Machine Learning
https://doi.org/10.1007/s10994-023-06382-1

Heterogeneous multi-task Gaussian Cox processes

Feng Zhou[1,2] · Quyu Kong[3] · Zhijie Deng[2,4] · Fengxiang He[5] · Peng Cui[2] · Jun Zhu[2]

Received: 15 November 2022 / Revised: 3 July 2023 / Accepted: 31 July 2023

Abstract

This paper presents a novel extension of multi-task Gaussian Cox processes for modeling multiple heterogeneous correlated tasks jointly, e.g., classification and regression, via multi-output Gaussian processes (MOGP). A MOGP prior over the parameters of the dedicated likelihoods for classification, regression and point process tasks can facilitate sharing of information between heterogeneous tasks, while allowing for nonparametric parameter estimation. To circumvent the non-conjugate Bayesian inference in the MOGP modulated heterogeneous multi-task framework, we employ the data augmentation technique and derive a mean-field approximation to realize closed-form iterative updates for estimating model parameters. We demonstrate the performance and inference on both 1D synthetic data as well as 2D urban data of Vancouver.

Keywords Heterogeneous correlation · Multi-task learning · Cox process · Multi-output Gaussian processes · Conditionally conjugate

Editor: Willem Waegeman.

✉ Jun Zhu
dcszj@tsinghua.edu.cn

Feng Zhou
feng.zhou@ruc.edu.cn

Quyu Kong
quyu.kong@uts.edu.au

Zhijie Deng
zhijied@sjtu.edu.cn

Fengxiang He
fengxiang.f.he@gmail.com

Peng Cui
cuip22@mails.tsinghua.edu.cn

[1] Center for Applied Statistics and School of Statistics, Renmin University of China, Beijing, China

[2] Department of Computer Science and Technology, BNRist Center, THU-Bosch Joint ML Center, Tsinghua University, Beijing, China

[3] Data Science Institute, University of Technology Sydney, Sydney, Australia

[4] Qing Yuan Research Institute, Shanghai Jiao Tong University, Shanghai, China

[5] JD Explore Academy, JD.com Inc, Beijing, China

Published online: 08 September 2023

1 Introduction

Inhomogeneous Poisson process data defined on a continuous spatio-temporal domain has attracted immense attention recently in a wide variety of applications, including reliability analysis in manufacturing systems (Soleimani et al., 2017), event capture in sensing regions (Mutny and Krause, 2021), crime prediction in urban area (Shirota and Gelfand, 2017) and disease diagnosis based on medical records (Lasko, 2014). The reliable training of an inhomogeneous Poisson process model critically relies on a large amount of data to avoid overfitting, especially when modeling high-dimensional point processes. However, one challenge is that the available training data is routinely sparse or even partially missing in specific applications. Taking manufacturing failure and healthcare analysis as motivating examples: the modern manufacturing machines are reliable and sparsely fail; the individuals with healthy constitution will not visit hospital very often. The data missing problems also arise, e.g., the event location capture is intermittent for sensing systems because of weather or other related barriers. To handle data sparse/missing problems, the correlation between multiple tasks can be exploited to facilitate sharing of information between all tasks to improve the generalization capabilities, forming a multi-task learning paradigm.

A popular approach to modeling multi-task inhomogeneous Poisson processes is to use Gaussian process (GP) (Williams & Rasmussen, 2006) based Bayesian framework to induce correlation among tasks. This kind of multi-task inhomogeneous Poisson processes are also called multi-task Cox processes (Møller et al., 1998). Multi-task Cox processes have been investigated extensively in recent years, e.g., hierarchical-GP based version (Lian et al., 2015) and multi-output Gaussian processes (MOGP) based versions (Aglietti et al., 2019; Jahani et al., 2021). Yet to our knowledge, all the aforementioned works focus on *homogeneous* multi-task Cox processes learning, i.e., all correlated tasks are exclusively point process tasks. It is not free to apply them to the more general *heterogeneous* multi-task scenarios where correlated tasks include other types of tasks except Cox processes. Take the urban data of Vancouver in Fig. 3 as a motivating example where we have three types of tasks: employment income (regression), education degree (classification), theft of vehicle (Cox process) and non-market house (Cox process). When the crime data is missing in certain areas of the city, training on this single task is prone to overfitting since the model may try to fit the available data too closely, leading to inaccurate predictions or poor generalization to unseen data. Can we leverage the information of employment income, education degree and non-market housing to assist the prediction of crime rate in the missing areas? Or, can we make use of income, education and crime to help predict the number of non-market housing projects in certain missing areas? Based on our knowledge, only a few heterogeneous frameworks exist, such as Moreno-Muñoz et al. (2018). However, Moreno-Muñoz et al. (2018) discretized the point process task into Poisson distribution problems and does not preserve conjugate operations. To make further progress, we generalize the homogeneous multi-task Cox processes to the heterogeneous setup using MOGP to enable the transfer of knowledge between supervised (regression and classification) and unsupervised tasks (Cox processes).

