missingness and uses temporal point processes to model time. We construct the model by intro-

[1]Department of Computer Science, Aalto University, Espoo, Finland [2]Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark. Correspondence to: Maksim Sinelnikov <maksim.sinelnikov@aalto.fi>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

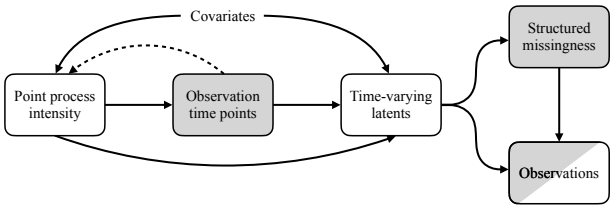

Figure 1. Conceptual overview of our model. Shaded, partially-shaded, and blank rectangles refer to observed, partially observed, and latent components. Dashed arrow corresponds to the dependence of the point process intensity on the previous time points.

ducing three sets of latent variables with GP priors, which model observations, missingness masks and point process. To adapt the model for longitudinal data, we rely on longitudinal additive kernels (Ramchandran et al., 2021) for the latent representations of observations and masks. Additionally, to use the information embedded in the temporal point process, we provide the intensity of the point process as an additional input to the GP kernel functions of observations and missingness masks. See Figure 1 for a high-level summary of our model and Appendix G for an extended visualization. We present two variations of our proposal, the simplified *longitudinal latent variable model with structured missingness* (LLSM) without a temporal point process, and the full model, *longitudinal latent variable model for high-dimensional point process with structured missingness* (LLPPSM). To summarize, our contributions are that

(i) we present a latent variable model able to capture structured missingness in the context of longitudinal data;

(ii) we extend this model by a temporal point process and use the inferred intensity of the process as an additional input to the model;

(iii) we compare the performance of our two model variants against baseline methods on several datasets and report state-of-the-art results on a variety of tasks.

2. Related Work

We summarize and compare previous methods in Table 1.

Challenges with missing data. In his pioneering work, Rubin (1976) identified three classes of missing data: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). For MCAR, the missingness mechanism is independent of both observed and unobserved variables. In case of MAR, the missingness depends on the observed attributes. Whereas, if data is MNAR, missing readings are dependent on the unobserved data, or systematic factors that are not accounted for in the experiment. The last two are examples of structured missingness. Although MCAR case can be handled by simply excluding missing elements from the analysis without introducing a bias, the same does not hold for the other scenarios. Despite the utility of Rubin's taxonomies, Mitra et al. (2023) emphasized that they do not fully account for high-dimensional patterns of structured missingness, frequently encountered in modern ML applications. They also proposed a set of current grand challenges in learning from data with structured missingness and claimed that the field of learning with missing values needs to be further advanced.

DLVMs for missing data. Various methods have been proposed to address the challenge of missing values within generative models. Collier et al. (2020) employed a variational autoencoder by concatenating the input with a missingness mask. While this approach can model MAR and MNAR scenarios, it fails to model temporal correlations and does not take into account auxiliary covariate information. Mattei & Frellsen (2019) used importance sampling to derive a Missing Importance Weighted Autoencoder (MIWAE) bound for training DLVMs under MAR condition. Building upon this work, Ipsen et al. (2021) expanded this method to MNAR scenario by directly modeling the missingness mask. However, both these approaches lack the ability to model temporal correlation in the latent space, hence they are not suitable for longitudinal setting.

Temporal DLVMs. To model temporal data, a set of methodologies has emerged, exploring the use of GP priors for latent variables. Casale et al. (2018) introduced the GPP-VAE model to integrate both view and object information in a GP prior through the product kernel. Fortuin et al. (2020) proposed the GP-VAE that assigns individual GP prior for the time-series of each subject. While these methods allow to model subject-specific temporal structure, they have limited or no functionality to account for possible other auxiliary covariates. Ramchandran et al. (2021) introduced L-VAE, a model that uses a multi-output additive GP prior and is well-suited for longitudinal data by leveraging carefully designed interaction kernels. The main drawback of all these approaches is their limitation to MCAR modeling which is naïve in many domains, such as healthcare.

Another direction of research to deal with temporal data focuses on recurrent neural networks (RNNs). Che et al. (2018) developed GRU-D which incorporates *masking* and a *time interval* into a deep model architecture, making it possible to model structured missingness patterns. They also proposed to use a decaying mechanism to handle irregularly-sampled timestamps. Luo et al. (2018) extended this model for time-series imputation by employing generative adversarial networks (GRUI-GAN). However, it is not straightforward to incorporate auxiliary information into these models.

Table 1. A summary of related methods.

Model	Temporal structure	Other covariates	Structured missingness	Modelling timestamps	Generative	Reference
VAE missing	✗	✗	✓	✗	✓	Collier et al. (2020)
not-MIWAE	✗	✗	✓	✗	✓	Ipsen et al. (2021)
GRUI-GAN	✓	✗	✓	✗	✓	Luo et al. (2018)
GPPVAE	✓	Limited	✗	✗	✓	Casale et al. (2018)
GP-VAE	✓	✗	✗	✗	✓	Fortuin et al. (2020)
L-VAE	✓	✓	✗	✗	✓	Ramchandran et al. (2021)
GPRPP	✓	✗	✗	✓	Limited to timestamps	Liu & Hauskrecht (2019)
LLSM	✓	✓	✓	✗	✓	This work
LLPPSM	✓	✓	✓	✓	✓	This work

Longitudinal data analysis. An additional line of research that gained popularity in recent years refers to modeling of longitudinal data. Several works have focused on addressing the dependence between timestamps and longitudinal observations in order to avoid bias during the inference (Pullenayegum & Lim, 2016; Xu et al., 2022; Sang et al., 2022). However, although previous methods take into account auxiliary covariate information, they have two limitations. First, they are not applicable for the high-dimensional setting considered in our work as these previous methods were derived for one-dimensional case and employ purely statistical techniques. Second, to the best of our knowledge, the previous methods do not assume missing data mechanisms to depend on timestamps.

Temporal Point Process. Modelling temporal point processes has been a subject of several studies in recent years. Classical statistical approaches (Puri & Tuan, 1986) use maximum likelihood estimation to infer the parameters of a model. For this, they require specifying a parametric form of the intensity function which significantly limits their applications. Neural network-based models typically employ RNNs (Du et al., 2016). Lloyd et al. (2015) proposed to model intensity function with Gaussian processes and a squared link function that leads to closed-form solution. John & Hensman (2018) extended this model to variational Fourier features. Both approaches use the current time point, and don't take the previous history of events into account which limits them to model only inhomogeneous Poisson processes, a potentially unrealistic assumption in real-world scenarios. Liu & Hauskrecht (2019) overcome this problem by incorporating the previous D timestamps into the computation of the GP kernel function.

3. Methods

3.1. Background

Problem setup. We are given $N = \sum_{p=1}^{P} n_p$ observations, where P denotes the number of unique instances, e.g., patients, and n_p is the number of observations of instance p. The longitudinal *response variables* (or *marks*) of instance p are denoted as $\boldsymbol{y}_p = [y_1^p, \dots, y_{n_p}^p] \in \mathbb{R}^{K \times n_p}$, where each sample $y_i^p \in \mathcal{Y} = \mathbb{R}^K$. Each sample $y_i \in \mathcal{Y}$ can be split into observed and missing parts, y_i^{o} and y_i^{m}, with a corresponding *binary mask* $m_i \in \{0,1\}^K$ specifying which features of y_i are missing (1 is observed, 0 is missing). The *auxiliary covariate* information of instance p is $\boldsymbol{x}_p = [x_1^p, \dots, x_{n_p}^p]$, where each $x_i^p \in \mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_Q$ is a Q-dimensional vector. Covariates can be both discrete and continuous and represent, for instance, a patient's age, their gender, etc. For notational convenience, we separately denote the measurement *time points* of the subject p as $\boldsymbol{t}^p = [t_1^p, \dots, t_{n_p}^p]^T$ modeled as random variables, and $\mathcal{X}^{\text{static}}$ as the set of covariates that do not depend on time, such that $t \subset \mathcal{X}$ and $\mathcal{X}^{\text{static}} \subset \mathcal{X}$. Observations from all P instances, e.g., patients, form our longitudinal data matrix $\boldsymbol{y}$, matrix of missing values $\boldsymbol{m}$, covariate matrix $\boldsymbol{x}$, and vector of timestamps $\boldsymbol{t}$, defined respectively as

$$
\begin{aligned}
\boldsymbol{y} &= [\boldsymbol{y}_1, \dots, \boldsymbol{y}_P] = [y_1, \dots, y_N] &&\in \mathbb{R}^{K \times N}, \\
\boldsymbol{m} &= [\boldsymbol{m}_1, \dots, \boldsymbol{m}_P] = [m_1, \dots, m_N] &&\in \{0,1\}^{K \times N}, \\
\boldsymbol{x} &= [\boldsymbol{x}_1, \dots, \boldsymbol{x}_P] = [x_1, \dots, x_N] &&\quad (\text{size } Q \times N), \\
\boldsymbol{t} &= [\boldsymbol{t}_1^T, \dots, \boldsymbol{t}_P^T]^T = [t_1, \dots, t_N]^T &&\in \mathbb{R}^N.
\end{aligned}
$$

We rely on a latent space $\mathcal{Z} = \mathbb{R}^L$ and combine the latent embedding for all N samples as $\boldsymbol{z} = [z_1, \dots, z_N] \in \mathbb{R}^{L \times N}$.

Variational autoencoders. Assuming a deep latent variable model $p_\omega(y, z) = p_\psi(y|z)p_\theta(z)$, inference of the posterior $p_\omega(z|y) = p_\psi(y|z)p_\theta(z)/p_\omega(y)$ is in general intractable, as the evidence $p_\omega(y)$ cannot be computed analytically due to the highly non-linear relationship between z and y. Common practice is to rely on amortized inference (Kingma & Welling, 2014; Rezende et al., 2014), where a parameterized approximation, $q_\phi(\boldsymbol{z}|\boldsymbol{y})$, to the true posterior is inferred by optimizing a lower bound to the evidence,

$$
\log p_\omega(\boldsymbol{y}) \geq \mathbb{E}_q \left[\log p_\psi(\boldsymbol{y}|\boldsymbol{z})\right] - \text{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{y})||p_\theta(\boldsymbol{z})),
$$

with respect to all parameters. Usually, likelihood $p_\psi(\boldsymbol{y}|\boldsymbol{z})$, prior $p_\theta(\boldsymbol{z})$, and variational posterior $q_\phi(\boldsymbol{z}|\boldsymbol{y})$ are assumed to be mean-field, i.e., to factorize over their respective random variables.

GP-prior variational autoencoder. Despite the computational efficiency provided by a factorized prior $p_\theta(\boldsymbol{z})$ over the latent variables, its major limitation is the inability to model correlations between data samples. Prior work addressed this by combining VAEs with GPs, creating a powerful probabilistic model for this task (Casale et al., 2018; Fortuin et al., 2020; Ramchandran et al., 2021). The key difference is that the factorized prior $p_\theta(\boldsymbol{z})$ is replaced by a GP-prior $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ which depends on auxiliary information $\boldsymbol{x}$. The conditional generative model is then given as

$$p_\omega(\boldsymbol{y}|\boldsymbol{x}) = \int \prod_{i=1}^{N} p_\psi(y_i|z_i) p_\theta(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z}.$$

Defining a mapping from the covariates to the latent space, $\mathbf{f} : \mathcal{X} \to \mathcal{Z}$, such that $z = \mathbf{f}(x) = [f_1(x), \dots, f_L(x)]^T$, these models assume a GP-prior over each f_l,

$$f_l(x) \sim \mathcal{GP}\left(\mu_l(x), k_l(x, x'|\theta_l)\right),$$

where $\mu_l(x)$ is the mean function and $k_l(x, x'|\theta_l)$ is the covariance function parameterized by θ_l.

Given this prior, the l-th latent variable $\bar{\boldsymbol{z}}_l = f_l(\boldsymbol{x}) = [f_l(x_1), \dots, f_l(x_N)]^T$ follows a multivariate Gaussian distribution across the N data samples

$$p_{\theta_l}(\bar{\boldsymbol{z}}_l|\boldsymbol{x}) = p_{\theta_l}(f_l(\boldsymbol{x})) = \mathcal{N}\Big(\bar{\boldsymbol{z}}_l \Big| \mathbf{0}, K^{(l)}_{\boldsymbol{xx}}\Big),$$

where $K^{(l)}_{\boldsymbol{xx}}$ is a $N \times N$ covariance matrix such that $\{K^{(l)}_{\boldsymbol{xx}}\}_{ij} = k_l(x_i, x_j|\theta_l)$. We follow common practice and factorize the GP-prior across its L dimensions, such that the conditional prior is given as

$$p_\theta(\boldsymbol{z}|\boldsymbol{x}) = \prod_{l=1}^{L} p_{\theta_l}(\bar{\boldsymbol{z}}_l|\boldsymbol{x}) = \prod_{l=1}^{L} \mathcal{N}\left(\bar{\boldsymbol{z}}_l \Big| \mathbf{0}, K^{(l)}_{\boldsymbol{xx}}\right).$$

The main distinction among previous GP-prior models lies in the choice of covariance functions. GPPVAE (Casale et al., 2018) and GP-VAE (Fortuin et al., 2020) both rely on restricted kernels that do not adequately model longitudinal data structure. Instead, we adopt the proposal by Ramchandran et al. (2021) who introduce a flexible additive kernel structure that is specifically designed for longitudinal data and is capable of employing various interactions between continuous and categorical covariates

$$k_l(x, x'|\theta_l) = \sum_{r=1}^{R} k_{l,r}(x, x'|\theta_{l,r}) + \sigma^2_{zl},$$

such that

$$p_\theta(\boldsymbol{z}|\boldsymbol{x}) = \prod_{l=1}^{L} \mathcal{N}\left(\bar{\boldsymbol{z}}_l \Big| \mathbf{0}, \sum_{r=1}^{R} K^{(l,r)}_{\boldsymbol{xx}} + \sigma^2_{zl} I_N\right).$$

3.2. Modeling structured missingness

These GP-prior models are capable of dealing with missing values solely by substituting zeros or alternative values and propagating y through encoder-decoder structure to perform imputation. However, this approach lacks the ability to model specific missingness patterns therefore making them suitable only for an MCAR scenario, an unrealistic assumption in many real applications. In this work, we solve this constraint and propose the *longitudinal latent variable model with structured missingness (LLSM)*.

To model non-random missingness in VAE models various approaches exist. For example, Mattei & Frellsen (2019) and Ipsen et al. (2021) model the dependency between y and m directly, whereas Collier et al. (2020) propose a VAE model that incorporates an additional latent variable to account for structured missingness. In this work, we follow Collier et al. (2020) and introduce a second latent variable $z^m \in \mathbb{R}^{L_m}$ associated with a missingness mask m, while referring to the latent variables associated to y from now on as $z^y \in \mathbb{R}^{L_y}$. To properly model MNAR we assume that m depends on both z^m and z^y. The joint likelihood for a single sample is given as

$$\begin{aligned} p_\omega(y^{\text{o}}, z^y, m, z^m|x) = p_{\psi_y}(y^{\text{o}}|z^y, z^m, m) p_{\psi_m}(m|z^y, z^m) \\ \cdot p_{\theta_y}(z^y|x) p_{\theta_m}(z^m|x), \end{aligned} \tag{1}$$

where y^{o} refers to the observed features specified by m. Also, by optionally conditioning y on z^m, we can use any information contained in the missing mask, e.g., during an imputation task. To model correlation within the missingness patterns, e.g., across time, or within a patient, we assign z^m a GP prior as well,

$$z^m(x) \sim \mathcal{GP}\left(0, k(x, x'|\theta_m)\right).$$

Additionally, we assume that y^o and m are distributed as

$$\begin{aligned} p_{\psi_y}(y^{\text{o}}|z^y, z^m, m) &= \mathcal{N}(y|g_{\psi_y}(z^y, z^m), \boldsymbol{\sigma}^2) \odot m \\ p_{\psi_m}(m|z^y, z^m) &= \mathcal{B}er(m|g_{\psi_m}(z^y, z^m)), \end{aligned}$$

where the decoder functions g_{ψ_y} and g_{ψ_m} are parameterized by neural networks, $\odot$ denotes an element-wise Hadamard product, and the observational variance parameters $\boldsymbol{\sigma}^2 = \text{diag}(\sigma^2_{y_1}, \dots, \sigma^2_{y_K})$ are optimized jointly with all other parameters via gradient descent. The graphical model of LLSM is shown in Figure 2.

We approximate the intractable posterior across all N samples $p(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{x}, \boldsymbol{y}^{\text{o}}, \boldsymbol{m})$ using a mean-field amortized inference distribution

$$\begin{aligned} q_\phi(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m}) &= q_{\phi_y}(\boldsymbol{z}^y|\boldsymbol{y}^{\text{o}}) q_{\phi_m}(\boldsymbol{z}^m|\boldsymbol{m}) \\ &= \prod_{i=1}^{N} q_{\phi_y}(z^y_i|y^{\text{o}}_i) q_{\phi_m}(z^m_i|m_i) \end{aligned} \tag{2}$$

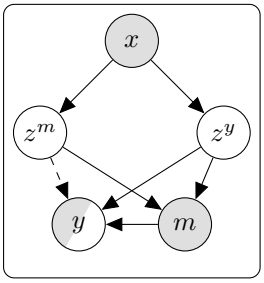

Figure 2. The graphical model of LLSM. Shaded, partially-shaded, and blank circles refer to observed, partially observed, and latent variables. The dashed arrow highlights an optional dependency.

where $q_{\phi_y}(z_i^y|y_i^o)$ and $q_{\phi_m}(z_i^m|m_i)$ are diagonal Gaussian distributions parameterized by neural network-based mappings from the corresponding inputs y_i^o and m_i. The lower bound on the evidence to be optimized is given as

$$\begin{aligned}\log p(\boldsymbol{y}^o, \boldsymbol{m}|\boldsymbol{x}) \geq\ & \mathbb{E}_{q_\phi}\left[\log p_{\psi_y}(\boldsymbol{y}^o|\boldsymbol{z}^y, \boldsymbol{z}^m, \boldsymbol{m})\right] \\ & + \mathbb{E}_{q_\phi}\left[\log p_{\psi_m}(\boldsymbol{m}|\boldsymbol{z}^y, \boldsymbol{z}^m)\right] \\ & - \hat{\mathrm{KL}}(q_{\phi_y}(\boldsymbol{z}^y|\boldsymbol{y}^o)||p_{\theta_y}(\boldsymbol{z}^y|\boldsymbol{x})) \\ & - \hat{\mathrm{KL}}(q_{\phi_m}(\boldsymbol{z}^m|\boldsymbol{m})||p_{\theta_m}(\boldsymbol{z}^m|\boldsymbol{x})),\end{aligned}$$

where, in order to maintain computational tractability, and to be able to perform mini-batching, we substitute the KL terms with the corresponding upper bounds $\hat{\mathrm{KL}}$ derived by Ramchandran et al. (2021) that, for longitudinal data, are tighter than the well-known bound proposed by Titsias (2009). Further details on the lower bound and KL upper bound are given in Appendix A and Appendix B.

3.3. Modeling Time

Prior work relying on GP-based priors suffers from a second restriction. They often rely on time t as a primary, or even the only (Fortuin et al., 2020), covariate that is used in the covariance function. This implies that the similarity between two measurements y and y' is directly contingent on the corresponding t and t', e.g., their temporal difference when employing a stationary kernel. While this assumption is reasonable in some use cases, it is too limiting to be universally applicable. For instance, consider a scenario where a patient develops a disease, and since the progression varies for each individual, it may be more appropriate for similarity to be determined not solely by the elapsed time since the onset of the disease, but rather by their current well-being, which can be captured by factors such as the frequency of doctor visits. This example highlights a possible bias that may occur if the dependence between timepoints and longitudinal observations is ignored. To properly account for such variations, we model t by a temporal point process and add the intensity of this point process as an additional input to the GP kernel computation.

Temporal point processes (TPP). A TPP (Cox & Isham, 1980) is a stochastic process over variable-length sequences in some time interval $\mathcal{T} = [0, T]$ defined via an intensity function $\lambda(t)$. The probability density function of N observed points $\boldsymbol{t} = \{t_i \in \mathcal{T}\}$ is defined as

$$p(\boldsymbol{t}|\lambda) = \exp\Big(-\int_{\mathcal{T}} \lambda(t)dt\Big) \prod_{i=1}^{N} \lambda(t_i).$$

TPPs can be divided into roughly two classes: inhomogeneous Poisson processes, where the intensity function only depends on the current time point t, and self-exciting processes, where the occurrence of events changes the intensity. One type of these self-exciting processes, known as the *Hawkes process* (Hawkes, 1971), has an intensity function

$$\lambda(t) = \mu + \sum_{t_j < t} \nu(t - t_j),$$

where ν is a *triggering kernel* that characterizes the influence of past events on intensity at time t and μ is a corresponding baseline. Inspired by the broad applicability of Hawkes processes (Hawkes, 2018), we adopt the proposal by Liu & Hauskrecht (2019) to model such self-exciting processes with GPs by computing kernels from the last D timestamps.

GP point processes. Given a latent variable z^λ with

$$\begin{aligned} z^\lambda(t) &\sim \mathcal{GP}\big(0, k_{\theta_\lambda}(v_D, {v_D}')\big) \\ v_D &= t - t_D, \end{aligned} \tag{3}$$

where t_D denotes D previous timestamps before t, v_D are the elapsed times between t and t_D, and k_{θ_λ} is an additive kernel structure, we model the intensity as

$$\lambda(t) = (z^\lambda(t) + \beta)^2 \tag{4}$$

where β is either a trainable baseline or a function that can depend on static covariates (John & Hensman, 2018). We choose a squared link function as it provides an analytical tractability (Lloyd et al., 2015).

The posterior distribution of the intensity,

$$p(\lambda|\boldsymbol{t}) = \frac{p(\lambda)\exp\big(-\int_{\mathcal{T}}\lambda(t)dt\big)\prod_{i=1}^{N}\lambda(t_i)}{\int p(\lambda)\exp\big(-\int_{\mathcal{T}}\lambda(t)dt\big)\prod_{i=1}^{N}\lambda(t_i)d\lambda},$$

is intractable due to the integration over λ. To overcome this challenge, we approximate it with a variational distribution $p(z^\lambda|\boldsymbol{u})q(\boldsymbol{u})$ that relies on inducing points $\boldsymbol{u}$ for additional scalability (Quiñonero Candela & Rasmussen, 2005). See Appendix C for a more detailed discussion.

LLPPSM. Combining such a point process with our LLSM model allows us to capture intricate missingness patterns and to effectively leverage information embedded

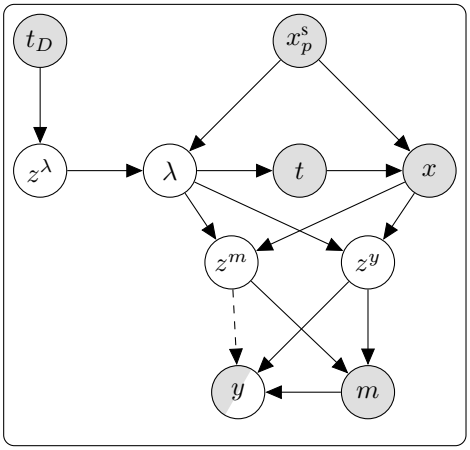

Figure 3. The graphical model of LLPPSM. (Partially) shaded, and blank circles refer to (partially) observed, and latent variables. The dashed arrow highlights an optional dependency, x_p^s are static covariates and t_D the D previous time steps to t.

in the time points. We call this model *longitudinal latent variable model for high-dimensional point process with structured missingness (LLPPSM)*. See Figure 3 for its plate diagram.