Most existing Cox process works focus on the log Gaussian Cox process (LGCP) (Møller et al., 1998) where a GP function is passed through an exponential link function to model the positive intensity rate. Due to the nonconjugacy between point process likelihood and GP prior, practitioners need to apply Markov chain Monte Carlo (MCMC) (Neal, 1993) or variational inference (Blei et al., 2017) methods to infer the posterior distribution of model parameters. For MCMC, the specialised MCMC algorithms, such as Metropolis-adjusted Langevin algorithm (MALA) (Møller et al., 1998; Besag, 1994), as well as the probabilistic programming languages based on MCMC (Wood et al., 2014) where one does not need to write a sampler by hand, can be used for sampling from the posterior of intensity function. Although MCMC provides the guarantee of asymptotic consistency, this accuracy comes at the expense of a high computational cost. On the contrary, variational inference can be faster than MCMC, although it induces approximation error. For the efficiency reason, we focus on variational inference in this work. For variational inference, a Gaussian variational posterior is typically assumed to render the evidence lower bound (ELBO) tractable (Dezfouli & Bonilla, 2015; Lloyd et al., 2015). While this variational inference method is quite generic, it can exhibit low efficiency (although it is still faster than MCMC) (Wenzel et al., 2019), exposing opportunities for improvement. It is worth noting that the same problem also occurs in GP classification tasks. This work remediates these issues by basing our model on sigmoidal Gaussian Cox process (SGCP) (Adams et al., 2009), using a scaled sigmoid function as link function in point process tasks, and the logistic regression model in classification tasks. The reason we choose sigmoid as link function in both types of tasks is we can exploit the data augmentation technique (Polson et al., 2013; Donner & Opper, 2018) to construct a mean-field approximation that has closed-form iterative updates. As shown later, the proposed mean-field approximation exhibits superior efficiency and fast convergence.

Specifically, we make the following contributions. (1) From a modeling perspective, we establish a MOGP based *heterogeneous multi-task Gaussian Cox processes* (HMGCP) model that provides an extension of the homogeneous version to account for multiple heterogeneous correlated tasks. (2) From an inference perspective, we adopt the data augmentation technique to derive *an efficient mean-field approximation with analytical expressions*. As far as we know, this work should be the first attempt to use data augmentation in the MOGP setting. (3) In experiments, we provide evidence of the benefits of modeling heterogeneous correlated tasks and the predominant efficiency and convergence of our inference method.

2 Related work

2.1 Multi-output Gaussian processes

Multi-output Gaussian processes (Álvarez et al., 2012) extend the single-output Gaussian process to model vector-valued functions, providing a powerful Bayesian tool for multi-task learning as it accounts for the correlation between multiple outputs. Bonilla et al. (2007) has shown that if multiple outputs are correlated, exploiting such correlation can provide insightful information about each output and better predictions in the case of sparse/missing data. More importantly, as a Bayesian nonparametric approach, it offers higher flexibility over parametric alternatives and a natural mechanism for uncertainty quantification. To define a MOGP, we need to define a suitable cross-covariance function that accounts for the correlation between multiple outputs, which leads to a valid covariance function for the joint GP (Álvarez et al., 2019). The two common ways to define cross-covariance functions are linear model of coregionalization (LMC) (Journel & Huijbregts, 1976) and process convolution (Ver Hoef & Barry, 1998). In this work, we focus on the LMC approach.