Defining z^λ and λ as in Equations (3) and (4), we extend the GP priors for z^y and z^m by letting their covariance kernel depend on the rate of the TPP λ as well, i.e.,

$$z^y(x,\lambda) \sim \mathcal{GP}\big(0, k((x,\lambda(t)),(x',\lambda(t'))|\theta_y)\big),$$

and analogously for z^m. As inference of the full model remains intractable, we once again rely on variational inference and use the following variational approximation

$$\begin{aligned} &q(\boldsymbol{z}^y, \boldsymbol{z}^m, z^\lambda, \boldsymbol{u}|\boldsymbol{y}^o, \boldsymbol{m}) \\ &\quad = q_{\phi_y}(\boldsymbol{z}^y|\boldsymbol{y}^o) q_{\phi_m}(\boldsymbol{z}^m|\boldsymbol{m}) p(z^\lambda|\boldsymbol{u}) q(\boldsymbol{u}), \end{aligned}$$

where $q_{\phi_y}(\boldsymbol{z}^y|\boldsymbol{y}^o)$ and $q_{\phi_m}(\boldsymbol{z}^m|\boldsymbol{m})$ are defined as in Equation (2), and $\boldsymbol{u}$ are the inducing points of z^λ. The bound to be optimized is given as

$$\begin{aligned} &\log p(\boldsymbol{y}^o, \boldsymbol{m}, \boldsymbol{t}|\boldsymbol{x}^s) \geq \\ &\mathbb{E}_q \left[\log \left(p_{\psi_y}(\boldsymbol{y}^o|\boldsymbol{z}^y, \boldsymbol{z}^m, \boldsymbol{m}) p_{\psi_m}(\boldsymbol{m}|\boldsymbol{z}^y, \boldsymbol{z}^m) p(\boldsymbol{t}|\lambda)\right)\right] \\ &\quad - \mathrm{KL}(q(\boldsymbol{u})||p(\boldsymbol{u})) \\ &\quad - \mathbb{E}_{p(z^\lambda|\boldsymbol{u})\cdot q(\boldsymbol{u})}[\hat{\mathrm{KL}}(q_{\phi_y}(\boldsymbol{z}^y|\boldsymbol{y}^o)||p_{\theta_y}(\boldsymbol{z}^y|\boldsymbol{x}, \lambda(\boldsymbol{t}))] \\ &\quad - \mathbb{E}_{p(z^\lambda|\boldsymbol{u})\cdot q(\boldsymbol{u})}[\hat{\mathrm{KL}}(q_{\phi_m}(\boldsymbol{z}^m|\boldsymbol{m})||p_{\theta_m}(\boldsymbol{z}^m|\boldsymbol{x}, \lambda(\boldsymbol{t}))], \end{aligned}$$

where $\boldsymbol{x}^s$ refers to static covariates and $\boldsymbol{x}$ is composed of $\boldsymbol{x}^s$ and $\boldsymbol{t}$. See Appendix C for a detailed derivation and discussion.

3.4. Imputation and future prediction

Our method can be employed for various tasks such as imputation and future prediction. The imputation is done by substituting missing elements with some intermediate values (in our case zeros), and propagating them through the encoder-decoder structure of our model so that the decoder imputes the missing values.

Future predictions are obtained by evaluating the posterior predictive distribution $p(y_* \mid x_*, \boldsymbol{y}^o, \boldsymbol{x}, \boldsymbol{m})$ for new data y_* given covariates x_* and all training data. A detailed explanation together with the necessary derivations to approximate this intractable density is given in Appendix D.

3.5. Computational complexity and scalability

The complexity of LLSM is dominated by computation of KL divergence upper bounds $\hat{\mathrm{KL}}(q_{\phi_y}(\boldsymbol{z}^y|\boldsymbol{y}^o)||p_{\theta_y}(\boldsymbol{z}^y|\boldsymbol{x}))$ and $\hat{\mathrm{KL}}(q_{\phi_m}(\boldsymbol{z}^m|\boldsymbol{m})||p_{\theta_m}(\boldsymbol{z}^m|\boldsymbol{x}))$, which, by employing the techniques from Ramchandran et al. (2021), have complexity $\mathcal{O}(\sum_{p=1}^P {n_p}^3 + NM^2)$, where M is the number of inducing points. We also adopt the mini-batching scheme from Ramchandran et al. (2021) that provides additional scalability to large-sized datasets in terms of memory consumption.

For LLPPSM, an additional complexity comes from the point process computation, which corresponds to $\mathcal{O}(NM^2D^2)$ (Liu & Hauskrecht, 2019), therefore the total complexity is $\mathcal{O}(\sum_{p=1}^P {n_p}^3 + NM^2D^2)$, where NM^2 vanishes as NM^2D^2 dominates it. When training LLPPSM, we also employ mini-batching in a similar fashion as for LLSM to achieve additional scalability.

4. Experiments

We demonstrate the efficiency of our proposal on various tasks, such as missing value imputation, long-term prediction, for synthetic as well as real-world healthcare datasets. We compare against a variety of models: GPPVAE (Casale et al., 2018) serves as a general GP-prior representative, L-VAE (Ramchandran et al., 2021) as a variant specifically designed for longitudinal type of data, GRUI-GAN (Luo et al., 2018) is a GAN-based model capable of modelling non-random missingness, and mean imputation/prediction is a common simple baseline. As GRUI-GAN is not designed for generative purposes, we only provide imputation results for this method. For each method we evaluate its mean-squared error (MSE) and report the mean performance as well as standard deviation computed over five runs. The lowest mean in each experiment is marked bold in the corresponding table. See Appendix E for further experimental details (e.g., hyperparameters, kernel structures) that are not specified in the main text and Appendix F for neural network architectures. An implementation of our proposed methods is available at `https://github.com/sinelnikovmaxim/MPP-VAE`.

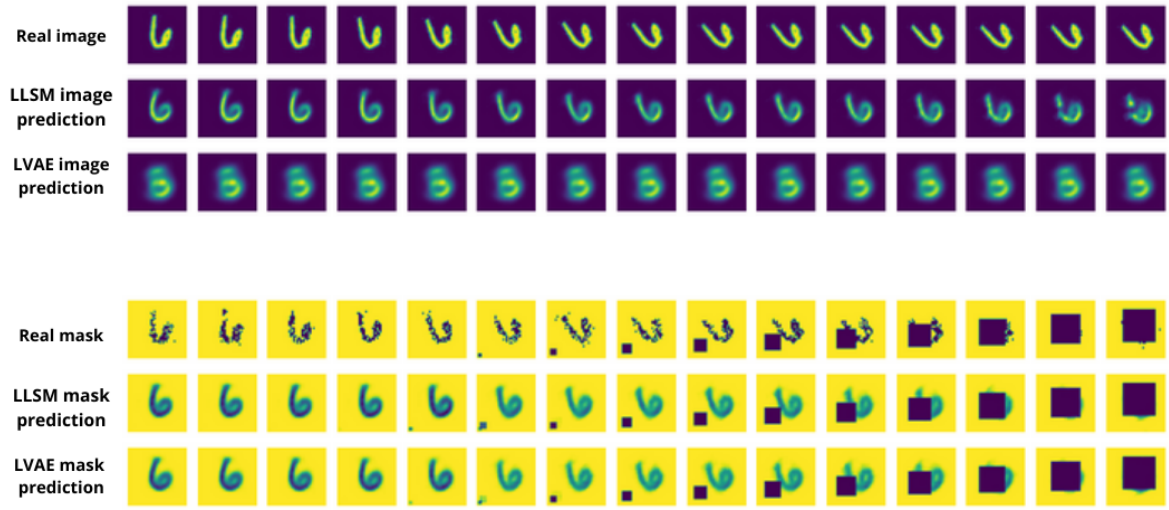

Figure 4. Future predictions of data (top) and missingness mask (bottom) on the regularly sampled health MNIST dataset. Although predictions of mask look almost identical, the prediction of data cannot be captured by L-VAE, whereas LLSM does it very accurately.

Table 2. Imputation MSEs on the regularly sampled health MNIST dataset. The percentage represents maximum probability of pixel being missing.

Method	50%	75%	90%
mean imputation	0.266±0.000	0.314±0.000	0.373±0.000
GPPVAE	0.248±0.004	0.291±0.004	0.379±0.015
GRUI-GAN	0.224±0.037	0.218±0.012	0.269±0.028
L-VAE	**0.124±0.009**	0.283±0.062	0.373±0.001
LLSM (ours)	**0.124±0.008**	**0.144±0.009**	**0.174±0.016**

Table 3. Future data prediction MSEs on the regularly sampled health MNIST dataset. The percentage represents maximum probability of pixel being missing.

Method	50%	75%	90%
mean prediction	0.040±0.000	0.042±0.000	0.048±0.000
GPPVAE	0.041±0.000	0.041±0.000	0.048±0.001
L-VAE	**0.021±0.002**	0.038±0.008	0.047±0.000
LLSM (ours)	0.023±0.002	**0.024±0.001**	**0.026±0.002**

4.1. Regularly sampled Health MNIST

To simulate a high-dimensional longitudinal dataset with structured missingness, we used a modified version of the MNIST dataset (LeCun et al., 1998) called Health MNIST (Krishnan et al., 2015). We chose two digits, '3' and '6', to represent two biological genders. We simulated a shared time-related effect by shifting all digit instances towards the right corner over time. In our experiments, half of the instances remain healthy and half get a disease. To demonstrate changes in the laboratory measurements of the diseased individuals, we rotated digits with the amount of rotation depending on the time from disease diagnosis. Each sample has in total five covariates: *time*, *id*, *diseasePresence*, *diseaseTime*, and *gender*, where *id* serves as the identifier of a specific instance. The timestamps of all observations are regularly sampled which is why we only evaluated LLSM.

To model MNAR, the probability of each pixel being missing depends on the color intensity of that pixel: the higher the intensity, the higher is the probability of the pixel being unobserved. For MAR, we applied a square-shaped missingness mask to the images. If a patient is healthy, no box is applied. For diseased patients, a mask is only applied upon the onset of the disease. Afterwards, the mask starts to increase in size linearly as time progresses. See Figure 4 for an illustration. In this case, missingness depends on both unobserved signal as well as on covariate information.

The training set consists of $P = 900$ unique instances, each having $n_p = 20$ time points. The test set contains 100 unique instances, with 15 last observations for each instance. When performing future prediction, the model conditions on all training data as well as first five observations of each instance, that are kept separately.

Table 2 shows that our method outperforms all baselines on the task of missingness imputation. The same holds for future prediction of data from covariates (Table 3), with the exception for the simplest missingness scenario where L-VAE is slightly better. We also performed future prediction of missingness mask by the same approach. Because none of the baseline methods is able to model the mask m explicitly, we separately modelled missingness by training an L-VAE with a Bernoulli likelihood for m. The results are shown in Table 4 and show that the mask prediction is almost identical except for the case with the highest missingness when LLSM is slightly better. In Figure 4, we demonstrate

Table 4. Future missingness prediction MSEs on the regularly sampled health MNIST dataset. The percentage represents maximum probability of pixel being missing.

Method	50%	75%	90%
L-VAE	$\mathbf{0.032}_{\pm 0.000}$	$\mathbf{0.038}_{\pm 0.002}$	$0.040_{\pm 0.002}$
LLSM (ours)	$\mathbf{0.031}_{\pm 0.002}$	$\mathbf{0.038}_{\pm 0.003}$	$\mathbf{0.038}_{\pm 0.002}$

Table 5. Imputation MSEs on the irregularly sampled health MNIST dataset. The percentage represents maximum probability of pixel being missing.

Method	50%	75%	90%
mean imputation	$0.259_{\pm 0.000}$	$0.335_{\pm 0.000}$	$0.380_{\pm 0.000}$
GPPVAE	$0.239_{\pm 0.002}$	$0.319_{\pm 0.001}$	$0.396_{\pm 0.004}$
GRUI-GAN	$0.165_{\pm 0.016}$	$0.203_{\pm 0.021}$	$0.277_{\pm 0.035}$
L-VAE	$0.171_{\pm 0.045}$	$0.264_{\pm 0.059}$	$0.379_{\pm 0.001}$
LLSM (ours)	$0.130_{\pm 0.007}$	$0.163_{\pm 0.011}$	$\mathbf{0.191}_{\pm 0.012}$
LLPPSM (ours)	$\mathbf{0.128}_{\pm 0.005}$	$\mathbf{0.162}_{\pm 0.007}$	$0.207_{\pm 0.028}$

visually the benefits of our model for future prediction for the case of 75% maximum probability of missingness.

4.2. Irregularly sampled Health MNIST

We modified the previous setup such that timepoints come from a random process and similarity in the observations depends on covariates as well as the underlying rate of the TPP that was discussed in Section 3.3. We implemented it in a following way: for healthy patients, the timestamps come from a homogeneous Poisson process with intensity $\lambda = 0.1$ and the digit is not modified, whereas for diseased patients, timestamps are generated according to a Hawkes process, with baseline intensity $\mu = 0.5$ and the digit rotation depends on the intensity of process at the moment: the higher the intensity, the stronger the rotation. For healthy patients we modelled MNAR as in the previous setup, while for diseased patients we also applied a square-shaped mask with its size being proportional to the intensity of the process at that moment. Table 5 shows that both of our models improve upon the baselines in all imputation scenarios. Moreover, Tables 6 and 7 show that LLPPSM outperforms LLSM in both future data and mask prediction tasks. The Figure 5 depicts that although LLSM is capable of capturing the general form of an image, it cannot model the rotation properly due to the limited kernel component related to time whereas LLPPSM does it well. The same holds for prediction of mask. The inferred mean intensity function of the point process for one individual can be found in Appendix G.

4.3. Physionet data

We evaluated our model on healthcare data from the 2012 Physionet Challenge (Silva et al., 2012). The dataset contains around 12,000 patients monitored on the intensive care unit (ICU) for 48 hours. We modelled measurements of 37 different attributes, such as glucose level, heart rate, body temperature, etc. The dataset is extremely sparse, with about 85% missing values. We cannot directly measure imputation performance due to the lack of ground truth data for missing values, hence, to test the learned representations of our models, we used this dataset only for future prediction. We predicted values of laboratory attributes given the knowledge of the first ten measurements for a patient in the test set. As auxiliary covariates, we employed the following variables: *time of the measurement*, *id*, *type of ICU*, *gender*, and *in-hospital mortality*. More information regarding Physionet data can be found in Appendix E.2. We present the results in Table 8. LLSM performs best, with LLPPSM performing worse. This can be explained by the fact that many observations are taken regularly each hour, making the temporal process be pseudo-stochastic, which is reflected in the intensity of the point process that starts to explode at each hour timepoint, hence modelling it just brings additional redundant information to the model.

Table 6. Future data prediction MSEs on the irregularly sampled health MNIST dataset. The percentage represents maximum probability of pixel being missing.

Method	50%	75%	90%
mean prediction	$0.039_{\pm 0.000}$	$0.044_{\pm 0.000}$	$0.047_{\pm 0.000}$
GPPVAE	$0.040_{\pm 0.000}$	$0.044_{\pm 0.001}$	$0.049_{\pm 0.000}$
L-VAE	$0.029_{\pm 0.005}$	$0.036_{\pm 0.006}$	$0.047_{\pm 0.000}$
LLSM (ours)	$0.030_{\pm 0.001}$	$0.031_{\pm 0.001}$	$0.031_{\pm 0.001}$
LLPPSM (ours)	$\mathbf{0.025}_{\pm 0.002}$	$\mathbf{0.027}_{\pm 0.001}$	$\mathbf{0.030}_{\pm 0.003}$

Table 7. Future missingness prediction MSEs on the irregularly sampled health MNIST dataset. The percentage represents maximum probability of pixel being missing.

Method	50%	75%	90%
L-VAE	$0.063_{\pm 0.000}$	$0.067_{\pm 0.001}$	$0.068_{\pm 0.001}$
LLSM (ours)	$0.063_{\pm 0.002}$	$0.067_{\pm 0.001}$	$0.066_{\pm 0.001}$
LLPPSM (ours)	$\mathbf{0.054}_{\pm 0.006}$	$\mathbf{0.056}_{\pm 0.005}$	$\mathbf{0.058}_{\pm 0.005}$

Table 8. Future prediction on Physionet dataset.

Method	MSE
mean prediction	$0.785_{\pm 0.000}$
GPPVAE	$0.786_{\pm 0.001}$
L-VAE	$0.720_{\pm 0.008}$
LLSM (ours)	$\mathbf{0.713}_{\pm 0.006}$
LLPPSM (ours)	$0.745_{\pm 0.009}$

5. Conclusions

In this work, we introduced a novel probabilistic framework for multivariate data with missing values. First, we

Figure 5. Future predictions of data (top) and missingness mask (bottom) on irregularly sampled health MNIST dataset.

developed a deep latent variable model, LLSM, that models structured missingness via separate set of latent variables. Second, we extended this model by utilizing temporal point process to account for stochastic nature of timepoints, LLPPSM. Our methods are specifically designed for longitudinal type of data by leveraging GPs to define priors for latent variables. We demonstrated excellent performance of both models on different representation learning tasks and expect them to become useful tools in the analysis of high-dimensional temporal and longitudinal data.

Acknowledgements

We acknowledge the computational resources provided by the Aalto Science-IT project. This work was supported by the Research Council of Finland (decision number: 359135).

Impact statement

This paper presents a work whose goal is to introduce a new method for analyzing high-dimensional data with missing values in the longitudinal scenario. We do not see any potential harmful societal consequences of our work.

References

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

Che, Z., Purushotham, S., Cho, K., Sontag, D., and Liu, Y. Recurrent neural networks for multivariate time series with missing value. Scientific reports, 8(1):6085, 2018.

Collier, M., Nazabal, A., and Williams, C. K. I. VAEs in the presence of missing data. In *the First ICML Workshop on The Art of Learning with Missing Values*, 2020.

Cox, D. R. and Isham, V. *Point processes*, volume 12. CRC Press, 1980.

Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., and Song, L. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

Fortuin, V., Baranchuk, D., Raetsch, G., and Mandt, S. GP-VAE: Deep probabilistic time series imputation. In *Proceedings of the 32rd International Conference on Artificial Intelligence and Statistics*, 2020.

Ghalebikesabi, S., Cornish, R., Holmes, C., and Kelly, L. Deep generative missingness pattern-set mixture models. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, 2021.

Hawkes, A. G. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

Hawkes, A. G. Hawkes processes and their applications to finance: A review. *Quant. Financ.*, 18(2):193–198, 2018.

Hensman, J., Fusi, N., and Lawrence, N. D. Gaussian processes for big data. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence*, 2013.

Ipsen, N. B., Mattei, P.-A., and Frellsen, J. not-MIWAE: deep generative modelling with missing not at random data. In *9th International Conference on Learning Representations*, 2021.

John, S. and Hensman, J. Large-scale cox process inference using variational fourier features. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

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

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

Krishnan, R. G., Shalit, U., and Sontag, D. Deep Kalman filters. In *arXiv preprint arXiv:1511.05121*, 2015.

LeCun, Y., Cortes, C., and Burges, C. The mnist database of handwritten digits. 1998.

Liu, S. and Hauskrecht, M. Nonparametric regressive point processes based on conditional Gaussian processes. In *Advances in Neural Information Processing Systems*, 2019.

Liu, X. *Methods and applications of longitudinal data analysis*. Elsevier, New York, 2015.

Lloyd, C., Gunter, T., Osborne, M., and Roberts, S. Variational inference for Gaussian process modulated Poisson processes. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

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

Mattei, P.-A. and Frellsen, J. MIWAE: Deep generative modelling and imputation of incomplete data sets. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Mitra, R., McGough, S. F., Chakraborti, T., Holmes, C., Copping, R., Hagenbuch, N., Biedermann, S., Noonan, J., Lehmann, B., Shenvi, A., Doan, X. V., Leslie, D., Bianconi, G., Sanchez-Garcia, R., Davies, A., Mackintosh, M., Andrinopoulou, E.-R., Basiri, A., Harbron, C., and MacArthur, B. D. Learning from data with structured missingness. *Nat. Mach. Intell.*, 5:13–23, 2023.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.

Pullenayegum, E. M. and Lim, L. S. Longitudinal data subject to irregular observation: A review of methods with a focus on visit processes, assumptions, and study design. *Statistical Methods in Medical Research*, 25(6): 2992–3014, 2016.

Puri, M. L. and Tuan, P. D. Maximum likelihood estimation for stationary point processes. In *Proceedings of the National Academy of Sciences*, 1986.

Quiñonero Candela, J. and Rasmussen, C. E. A unifying view of sparse approximate gaussian process regression. In *Journal of Machine Learning Research*, 2005.

Ramchandran, S., Tikhonov, G., Kujanpää, K., Koskinen, M., and Lähdesmäki, H. Longitudinal variational autoencoder. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, 2021.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Rubin, D. B. Inference and missing data. *Biometrika*, 63: 581–592, 1976.

Sang, P., Kong, D., and Yang, S. Functional principal component analysis for longitudinal observations with sampling at random, 2022.

Silva, I., Moody, G., Scott, D. J., Celi, L. A., and Mark, R. G. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. In *2012 Computing in Cardiology.IEEE*, 2012.

Titsias, M. Variational learning of inducing variables in sparse gaussian processes. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, 2009.

Xu, G., Zhang, J., Li, Y., and Guan, Y. Bias-correction and test for mark-point dependence with replicated marked point processes, 2022.

APPENDIX

A. Deriving the ELBO for LLSM

By introducing a variational distribution $q_\phi(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m})$, the marginal likelihood can be expanded as

$$\begin{aligned}
\log p_\omega(\boldsymbol{y}^{\text{o}}, \boldsymbol{m}|\boldsymbol{x}) &= \iint q_\phi(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m}) \log \frac{p_\omega(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m}) p_\omega(\boldsymbol{y}^{\text{o}}, \boldsymbol{m}|\boldsymbol{x})}{p_\omega(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m})} d\boldsymbol{z}^y d\boldsymbol{z}^m \\
&= \iint q_\phi(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m}) \log \frac{p_\omega(\boldsymbol{y}^{\text{o}}, \boldsymbol{m}, \boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{x})}{p_\omega(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m})} d\boldsymbol{z}^y d\boldsymbol{z}^m \\
&= \iint q_\phi(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m}) \log \frac{p_\psi(\boldsymbol{y}^{\text{o}}, \boldsymbol{m}|\boldsymbol{z}^y, \boldsymbol{z}^m) p_\theta(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{x})}{p_\omega(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m})} d\boldsymbol{z}^y d\boldsymbol{z}^m \\
&= \iint q_\phi(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m}) \log p_\psi(\boldsymbol{y}^{\text{o}}, \boldsymbol{m}|\boldsymbol{z}^y, \boldsymbol{z}^m) d\boldsymbol{z}^y d\boldsymbol{z}^m \\
&\quad + \iint q_\phi(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m}) \log \frac{p_\theta(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{x})}{p_\omega(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m})} d\boldsymbol{z}^y d\boldsymbol{z}^m \\
&= \mathbb{E}_{q_\phi}\left[\log p_\psi(\boldsymbol{y}^{\text{o}}, \boldsymbol{m}|\boldsymbol{z}^y, \boldsymbol{z}^m)\right] \\
&\quad + \iint q_\phi(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m}) \log \frac{p_\theta(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{x}) q_\phi(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m})}{p_\omega(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m}) q_\phi(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m})} d\boldsymbol{z}^y d\boldsymbol{z}^m \\
&= \mathbb{E}_{q_\phi}\left[\log p_\psi(\boldsymbol{y}^{\text{o}}, \boldsymbol{m}|\boldsymbol{z}^y, \boldsymbol{z}^m)\right] \\
&\quad + \iint q_\phi(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m}) \log \frac{q_\phi(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m})}{p_\omega(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m})} d\boldsymbol{z}^y d\boldsymbol{z}^m \\
&\quad - \iint q_\phi(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m}) \log \frac{q_\phi(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m})}{p_\theta(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{x})} d\boldsymbol{z}^y d\boldsymbol{z}^m \\
&= \mathbb{E}_{q_\phi}\left[\log p_\psi(\boldsymbol{y}^{\text{o}}, \boldsymbol{m}|\boldsymbol{z}^y, \boldsymbol{z}^m)\right] \\
&\quad + \text{KL}(q_\phi(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m})||p_\omega(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m})) \\
&\quad - \text{KL}(q_\phi(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m})||p_\theta(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{x})).
\end{aligned}$$