2.2 Multi-task Cox processes

Extensive works have been accumulated on the single-task Gaussian Cox process (Møller et al., 1998; Diggle et al., 2013). Recently, many works tried to extend the single-task Cox process to the multi-task setup to introduce correlation between tasks. For example, Lian et al. (2015) proposed a multi-task Cox process model that leverages information from all tasks via a hierarchical GP. In a different way, Aglietti et al. (2019) and Jahani et al. (2021) adopted the MOGP based on LMC and process convolution respectively to model the intensity functions of multiple Cox processes, which facilitates sharing of information and allows for flexible event occurrence rate. All these works exclusively focus on homogeneous multi-task Cox processes. On the contrary, we extend to the heterogeneous scenarios to enable transfer of knowledge between Cox process, regression and classification tasks.

2.3 Data augmentation

In GP regression, the conjugacy between likelihood and prior makes the posterior computing easy and closed-form. However, in GP classification and point process, such conjugacy no longer holds and one may resort to variational inference to approximate the true posterior. Most generic non-conjugate variational inference, assuming a Gaussian variational posterior to make the ELBO tractable, exhibits low efficiency due to computing of expectations (Dezfouli & Bonilla, 2015). Recently, another inference method based on data augmentation[1] has been established for GP classification (Polson et al., 2013; Wenzel et al., 2019) and point process (Donner & Opper, 2018; Zhou et al., 2020; Zhou et al., 2021; Zhou et al., 2022). The core idea is to augment likelihood by auxiliary latent variables to convert the non-conjugate problem to a conditionally conjugate one, thus making inference easy (Li et al., 2014). Here, such an idea is extended to the MOGP modulated multi-task framework.

3 Problem formulation

Traditionally, existing works have considered the homogeneous multi-task Cox processes learning where all tasks are Cox processes (Aglietti et al., 2019; Jahani et al., 2021). The homogeneous model is not applicable to the more general heterogeneous scenario which includes various types of tasks except Cox processes. In this work, we are interested in the more general heterogeneous scenario where correlated tasks are a mix of supervised (regression and classification) and unsupervised tasks (Cox processes). Let us consider a problem setting where we have data from I tasks, among which I_r tasks are regression problems with dataset $\mathcal{D}_r=\{\{(\mathbf{x}^r_{i,n},y^r_{i,n})\}_{n=1}^{N_i^r}\}_{i=1}^{I_r}$, I_c tasks are classification problems with dataset $\mathcal{D}_c=\{\{(\mathbf{x}^c_{i,n},y^c_{i,n})\}_{n=1}^{N_i^c}\}_{i=1}^{I_c}$ and I_p tasks are point process problems with dataset $\mathcal{D}_p=\{\{(\mathbf{x}^p_{i,n})\}_{n=1}^{N_i^p}\}_{i=1}^{I_p}$. $\mathbf{x}\in\mathcal{X}\subset\mathbb{R}^D$ is the D-dimensional input; $y\in\mathbb{R}$ is the output in regression tasks and $\{-1,1\}$ in classification tasks.[2] Point process tasks are unsupervised learning problems so they only include $\mathbf{x}$. Throughout the paper, we use index r, c, p to indicate regression, classification and point process tasks, respectively.

[1] The notion of data augmentation in statistics is different from that in deep learning.

[2] We focus on binary classification here. Extension to multi-class classification is discussed in Sect. D.

3.1 Heterogeneous likelihood

In order to use GP to represent the likelihood parameters in three types of tasks, we need to design the appropriate transformation to map the GP output to the domain of specific parameters. For regression tasks, following tradition, we use Gaussian distribution as likelihood, where the mean is modeled as a GP function and the variance is treated as a hyperparameter. For binary classification tasks, we use Bernoulli distribution (Uspensky, 1937) as likelihood whose parameter is modeled by the sigmoid transformation of a GP function, mapping $\mathbb{R} \rightarrow [0,1]$, which is also called logistic regression. For Cox process tasks, although many existing works focus on LGCP, our work adopts the SGCP instead, i.e., the intensity of i-th Cox process is assumed to be $\lambda_i(\mathbf{x}) = \bar{\lambda}_i s(g_i(\mathbf{x}))$ where a task-specific GP function g_i is passed through a sigmoid function $s(\cdot)$ and then scaled by an upper-bound $\bar{\lambda}_i$. The reason we choose the sigmoid link function in both classification and point process tasks is that we can exploit the data augmentation to make inference easy and fast. Specifically, three types of likelihoods are:

$$p(\mathbf{y}^r \mid \{g_i^r\}_{i=1}^{I_r}) = \prod_{i=1}^{I_r} \prod_{n=1}^{N_i^r} \mathcal{N}(y_{i,n}^r \mid g_{i,n}^r, \sigma_i^2), \tag{1a}$$

$$p(\mathbf{y}^c \mid \{g_i^c\}_{i=1}^{I_c}) = \prod_{i=1}^{I_c} \prod_{n=1}^{N_i^c} s(y_{i,n}^c g_{i,n}^c), \tag{1b}$$

$$p(\mathbf{x}^p \mid \{\bar{\lambda}_i, g_i^p\}_{i=1}^{I_p}) = \prod_{i=1}^{I_p} \prod_{n=1}^{N_i^p} \bar{\lambda}_i s(g_{i,n}^p) \exp\left(-\int_{\mathcal{X}} \bar{\lambda}_i s(g_i^p(\mathbf{x})) d\mathbf{x}\right), \tag{1c}$$

where g_i is the task-specific GP function and we call it latent function (Rasmussen, 2003) afterwards; g_i^r, g_i^c, g_i^p are the corresponding i-th output of the regression, classification and point process tasks, respectively; $g_{i,n}^{\cdot}$ indicates $g_i^{\cdot}(\mathbf{x}_{i,n}^{\cdot})$. Equation (1a) is the likelihood for regression; Eq. (1b) is the likelihood for binary classification; Eq. (1c) is the likelihood for point process (Daley and Vere-Jones, 2003).

3.2 MOGP prior

Instead of modeling each g_i independently, we apply the MOGP prior on g's to introduce correlation between multiple tasks in order to improve the generalization capability of our model especially when data is sparse or missing. In this work, we use the LMC (Journel & Huijbregts, 1976) approach to define the cross-covariance function. Specifically, we assume each latent function g_i is a linear combination of Q basis functions which are drawn from Q independent zero-mean GP prior, i.e., $\{f_q \sim \mathcal{GP}(0, k_q)\}_{q=1}^{Q}$ where k_q is a covariance function. Each latent function can be written as $g_i = \sum_{q=1}^{Q} w_{i,q} f_q$ where $w_{i,q} \in \mathbb{R}$ is the mixing weight capturing the contribution of q-th basis function to i-th latent function. It is easy to see that the mean of g_i is zero and the cross-covariance $k_{g_i,g_j}(\mathbf{x}, \mathbf{x}') = \text{cov}[g_i(\mathbf{x}), g_j(\mathbf{x}')] = \sum_{q=1}^{Q} w_{i,q} w_{j,q} k_q(\mathbf{x}, \mathbf{x}')$. If we define $\mathbf{g}_i$ to be the vector of latent function values on the inputs of i-th task, we have the following MOGP prior: $\mathbf{g} \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$, where $\mathbf{g} = [\mathbf{g}_1^\top, \dots, \mathbf{g}_I^\top]^\top$, $I = I_r + I_c + I_p$, $\mathbf{K}$ is a blockwise matrix with blocks given by $\{\mathbf{K}_{\mathbf{g}_i,\mathbf{g}_j}\}_{i=1,j=1}^{I,I}$ whose entries are $k_{g_i,g_j}(\mathbf{x}, \mathbf{x}')$. $\mathbf{x}$ and $\mathbf{x}'$ are the inputs of i-th and j-th tasks, respectively. It is worth noting that each task can have a different set of inputs, but when all tasks have the same set of inputs, e.g., $\mathbf{X}$, the computing of $\mathbf{K}$ can be simplified as the sum of Kronecker products $\mathbf{K} = \sum_{q=1}^{Q} \mathbf{w}_q \mathbf{w}_q^\top \otimes \mathbf{K}_q$ where $\mathbf{w}_q = [w_{1,q}, \dots, w_{I,q}]^\top$, $\mathbf{K}_q$ is the square matrix of $k_q(\mathbf{x}, \mathbf{x}')$ with $\mathbf{x}, \mathbf{x}' \in \mathbf{X}$ (Moreno-Muñoz et al., 2018). This property cooperates well with the inducing inputs formalism which is discussed later.