Hence,

$$\begin{aligned}
&\log p_\omega(\boldsymbol{y}^{\text{o}}, \boldsymbol{m}|\boldsymbol{x}) - \text{KL}(q_\phi(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m})||p_\omega(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m})) = \\
&\quad \mathbb{E}_{q_\phi}\left[\log p_\psi(\boldsymbol{y}^{\text{o}}, \boldsymbol{m}|\boldsymbol{z}^y, \boldsymbol{z}^m)\right] - \text{KL}(q_\phi(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m})||p_\theta(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{x})).
\end{aligned}$$

As the KL divergence term is positive, we get

$$\begin{aligned}
\log p_\omega(\boldsymbol{y}^{\text{o}}, \boldsymbol{m}|\boldsymbol{x}) &\geq \mathbb{E}_{q_\phi}\left[\log p_\psi(\boldsymbol{y}^{\text{o}}, \boldsymbol{m}|\boldsymbol{z}^y, \boldsymbol{z}^m)\right] - \text{KL}(q_\phi(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m})||p_\theta(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{x})) \\
&= \mathcal{L}(\phi, \psi, \theta; \boldsymbol{y}^{\text{o}}, \boldsymbol{m}).
\end{aligned}$$

By assuming the following factorizations:

$$\begin{aligned}
p_\psi(\boldsymbol{y}^{\text{o}}, \boldsymbol{m}|\boldsymbol{z}^y, \boldsymbol{z}^m) &= p_{\psi_y}(\boldsymbol{y}^{\text{o}}|\boldsymbol{z}^y, \boldsymbol{z}^m, \boldsymbol{m}) p_{\psi_m}(\boldsymbol{m}|\boldsymbol{z}^y, \boldsymbol{z}^m) \\
q_\phi(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m}) &= q_{\phi_y}(\boldsymbol{z}^y|\boldsymbol{y}^{\text{o}}) q_{\phi_m}(\boldsymbol{z}^m|\boldsymbol{m}) \\
p_\theta(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{x}) &= p_{\theta_y}(\boldsymbol{z}^y|\boldsymbol{x}) p_{\theta_m}(\boldsymbol{z}^m|\boldsymbol{x}),
\end{aligned}$$

the ELBO simplifies to

$$\begin{aligned}
\mathcal{L}(\phi, \psi, \theta; \boldsymbol{y}^{\text{o}}, \boldsymbol{m}) = \mathbb{E}_{q_\phi}\left[\log p_{\psi_y}(\boldsymbol{y}^{\text{o}}|\boldsymbol{z}^y, \boldsymbol{z}^m, \boldsymbol{m})\right] + \mathbb{E}_{q_\phi}\left[\log p_{\psi_m}(\boldsymbol{m}|\boldsymbol{z}^y, \boldsymbol{z}^m)\right] \\
- \text{KL}(q_{\phi_y}(\boldsymbol{z}^y|\boldsymbol{y}^{\text{o}})||p_{\theta_y}(\boldsymbol{z}^y|\boldsymbol{x})) - \text{KL}(q_{\phi_m}(\boldsymbol{z}^m|\boldsymbol{m})||p_{\theta_m}(\boldsymbol{z}^m|\boldsymbol{x})),
\end{aligned}$$

where ψ_y, ψ_m, ϕ_y, ϕ_m are parameterized by neural networks and q_y, q_m are corresponding variational distributions of $\boldsymbol{z}^y$ and $\boldsymbol{z}^m$.

By approximating the KL terms with the corresponding upper bounds $\hat{\text{KL}}$ that are necessary for computational scalability (see Appendix B), the final ELBO is given as

$$\begin{aligned}\mathcal{L}(\phi,\psi,\theta;\boldsymbol{y}^{\text{o}},\boldsymbol{m}) \geq \mathbb{E}_{q_\phi}\left[\log p_{\psi_y}(\boldsymbol{y}^{\text{o}}|\boldsymbol{z}^y,\boldsymbol{z}^m,\boldsymbol{m})\right] + \mathbb{E}_{q_\phi}\left[\log p_{\psi_m}(\boldsymbol{m}|\boldsymbol{z}^y,\boldsymbol{z}^m)\right] \\ - \hat{\text{KL}}(q_{\phi_y}(\boldsymbol{z}^y|\boldsymbol{y}^{\text{o}})||p_{\theta_y}(\boldsymbol{z}^y|\boldsymbol{x})) - \hat{\text{KL}}(q_{\phi_m}(\boldsymbol{z}^m|\boldsymbol{m})||p_{\theta_m}(\boldsymbol{z}^m|\boldsymbol{x})).\end{aligned}$$

B. Scalable KL Divergence Computation

Here we review the KL upper bound from Ramchandran et al. (2021) that implements a scalable KL divergence computation. Optimising the variational objective requires us to evaluate L KL terms $\text{KL}(\mathcal{N}(\bar{\boldsymbol{\mu}}_{\boldsymbol{l}}, W_l)||\mathcal{N}(\boldsymbol{0},\Sigma_l))$, where $\bar{\boldsymbol{\mu}}_{\boldsymbol{l}} = [\mu_{\phi,l}(y_1),\ldots,\mu_{\phi,l}(y_N)]^T, W_l = \text{diag}(\sigma^2_{\phi,l}(y_1),\ldots,\sigma^2_{\phi,l}(y_N))$, and $\Sigma_l = \sum_{r=1}^R K^{(r,l)}_{\boldsymbol{xx}} + \sigma^2_{zl} I_N$. For notational convenience, we drop the index l. The exact computation has $\mathcal{O}(N^3)$ complexity, making it impractical for large datasets. Therefore, instead of computing it, we will use an upper bound to the KL that comes from the fact that any lower bound for the prior GP marginal log-likelihood induces an upper bound to the KL divergence. Titsias (2009) proposed the free-form variational lower bound for a GP marginal log-likelihood $\log\mathcal{N}(\boldsymbol{z}|\boldsymbol{0},\Sigma)$ by assuming a set of M inducing locations $\boldsymbol{s} = [s_1,\ldots,s_M]$ in $\mathcal{X}$, with the corresponding inducing function values $\boldsymbol{u} = [f(s_1),\ldots,f(s_M)]^T = [u_1,\ldots,u_M]^T$, such that

$$\begin{aligned}p(\boldsymbol{u}) &= \mathcal{N}(\boldsymbol{u}|\boldsymbol{0}, K_{\boldsymbol{ss}}) \\ p(\boldsymbol{f}|\boldsymbol{u}) &= \mathcal{N}(\boldsymbol{f}|K_{\boldsymbol{xs}}K^{-1}_{\boldsymbol{ss}}\boldsymbol{u}, \tilde{K}), \\ \tilde{K} &= K_{\boldsymbol{xx}} - K_{\boldsymbol{xs}}K^{-1}_{\boldsymbol{ss}}K_{\boldsymbol{sx}} \\ p(\boldsymbol{z}|\boldsymbol{f}) &= \mathcal{N}(\boldsymbol{z}|\boldsymbol{f}, \sigma^2_z I_N),\end{aligned}$$

and the corresponding lower bound is $\mathcal{L}(\boldsymbol{z};\Sigma) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, K_{\boldsymbol{xs}}K^{-1}_{\boldsymbol{ss}}K_{\boldsymbol{sx}} + \sigma^2_z I_N) - \frac{1}{2\sigma^2_z}\text{tr}(\tilde{K})$, where $\text{tr}(\cdot)$ is the matrix trace. Titsias bound is known to be tight when M is large enough and the covariance function is smooth enough. Longitudinal data, however, always contain categorical covariates corresponding to instance *ids*, making the covariance function necessarily non-continuous.

To still get a tighter bound, we separate the additive component that corresponds to the instance *id* from the other covariates, so that covariance function has the following form $\Sigma = K^{(A)}_{\boldsymbol{xx}} + \hat{\Sigma}$, where $\hat{\Sigma} = \text{diag}(\hat{\Sigma}_1,\ldots,\hat{\Sigma}_P)$, $\hat{\Sigma}_p = K^{(R)}_{\boldsymbol{x}_p\boldsymbol{x}_p} + \sigma^2_z I_{n_p}$ contains all terms with instance-specific *id* and $K^{(A)}_{\boldsymbol{xx}} = \sum_{r=1}^{R-1} K^{(r)}_{\boldsymbol{xx}}$ contains the other components. Ramchandran et al. (2021) proposed the following upper bound for KL

$$\text{KL} \leq \frac{1}{2}\left(\text{tr}\left(\bar{\Sigma}^{-1}W\right) + \bar{\boldsymbol{\mu}}^T\bar{\Sigma}^{-1}\bar{\boldsymbol{\mu}} - N + \log|\bar{\Sigma}| - \log|W| + \sum_{p=1}^P \text{tr}\left(\hat{\Sigma}_p^{-1}\tilde{K}^{(A)}_{\boldsymbol{x}_p\boldsymbol{x}_p}\right)\right),$$

where $\bar{\Sigma} = K^{(A)}_{\boldsymbol{xs}}{K^{(A)}_{\boldsymbol{ss}}}^{-1}K^{(A)}_{\boldsymbol{sx}} + \hat{\Sigma}$ and $\tilde{K}^{(A)}_{\boldsymbol{x}_p\boldsymbol{x}_p} = K^{(A)}_{\boldsymbol{x}_p\boldsymbol{x}_p} - K^{(A)}_{\boldsymbol{x}_p\boldsymbol{s}}{K^{(A)}_{\boldsymbol{ss}}}^{-1}K^{(A)}_{\boldsymbol{s}\boldsymbol{x}_p}$. This bound has a computational complexity $\mathcal{O}(\sum_{p=1}^P {n_p}^3 + NM^2)$ and Ramchandran et al. (2021) proved it to be tighter than the corresponding bound by Titsias for longitudinal datasets. Despite the reduced complexity, a problem still remains. For every gradient descent step, the algorithm has to iterate through the entire dataset, requiring a substantial allocation of memory and computational time. This issue can be solved using a similar technique as the one presented by Hensman et al. (2013), with adaptation to the properties of a longitudinal kernel. We will only present the final bound and refer the reader to Ramchandran et al. (2021) for a detailed derivation. Defining I_{p_i} to be the index of the ith sample for the pth patient and $\bar{\boldsymbol{\mu}}_p = \left[\bar{\boldsymbol{\mu}}_{I_{p_i}},\ldots,\bar{\boldsymbol{\mu}}_{I_{n_p}}\right]^T$ to be a a sub-vector of $\bar{\boldsymbol{\mu}}$ that is related to the pth patient, the unbiased estimate of the KL divergence upper bound, computed

from the batch with instances $\mathrm{P}_{\text{batch}} \subset \{1, \ldots, P\}$, is defined as

$$\begin{aligned}\hat{\mathrm{KL}} = \frac{1}{2}\frac{P}{|\mathrm{P}_{\text{batch}}|}\sum_{p\in\mathrm{P}}\Bigg(&\left(K^{(A)}_{\boldsymbol{x}_p\boldsymbol{s}}K^{(A)^{-1}}_{\boldsymbol{s}\boldsymbol{s}}\boldsymbol{m}_H - \bar{\boldsymbol{\mu}}_p\right)^T\hat{\Sigma}_p^{-1}\left(K^{(A)}_{\boldsymbol{x}_p\boldsymbol{s}}K^{(A)^{-1}}_{\boldsymbol{s}\boldsymbol{s}}\boldsymbol{m}_H - \bar{\boldsymbol{\mu}}_p\right) + \sum_{i=1}^{n_p}(\hat{\Sigma}_p^{-1})_{ii}\sigma^2_\phi(\boldsymbol{y}_{I_{p_i}})\\ &+ \quad \log|\hat{\Sigma}_p| + \mathrm{tr}\left(\hat{\Sigma}_p^{-1}\tilde{K}^{(A)}_{\boldsymbol{x}_p\boldsymbol{x}_p}\right) + \mathrm{tr}\left(\left(K^{(A)^{-1}}_{\boldsymbol{s}\boldsymbol{s}}HK^{(A)^{-1}}_{\boldsymbol{s}\boldsymbol{s}}\right)\left(K^{(A)}_{\boldsymbol{s}\boldsymbol{x}_p}\hat{\Sigma}_p^{-1}K^{(A)}_{\boldsymbol{x}_p\boldsymbol{s}}\right)\right) - \sum_{i=1}^{n_p}\log\sigma^2_\phi(\boldsymbol{y}_{I_{p_i}})\Bigg)\\ &- \frac{N}{2} + \mathrm{KL}\left[\mathcal{N}(\boldsymbol{m}_H, H)||\mathcal{N}(\boldsymbol{0}, K^{(A)}_{\boldsymbol{s}\boldsymbol{s}})\right],\end{aligned}$$

where $\boldsymbol{m}_H$ and H are variational parameters computed via natural gradients.

C. Various LLPPSM specifications

We define three sets of latent variables: z^y, z^m and z^λ. Observations $\boldsymbol{y}$, masks $\boldsymbol{m}$ and timestamps $\boldsymbol{t}$ are modelled as random variables. The complete joint probability is given as:

$$\begin{aligned}p_\omega(\boldsymbol{y}^{\text{o}}, \boldsymbol{z}^y, \boldsymbol{m}, \boldsymbol{z}^m, \boldsymbol{t}, z^\lambda|\boldsymbol{x}^{\text{s}}, \boldsymbol{v}_D) =& p_{\psi_y}(\boldsymbol{y}^{\text{o}}|\boldsymbol{z}^y, \boldsymbol{z}^m, \boldsymbol{m})p_{\psi_m}(\boldsymbol{m}|\boldsymbol{z}^y, \boldsymbol{z}^m)p_{\theta_y}(\boldsymbol{z}^y|\boldsymbol{x}, \lambda(\boldsymbol{t}))p_{\theta_m}(\boldsymbol{z}^m|\boldsymbol{x}, \lambda(\boldsymbol{t}))\\ &p(\boldsymbol{t}|\lambda)p_{\theta_\lambda}(z^\lambda|\boldsymbol{v}_D),\end{aligned}$$

where $\boldsymbol{v}_D$ corresponds to elapsed times between $\boldsymbol{t}$ and D previous timepoints that occurred before $\boldsymbol{t}$, denoted as $\boldsymbol{t}_D$, $\boldsymbol{x}^{\text{s}}$ refers to static covariates and $\boldsymbol{x}$ consists of $\boldsymbol{t}$ and $\boldsymbol{x}^{\text{s}}$. To compute the marginal likelihood, we have to marginalize over the latent variables as

$$\begin{aligned}p_\omega(\boldsymbol{y}^{\text{o}}, \boldsymbol{m}, \boldsymbol{t}|\boldsymbol{x}^{\text{s}}, \boldsymbol{v}_D) &= \iiint p_\omega(\boldsymbol{y}^{\text{o}}, \boldsymbol{z}^y, \boldsymbol{m}, \boldsymbol{z}^m, \boldsymbol{t}, z^\lambda|\boldsymbol{x}^{\text{s}}, \boldsymbol{v}_D)d\boldsymbol{z}^y d\boldsymbol{z}^m dz^\lambda\\ &= \iiint p_{\psi_y}(\boldsymbol{y}^{\text{o}}|\boldsymbol{z}^y, \boldsymbol{z}^m, \boldsymbol{m})p_{\psi_m}(\boldsymbol{m}|\boldsymbol{z}^y, \boldsymbol{z}^m)p_{\theta_y}(\boldsymbol{z}^y|\boldsymbol{x}, \lambda(\boldsymbol{t}))p_{\theta_m}(\boldsymbol{z}^m|\boldsymbol{x}, \lambda(\boldsymbol{t}))\\ &\quad \cdot p(\boldsymbol{t}|\lambda)p_{\theta_\lambda}(z^\lambda|\boldsymbol{v}_D)d\boldsymbol{z}^y d\boldsymbol{z}^m dz^\lambda.\end{aligned}$$

Factorizing the joint likelihood w.r.t. the observation given the latent variables, we get

$$\begin{aligned}p_\omega(\boldsymbol{y}^{\text{o}}, \boldsymbol{m}, \boldsymbol{t}|\boldsymbol{x}^{\text{s}}, \boldsymbol{v}_D) = \iiint \prod_{i=1}^{N} &p_{\psi_y}(y_i^{\text{o}}|z_i^y, z_i^m, m_i)p_{\psi_m}(m_i|z_i^y, z_i^m)p_{\theta_y}(\boldsymbol{z}^y|\boldsymbol{x}, \lambda(\boldsymbol{t}))p_{\theta_m}(\boldsymbol{z}^m|\boldsymbol{x}, \lambda(\boldsymbol{t}))\\ &p(\boldsymbol{t}|\lambda)p_{\theta_\lambda}(z^\lambda|\boldsymbol{v}_D)d\boldsymbol{z}^y d\boldsymbol{z}^m dz^\lambda.\end{aligned}$$

We assume the following Gaussian process priors

$$\begin{aligned}z^\lambda(t) &\sim \mathcal{G}P\left(0, k(v_D, {v_D}'|\theta_\lambda)\right)\\ z^y(x, \lambda) &\sim \mathcal{G}P\left(0, k((x, \lambda(t)), (x', \lambda(t'))|\theta_y)\right)\\ z^m(x, \lambda) &\sim \mathcal{G}P\left(0, k((x, \lambda(t)), (x', \lambda(t'))|\theta_m)\right).\end{aligned}$$

The decoders of $\boldsymbol{y}$ and $\boldsymbol{m}$ are parameterized by neural networks that predict the mean of the generative distributions

$$\begin{aligned}p_{\psi_y}(y_i^{\text{o}}|z_i^y, z_i^m, m_i) &= N(y_i\big|g_{\psi_y}(z_i^y, z_i^m), \text{diag}(\sigma^2_{y_1}, \ldots, \sigma^2_{y_K})) \odot m_i\\ p_{\psi_m}(m_i\big|z_i^y, z_i^m) &= Ber(m_i|g_{\psi_m}(z_i^y, z_i^m)),\end{aligned}$$

where $\sigma^2_{y_1}, \ldots, \sigma^2_{y_K}$ are jointly optimised via gradient descent, and $\odot$ is the elementwise Hadamard product.

The likelihood of the temporal point process for instance p is given as

$$\begin{aligned}p(\boldsymbol{t}_p|\lambda) &= \exp\Big(-\int_{\mathcal{T}}\lambda(t)dt\Big)\prod_{i=1}^{n_p}\lambda(t_i),\\ \text{where} \qquad \lambda(t) &= (z^\lambda(t) + \beta)^2,\end{aligned}$$

and β is a learnable offset.

The covariances of the GPs of z^y and z^m are parameterized by the additive kernels discussed in Section 3. For z^λ, we instead rely on the following additive structure:

$$k(v_D, {v_D}'|\theta_\lambda) = \sum_{d=1}^{D} \mathbb{1}(v_d)\mathbb{1}({v_d}') \cdot \gamma_d \cdot \exp\left(-\frac{(v_d - {v_d}')^2}{2l_d^2}\right),$$
$$\mathbb{1}(v_d) = \begin{cases} 1, \text{if } v_d < \infty \\ 0, \text{otherwise} \end{cases},$$

where v_d is the elapsed time between t and dth timepoint, t_d, that happened before t, and infinite, if there is no available information about the past event. Hence, the above kernel depends on past events and on the current time point.

As the posterior inference is not tractable, we rely on variational inference and choose the following approximation to the posterior

$$q(\boldsymbol{z}^y, \boldsymbol{z}^m, z^\lambda, \boldsymbol{u} \big| \boldsymbol{y}^{\text{o}}, \boldsymbol{m}) = q_{\phi_y}(\boldsymbol{z}^y|\boldsymbol{y}^{\text{o}}) \cdot q_{\phi_m}(\boldsymbol{z}^m|\boldsymbol{m}) \cdot p(z^\lambda|\boldsymbol{u}) \cdot q(\boldsymbol{u}),$$

where we also employed inducing points of z^λ, denoted by $\boldsymbol{u}$ with $q(\boldsymbol{u}) = N(m_\lambda, S)$. Note that these inducing points are different from those of the scalable KL bound in Appendix B. To shorten the notation we will denote this variational posterior as q in the following derivations. The ELBO is given as

$$\begin{aligned}
\mathcal{L} &= \mathbb{E}_q\left[\log \frac{p_\omega(\boldsymbol{y}^{\text{o}}, \boldsymbol{m}, \boldsymbol{t}, \boldsymbol{z}^y, \boldsymbol{z}^m, z^\lambda, \boldsymbol{u}|\boldsymbol{x}^{\text{s}}, \boldsymbol{v}_D)}{q_{\phi_y}(\boldsymbol{z}^y|\boldsymbol{y}^{\text{o}}) \cdot q_{\phi_m}(\boldsymbol{z}^m|\boldsymbol{m}) \cdot p(z^\lambda|\boldsymbol{u}) \cdot q(\boldsymbol{u})}\right] \\
&= \mathbb{E}_q\left[\log\left(p_{\psi_y}(\boldsymbol{y}^{\text{o}}|\boldsymbol{z}^y, \boldsymbol{z}^m, \boldsymbol{m}) p_{\psi_m}(\boldsymbol{m}|\boldsymbol{z}^y, \boldsymbol{z}^m) p(\boldsymbol{t}|\lambda)\right)\right] - \mathbb{E}_q\left[\log(\frac{q(\boldsymbol{u})}{p(\boldsymbol{u})})\right] \\
&\quad - \mathbb{E}_q\left[\log(\frac{q_{\phi_y}(\boldsymbol{z}^y|\boldsymbol{y}^{\text{o}})}{p(\boldsymbol{z}^y|\boldsymbol{x}, \lambda(\boldsymbol{t}))})\right] - \mathbb{E}_q\left[\log(\frac{q_{\phi_m}(\boldsymbol{z}^m|\boldsymbol{m})}{p(\boldsymbol{z}^m|\boldsymbol{x}, \lambda(\boldsymbol{t}))})\right] \\
&= \mathbb{E}_q\left[\log\left(p_{\psi_y}(\boldsymbol{y}^{\text{o}}|\boldsymbol{z}^y, \boldsymbol{z}^m, \boldsymbol{m}) p_{\psi_m}(\boldsymbol{m}|\boldsymbol{z}^y, \boldsymbol{z}^m) p(\boldsymbol{t}|\lambda)\right)\right] - \text{KL}(q(\boldsymbol{u})||p(\boldsymbol{u})) \\
&\quad - \mathbb{E}_{p(z^\lambda|\boldsymbol{u})\cdot q(\boldsymbol{u})}\left[\text{KL}(q_{\phi_y}(\boldsymbol{z}^y|\boldsymbol{y}^{\text{o}})||p(\boldsymbol{z}^y|\boldsymbol{x}, \lambda(\boldsymbol{t}))\right] - \mathbb{E}_{p(z^\lambda|\boldsymbol{u})\cdot q(\boldsymbol{u})}\left[\text{KL}(q_{\phi_m}(\boldsymbol{z}^m|\boldsymbol{m})||p(\boldsymbol{z}^m|\boldsymbol{x}, \lambda(\boldsymbol{t}))\right] \\
&\geq \mathbb{E}_q\left[\log\left(p_{\psi_y}(\boldsymbol{y}^{\text{o}}|\boldsymbol{z}^y, \boldsymbol{z}^m, \boldsymbol{m}) p_{\psi_m}(\boldsymbol{m}|\boldsymbol{z}^y, \boldsymbol{z}^m) p(\boldsymbol{t}|\lambda)\right)\right] - \text{KL}(q(\boldsymbol{u})||p(\boldsymbol{u})) \\
&\quad - \mathbb{E}_{p(z^\lambda|\boldsymbol{u})\cdot q(\boldsymbol{u})}\left[\hat{\text{KL}}(q_{\phi_y}(\boldsymbol{z}^y|\boldsymbol{y}^{\text{o}})||p(\boldsymbol{z}^y|\boldsymbol{x}, \lambda(\boldsymbol{t}))\right] - \mathbb{E}_{p(z^\lambda|\boldsymbol{u})\cdot q(\boldsymbol{u})}\left[\hat{\text{KL}}(q_{\phi_m}(\boldsymbol{z}^m|\boldsymbol{m})||p(\boldsymbol{z}^m|\boldsymbol{x}, \lambda(\boldsymbol{t}))\right],
\end{aligned}$$