4 Inference

According to Bayes' theorem, the posterior of latent functions and intensity upper-bounds can be computed as:

$$p(g, \bar{\lambda} \mid \mathbf{y}^r, \mathbf{y}^c, \mathbf{x}^p) \propto$$
$$\underbrace{p(\mathbf{y}^r \mid \{g_i^r\}_{i=1}^{I_r})}_{\text{regression}} \underbrace{p(\mathbf{y}^c \mid \{g_i^c\}_{i=1}^{I_c})}_{\text{classification}} \underbrace{p(\mathbf{x}^p \mid \{\bar{\lambda}_i, g_i^p\}_{i=1}^{I_p})}_{\text{Cox process}} \underbrace{p(g)}_{\text{MOGP}} p(\bar{\lambda}),$$

where $g = [g_1, \dots, g_I]^\top$, $\bar{\lambda} = [\bar{\lambda}_1, \dots, \bar{\lambda}_{I_p}]^\top$, $p(g)$ is the infinite-dimensional version of MOGP, $p(\bar{\lambda}) \propto \prod_{i=1}^{I_p} \frac{1}{\bar{\lambda}_i}$ is the improper prior. The likelihood of regression is conjugate to the prior. However, such conjugacy is no longer valid for classification and Cox process tasks, so the posterior has no closed-form solution.

To address the non-conjugate issue for classification or Cox process, many works applied the variational inference that assumed a Gaussian variational distribution to render the ELBO tractable (Dezfouli & Bonilla, 2015; Hensman et al., 2015; Aglietti et al., 2019; Jahani et al., 2021). However, such generic variational inference exhibits low efficiency due to computing of expectations in ELBO (Wenzel et al., 2019). In this work, borrowing the idea of data augmentation, we augment Pólya-Gamma latent variables (Polson et al., 2013) and marked Poisson latent processes (Donner & Opper, 2018) into the likelihood of classification and Cox process. Finally, the augmented likelihood is conditionally conjugate to the MOGP prior. Based on the augmented model, we derive a mean-field approximation with closed-form iterative updates to provide an approximate posterior. The proofs of all relevant formulas below are provided in the appendix.

4.1 Augmentation for classification tasks

Polson et al. (2013) proposed a novel Pólya-Gamma augmentation strategy for Bayesian logistic regression. The core idea is that the binomial likelihood parametrized by log odds can be represented as a mixture of Gaussians w.r.t. a Pólya-Gamma distribution.

If $\omega \sim p_{\mathrm{PG}}(\omega \mid b, 0)$ denotes the Pólya-Gamma random variable with $\omega \in \mathbb{R}^+$ and $b > 0$, the following integral identity holds for $a \in \mathbb{R}$:

$$\frac{(e^z)^a}{(1+e^z)^b} = 2^{-b} e^{(a-b/2)z} \int_0^\infty e^{-z^2\omega/2} p_{\mathrm{PG}}(\omega \mid b, 0) d\omega.$$

In this work, we do not need to know the exact form of the Pólya-Gamma distribution, but only its first moment. Setting $a = b = 1$ yields the factorization of sigmoid function:

$$s(z) = \frac{e^z}{1+e^z} = \int_0^\infty e^{h(\omega,z)} p_{\mathrm{PG}}(\omega \mid 1,0) d\omega, \tag{2}$$

where $h(\omega, z) = z/2 - z^2\omega/2 - \log 2$. Substituting Eq. (2) into the classification likelihood in Eq. (1b), we obtain the augmented classification likelihood which has the elegant conditionally conjugate property. After augmenting Pólya-Gamma random variables, the logistic regression likelihood in Eq. (1b) is augmented to be:

$$p(\mathbf{y}^c, \boldsymbol{\omega}^c \mid \{g_i^c\}_{i=1}^{I_c}) = \prod_{i=1}^{I_c} \prod_{n=1}^{N_i^c} e^{h(\omega^c_{i,n}, y^c_{i,n} g^c_{i,n})} p_{\mathrm{PG}}(\omega^c_{i,n} \mid 1, 0), \tag{3}$$

where $\omega^c_{i,n}$ is the Pólya-Gamma latent variable on the n-th observed sample in the i-th classification task, $\boldsymbol{\omega}_i^c = [\omega^c_{i,1}, \dots, \omega^c_{i,N_i^c}]^\top$, $\boldsymbol{\omega}^c = [\boldsymbol{\omega}_1^{c\top}, \dots, \boldsymbol{\omega}_{I_c}^{c\ \top}]^\top$. The derivation is provided in Section A. The augmented classification likelihood in Eq. (3) is conditionally conjugate to the MOGP prior.