where $\hat{\text{KL}}$ is the corresponding upper bound for KL divergence from Appendix B and expectations involving upper bounds are estimated via Monte Carlo sampling. The generative part can be further decomposed into

$$\begin{aligned}
&\mathbb{E}_q\left[\log\left(p_{\psi_y}(\boldsymbol{y}^{\text{o}}|\boldsymbol{z}^y, \boldsymbol{z}^m, \boldsymbol{m}) p_{\psi_m}(\boldsymbol{m}|\boldsymbol{z}^y, \boldsymbol{z}^m) p(\boldsymbol{t}|\lambda)\right)\right] \\
&\quad = \mathbb{E}_{q_\phi}\left[\log\left(p_{\psi_y}(\boldsymbol{y}^{\text{o}}|\boldsymbol{z}^y, \boldsymbol{z}^m, \boldsymbol{m})\right)\right] + \mathbb{E}_{q_\phi}[\log\left(p_{\psi_m}(\boldsymbol{m}|\boldsymbol{z}^y, \boldsymbol{z}^m)\right)] + \mathbb{E}_{p(z^\lambda|\boldsymbol{u})\cdot q(\boldsymbol{u})}\left[\log(p(\boldsymbol{t}|\lambda)\right],
\end{aligned}$$

where $q_\phi(\boldsymbol{z}^y, \boldsymbol{z}^m|\boldsymbol{y}^{\text{o}}, \boldsymbol{m}) = q_{\phi_y}(\boldsymbol{z}^y|\boldsymbol{y}^{\text{o}}) \cdot q_{\phi_m}(\boldsymbol{z}^m|\boldsymbol{m})$.

The expectations involving $\boldsymbol{z}^y$ and $\boldsymbol{z}^m$ are computed by sampling from their variational distributions and utilizing the reparameterization trick (Kingma & Welling, 2014), whereas expectation involving likelihood of the point process can be evaluated in a closed from due to the chosen squared link function as we show below for the timepoints of individual p.

To lighten the notation, we use $\mathcal{L}_T := \mathbb{E}_{p(\boldsymbol{z}^\lambda|\boldsymbol{u})\cdot q(\boldsymbol{u})}\left[\log(p(\boldsymbol{t}|\lambda)\right]$, drop subscript p denoting the individual and drop subscript

D from v_D. Also, by $\boldsymbol{s}$ we denote inducing point locations of $\boldsymbol{u}$. First, we integrate out $\boldsymbol{u}$:

$$
\begin{aligned}
q(z^\lambda) &= \int_U p(z^\lambda|\boldsymbol{u}) \cdot q(\boldsymbol{u}) d\boldsymbol{u} = N(z^\lambda|\tilde{\mu}, \tilde{\Sigma}), \\
\text{where} \qquad \tilde{\mu}(v) &= k_{v\boldsymbol{s}} K_{\boldsymbol{ss}}^{-1} m_\lambda, \\
\tilde{\Sigma}(v, v') &= K_{vv'} - k_{v\boldsymbol{s}} K_{\boldsymbol{ss}}^{-1} k_{\boldsymbol{s}v'} + k_{v\boldsymbol{s}} K_{\boldsymbol{ss}}^{-1} S K_{\boldsymbol{ss}}^{-1} k_{\boldsymbol{s}v'},
\end{aligned}
$$

and U is a space of inducing values. We write $\mathcal{L}_T$ as

$$
\mathcal{L}_T = \sum_n \mathbb{E}_{q(z^\lambda)} \left[\log \lambda(t_n)\right] - \mathbb{E}_{q(z^\lambda)} \Big[\int_T \lambda(t) dt \Big] = \sum_n \mathbb{E}_{q(z^\lambda)} \left[\log \left(z^\lambda(t_n) + \beta\right)^2\right] - \underbrace{\mathbb{E}_{q(z^\lambda)} \Big[\int_T \lambda(t) dt \Big]}_{:=\mathcal{L}_t},
$$

where we sum over all timepoints of the individual. Via Lloyd et al. (2015), we have

$$
\begin{aligned}
\mathbb{E}_{q(z^\lambda)} \left[\log \left(z^\lambda(t_n) + \beta\right)^2\right] &= \int_{-\infty}^{\infty} \log((z^\lambda(t_n) + \beta)^2) N(z^\lambda(t_n)|\tilde{\mu}, \tilde{\sigma}^2) dz^\lambda(t_n) \\
&= -\tilde{G}\Big(- \frac{(\tilde{\mu} + \beta)^2}{2\tilde{\sigma}^2}\Big) + \log\Big(\frac{\tilde{\sigma}^2}{2}\Big) - C,
\end{aligned}
$$

where $\tilde{\sigma}^2$ is the diagonal element of $\tilde{\Sigma}$, C is the Euler-Mascheroni constant and $\tilde{G}$ is the confluent hyper-geometric function. Following the derivations of Liu & Hauskrecht (2019), we compute $\mathcal{L}_t$ as:

$$
\begin{aligned}
\mathcal{L}_t &= \mathbb{E}_{q(z^\lambda)} \left[\int_T \lambda(t) dt\right] = \mathbb{E}_{q(z^\lambda)} \left[\int_T (z^\lambda(t) + \beta)^2 dt\right] \\
&= \int_T \mathbb{E}_{q(z^\lambda)} \left[(z^\lambda(t) + \beta)^2\right] dt \\
&= \int_T \left(\mathbb{E}_{q(z^\lambda)} \left[(z^\lambda(t)^2\right] + 2\beta \mathbb{E}_{q(z^\lambda)} \left[z^\lambda(t)\right] + \beta^2\right) dt \\
&= \int_T \mathbb{E}_{q(z^\lambda)} \left[z^\lambda(t)\right]^2 dt + \int_T \text{Var}_{q(z^\lambda)} \left[z^\lambda(t)\right] dt + 2\beta \int_T \mathbb{E}_{q(z^\lambda)} \left[z^\lambda(t)\right] dt + \beta^2 |T| \\
&= \sum_n \left[\int_{t_{n-1}}^{t_n} \mathbb{E}_{q(z^\lambda)} \left[z^\lambda(t)\right]^2 dt + \int_{t_{n-1}}^{t_n} \text{Var}_{q(z^\lambda)} \left[z^\lambda(t)\right] dt + 2\beta \int_{t_{n-1}}^{t_n} \mathbb{E}_{q(z^\lambda)} \left[z^\lambda(t)\right] dt\right] + \beta^2 |T|.
\end{aligned}
$$

Each integral is computed as follows:

$$
\begin{aligned}
\int_{\boldsymbol{t}_{n-1}}^{\boldsymbol{t}_n} \mathbb{E}_{q(z^\lambda)} \left[z^\lambda(t)\right]^2 dt &= m_\lambda^T K_{\boldsymbol{ss}}^{-1} \Psi_n K_{\boldsymbol{ss}}^{-1} m_\lambda, \\
\int_{t_{n-1}}^{t_n} \text{Var}_{q(z^\lambda)} \left[z^\lambda(t)\right] dt &= \sum_{d=1}^{D} \gamma_d \int_{t_{n-1}}^{t_n} \mathbb{1}(v_d) dt - \text{tr}(K_{\boldsymbol{ss}}^{-1} \Psi_n) + \text{tr}(K_{\boldsymbol{ss}}^{-1} S K_{\boldsymbol{ss}}^{-1} \Psi_n), \\
\int_{t_{n-1}}^{t_n} \mathbb{E}_{q(z^\lambda)} \left[z^\lambda(t)\right] dt &= \Phi_n(\boldsymbol{s})^T K_{\boldsymbol{ss}}^{-1} m_\lambda, \\
\Psi_n(\boldsymbol{s}, \boldsymbol{s}') &= \int_{t_{n-1}}^{t_n} K(\boldsymbol{s}, v(t)) K(v(t), \boldsymbol{s}') dt, \\
\Phi_n(\boldsymbol{s}) &= \int_{t_{n-1}}^{t_n} K(\boldsymbol{s}, v(t)) dt.
\end{aligned}
$$

Φ and Ψ each have closed form solutions which we obtain by evaluating the integrals for the sum of SE kernels.

$$\Phi_n(\boldsymbol{s}) = \sum_{d=1}^{D} \mathbb{1}(\boldsymbol{s}_d)\mathbb{1}(v_{d_n})\gamma_d \frac{\sqrt{\pi l_d^2}}{\sqrt{2}} \left[\text{erf}\left(\frac{v_{dn} - \boldsymbol{s}_d}{\sqrt{2l_d^2}} \right) - \text{erf}\left(\frac{v_{dn-1} - \boldsymbol{s}_d}{\sqrt{2l_d^2}} \right) \right],$$

$$\Psi_n(\boldsymbol{s}, \boldsymbol{s}') = \sum_{i,j=1}^{D} \mathbb{1}(\boldsymbol{s}_i)\mathbb{1}(\boldsymbol{s}'_j)\mathbb{1}(v_{in})\mathbb{1}(v_{j_n})\gamma_i\gamma_j \frac{\sqrt{\pi l_i^2 l_j^2}}{\sqrt{2 \cdot (l_i^2 + l_j^2)}} \exp\left(-\frac{(\boldsymbol{s}_i + t_{in} - \boldsymbol{s}'_j - t_{j_n})^2}{2(l_i^2 + l_j^2)} \right)$$
$$\left[\text{erf}\left(\frac{l_i^2(v_{j_n} - \boldsymbol{s}'_j) + l_j^2(v_{in} - \boldsymbol{s}_i)}{\sqrt{2l_i^2 l_j^2 (l_i^2 + l_j^2)}} \right) - \text{erf}\left(\frac{l_i^2(v_{j_{n-1}} - \boldsymbol{s}'_j) + l_j^2(v_{in-1} - \boldsymbol{s}_i)}{\sqrt{2l_i^2 l_j^2 (l_i^2 + l_j^2)}} \right) \right],$$

where l_i, γ_i are kernel hyperparameters.

D. Predictive distribution

Given training samples $\boldsymbol{y}^{\text{o}}$, covariate information $\boldsymbol{x}$ and masks $\boldsymbol{m}$, the predictive distribution for a new observation y_*, given covariates x_* is

$$\begin{aligned} p(y_* \mid x_*, \boldsymbol{y}^{\text{o}}, \boldsymbol{x}, \boldsymbol{m}) &= \int p(y_* \mid z_*^y, z_*^m) p(z_*^y, z_*^m \mid x_*, \boldsymbol{y}^{\text{o}}, \boldsymbol{x}, \boldsymbol{m})\, dz_*^y\, dz_*^m \\ &\approx \int \underbrace{p(y_* \mid z_*^y, z_*^m)}_{\text{decode GP predictions}} \underbrace{p(z_*^y \mid x_*, \lambda(t_*), \boldsymbol{z}^y, \boldsymbol{x}, \lambda(\boldsymbol{t}))}_{\text{GP posterior of } z_*^y} \underbrace{p(z_*^m \mid x_*, \lambda(t_*), \boldsymbol{z}^m, \boldsymbol{x}, \lambda(\boldsymbol{t}))}_{\text{GP posterior of } z_*^m} \\ &\quad \underbrace{q(z_*^\lambda)}_{\text{variational posterior of } z_*^\lambda} \underbrace{q_{\phi_y}(\boldsymbol{z}^y \mid \boldsymbol{y}^{\text{o}})}_{\text{encode data}} \underbrace{q_{\phi_m}(\boldsymbol{z}^m \mid \boldsymbol{m})}_{\text{encode mask}}\, dz_*^\lambda\, dz_*^y\, dz_*^m\, d\boldsymbol{z}^y\, d\boldsymbol{z}^m, \end{aligned}$$

where timestamps $\boldsymbol{t}$ and t_* belong to $\boldsymbol{x}$ and x_*, respectively, and $p(z_*^y \mid x_*, \lambda(t_*), \boldsymbol{z}^y, \boldsymbol{x}, \lambda(\boldsymbol{t}))$ and $p(z_*^m \mid x_*, \lambda(t_*), \boldsymbol{z}^m, \boldsymbol{x}, \lambda(\boldsymbol{t}))$ are GP posteriors such that

$$\begin{aligned} &p(z_* \mid x_*, \lambda(t_*), \boldsymbol{z}, \boldsymbol{x}, \lambda(\boldsymbol{t})) = \mathcal{N}(\tilde{\mu}, \tilde{\Sigma}), \\ &\tilde{\mu} = K_{w_*\boldsymbol{w}}(K_{\boldsymbol{w}\boldsymbol{w}} + \sigma_z^2 I_N)^{-1}\boldsymbol{z}, \\ &\tilde{\Sigma} = K_{w_*w_*} + \sigma_z^2 I_{N_*} - K_{w_*\boldsymbol{w}}(K_{\boldsymbol{w}\boldsymbol{w}} + \sigma_z^2 I_N)^{-1} K_{\boldsymbol{w}w_*}, \end{aligned}$$

where $\boldsymbol{w} = (\boldsymbol{x}, \lambda(\boldsymbol{t}))$, $w_* = (x_*, \lambda(t_*))$ and N_* is a number of elements for prediction. Above, we dropped the superscripts, meaning that the same formulae hold for both z^y and z^m with respect to their kernel hyperparameters. The same approach holds for deriving $p(m_* \mid x_*, \boldsymbol{y}^{\text{o}}, \boldsymbol{x}, \boldsymbol{m})$. In order to sample from these predictive distributions, ancestral sampling can be employed. Computing the predictive distributions scales cubically for this case. To get the idea of how to implement a scalable predictive distribution using low-rank inducing point approximation, we refer the reader to the derivations by Ramchandran et al. (2021).

E. Experimental setups

We employed identical kernel structures for the GPs of both z^y and z^m across all of the datasets mentioned below. Nonetheless, it's important to note that, in general, these kernel structures could differ between the two. We also selected sixty inducing points for each GP model for all setups and chose the latent dimension to be 32.

E.1. HealthMNIST variants

For the *Health MNIST regularly sampled* dataset we use the following covariates: *time*, *id*, *diseasePresence*, *diseaseTime* and *gender*. When running LLSM on the corresponding dataset, we relied on the following additive kernel structure

$$f_{\text{ca}}(\text{id}) + f_{\text{se}}(\text{time}) + f_{\text{ca}\times\text{se}}(\text{id} \times \text{time}) + f_{\text{ca}\times\text{se}}(\text{gender} \times \text{time}) + f_{\text{ca}\times\text{se}}(\text{diseasePresence} \times \text{diseaseTime}),$$

where se denotes squared exponential kernel and ca is referred to categorical one.

For *Health MNIST irregularly sampled* dataset, *time*, *id*, *gender* and *diseasePresence* were used as covariates for LLSM and LLPPSM. In case of LLPPSM, we also added intensity of the point process for covariance computation. When using LLSM, we employed the following kernel structure

$$f_{\text{ca}}(\text{id}) + f_{\text{se}}(\text{time}) + f_{\text{ca}\times\text{se}}(\text{id} \times \text{time}) + f_{\text{ca}\times\text{se}}(\text{gender} \times \text{time}),$$

whereas for LLPPSM

$$f_{\text{ca}}(\text{id}) + f_{\text{se}}(\text{time}) + f_{\text{ca}\times\text{se}}(\text{id} \times \text{time}) + f_{\text{ca}\times\text{se}}(\text{gender} \times \text{time}) + f_{\text{ca}\times\text{se}}(\text{id} \times \text{intensity}) + f_{\text{ca}\times\text{se}}(\text{gender} \times \text{intensity}).$$

For both variants, we used the Adam optimiser (Kingma & Ba, 2015) as implemented in Pytorch (Paszke et al., 2019), with a learning rate equal to 0.001, which was selected based on cross-validation. After having pretrained a standard VAE, we trained both LLSM and LLPPSM on 1000 epochs, employing early stopping.

When training LLPPSM, we defined separate β parameters for "healthy" and "sick" instances and optimised them jointly together with other parameters. For the temporal point process part of the model, the number of previous timestamps, D, is 15.

E.2. Physionet dataset

We selected 2000 patients for training, 1917 for validation and performed future prediction on 100 patients, not included in training and validation sets. We used the following covariates: *time*, *id*, *type of ICU*, *gender* and *in-hospital mortality*, with the corresponding additive kernel structure for LLSM

$$\begin{aligned} &f_{\text{ca}}(\text{id}) + f_{\text{se}}(\text{time}) + f_{\text{ca}\times\text{se}}(\text{id} \times \text{time}) + f_{\text{ca}\times\text{se}}(\text{gender} \times \text{time}) \\ &+f_{\text{ca}\times\text{se}}(\text{type of ICU} \times \text{time}) + f_{\text{ca}\times\text{se}}(\text{in-hospital mortality} \times \text{time}), \end{aligned}$$

and for LLPPSM, including intensity as an additional variable

$$\begin{aligned} &f_{\text{ca}}(\text{id}) + f_{\text{se}}(\text{time}) + f_{\text{ca}\times\text{se}}(\text{id} \times \text{time}) + f_{\text{ca}\times\text{se}}(\text{gender} \times \text{time}) + f_{\text{ca}\times\text{se}}(\text{type of ICU} \times \text{time}) \\ &+f_{\text{ca}\times\text{se}}(\text{in-hospital mortality} \times \text{time}) + f_{\text{ca}\times\text{se}}(\text{id} \times \text{intensity}) + f_{\text{ca}\times\text{se}}(\text{gender} \times \text{intensity}) \\ &+f_{\text{ca}\times\text{se}}(\text{type of ICU} \times \text{intensity}) + f_{\text{ca}\times\text{se}}(\text{in-hospital mortality} \times \text{intensity}), \end{aligned}$$

The optimisation was done by Adam optimiser (Kingma & Ba, 2015) using Pytorch (Paszke et al., 2019), with the learning rate 0.001. Both models were trained for 400 epochs, employing early stopping, after having pretrained them by standard VAE.

When training LLPPSM, we defined separate β parameters based on in-hospital mortality attribute and optimised them jointly with other parameters. For temporal point process part, the number of previous timestamps, D, is 15.

Moreover, we found it useful to drop the dependence from z^m to y as was discussed in Section 3.2. Our intuition is that in this case, if the dependence is left, we are obliged to apply predictive distribution (Appendix D) for both z^y and z^m, which, due to the highly complex and sparse nature of this dataset, cannot be modelled highly accurately, leading to the accumulation of additional error.

F. Neural network architectures

Table 9 describes neural network architecture used for both Health MNIST setups that consists of convolutional and feedforward layers. Table 10 describes neural network architecture for the Physionet dataset. In this case, we employed a multi layered perceptron (MLP).

G. Supplementary figures

Table 9. Neural Network architecture used in Health MNIST variants

	Hyperparameter	Value
Inference network of both z^y and z^m	Dimensionality of input	36×36
	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 of both y and m	Dimensionality of input	L
	Number of transposed convolution layers	3
	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 10. Neural Network architecture used in Physionet dataset

	Hyperparameter	Value
Inference network of both z^y and z^m	Dimensionality of input	37
	Number of feedforward layers	2
	Width of feedforward layers	$300, 30$
	Dimensionality of latent space	L
	Activation function of layers	RELU
Generative network of both y and m	Dimensionality of input	L
	Number of feedforward layers	3
	Width of feedforward layers	$30, 30, 300$
	Activation function of layers	RELU

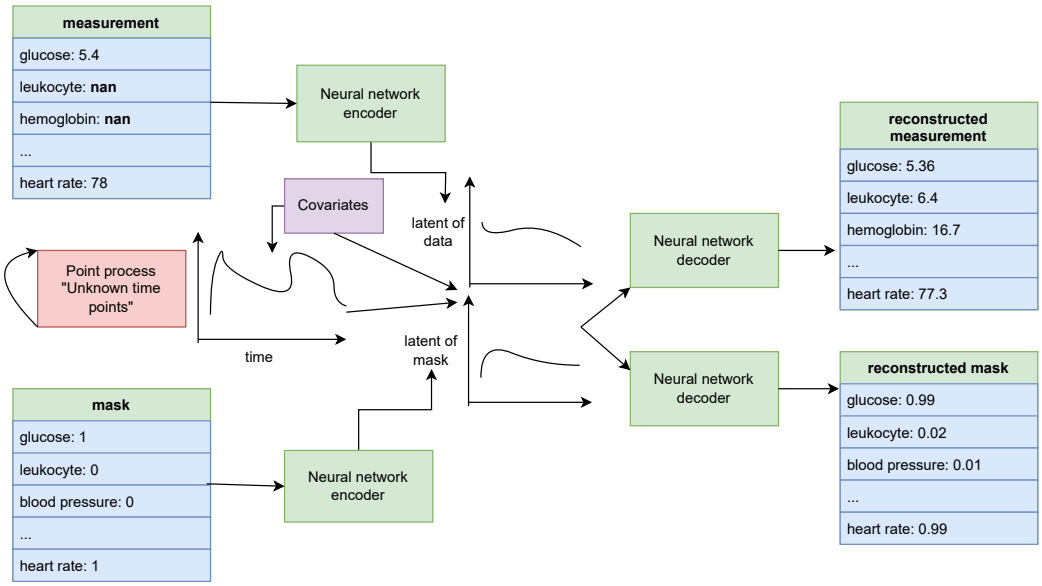

Figure 6. Detailed overview of our model

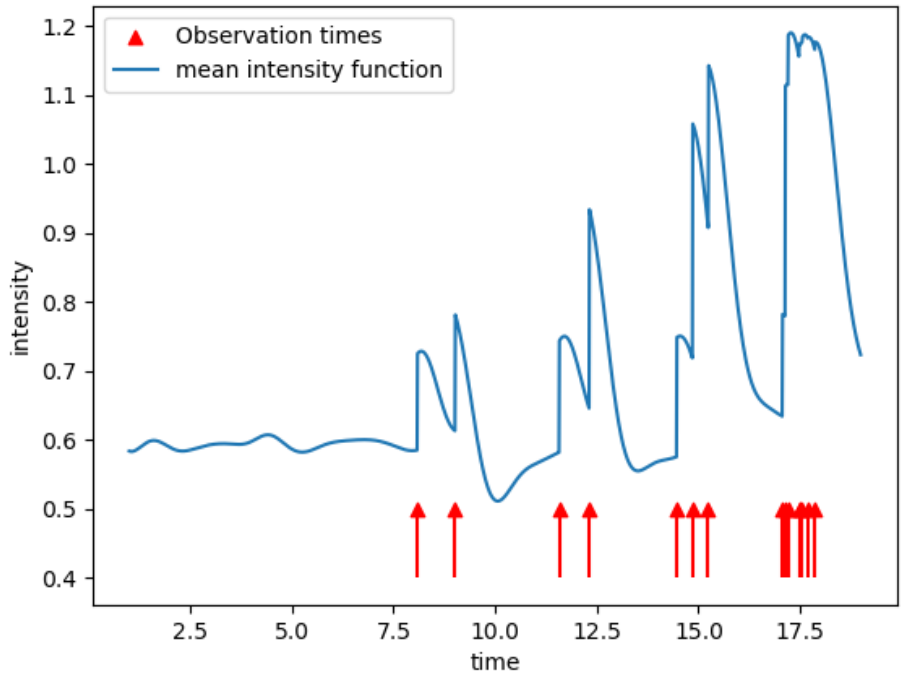

Figure 7. Inferred mean intensity function of the point process