4.2 Augmentation for Cox process tasks

The augmentation for Cox process is more challenging than classification because the Cox process likelihood depends not only on the latent function values on observed samples but also on the whole latent function due to the exponential integral term. Borrowing the idea from Donner and Opper (2018), in addition to augmenting Pólya-Gamma latent variables on observed samples as in classification tasks, we also augment a marked Poisson latent process to linearize the exponential integral term.

Define a marked Poisson process $\Pi = \{(\mathbf{x}_r, \omega_r)\} \sim p(\Pi \mid \bar{\lambda} p_{\mathrm{PG}}(\omega \mid 1, 0))$ where $\mathbf{x}_r$ is the location of r-th point, the Pólya-Gamma latent variable ω_r denotes the independent mark at each point $\mathbf{x}_r$, $p(\Pi \mid \bar{\lambda} p_{\mathrm{PG}}(\omega \mid 1, 0))$ denotes the probability measure of Π with intensity $\Lambda(\mathbf{x}, \omega) = \bar{\lambda} p_{\mathrm{PG}}(\omega \mid 1, 0)$. Given the marked Poisson process defined above, the following identity holds:

$$\exp\left(-\int_{\mathcal{X}} \bar{\lambda} s(g(\mathbf{x})) d\mathbf{x}\right) = \mathbb{E}_{p_\Lambda} \prod_{(\omega,\mathbf{x})\in\Pi} e^{h(\omega, -g(\mathbf{x}))}, \tag{4}$$

where p_Λ indicates $p(\Pi \mid \Lambda(\mathbf{x}, \omega) = \bar{\lambda} p_{\mathrm{PG}}(\omega \mid 1, 0))$. Substituting Eqs. (2) and 4 into the Cox process likelihood in Eq. (1c), we obtain the augmented Cox process likelihood which has the conditionally conjugate property. After augmenting the Pólya-Gamma latent variables on observed samples and the marked Poisson latent process, the Cox process likelihood in Eq. (1c) is augmented to be:

$$\begin{gathered} p(\mathbf{x}^p, \boldsymbol{\omega}^p, \Pi \mid \bar{\lambda}, \{g_i^p\}_{i=1}^{I_p}) = \\ \prod_{i=1}^{I_p} \prod_{n=1}^{N_i^p} \Lambda_i(\mathbf{x}^p_{i,n}, \omega^p_{i,n}) e^{h(\omega^p_{i,n}, g^p_{i,n})} p_{\Lambda_i}(\Pi_i \mid \bar{\lambda}_i) \prod_{(\omega,\mathbf{x})\in\Pi_i} e^{h(\omega, -g_i^p(\mathbf{x}))}, \end{gathered} \tag{5}$$

where $\omega^p_{i,n}$ is the Pólya-Gamma latent variable on n-th observed sample in the i-th Cox process task, $\boldsymbol{\omega}_i^p = [\omega^p_{i,1}, \dots, \omega^p_{i,N_i^p}]^\top$, $\boldsymbol{\omega}^p = [\boldsymbol{\omega}_1^{p\top}, \dots, \boldsymbol{\omega}_{I_p}^{p\ \top}]^\top$, $\Lambda_i(\mathbf{x}, \omega) = \bar{\lambda}_i p_{\mathrm{PG}}(\omega \mid 1, 0)$, $\Pi = \{\Pi_i\}_{i=1}^{I_p}$. The derivation is provided in Section B. The augmented Cox process likelihood in Eq. (5) is conditionally conjugate to the MOGP prior.

4.3 Mean-field approximation

Based on the augmented likelihoods for classification and Cox process in Eqs. (3) and (5), we obtain the augmented joint distribution for all variables:

$$
\begin{aligned}
&p(\mathbf{y}^r, \mathbf{y}^c, \mathbf{x}^p, \boldsymbol{\omega}^c, \boldsymbol{\omega}^p, \Pi, g, \bar{\lambda}) = \\
&\underbrace{p(\mathbf{y}^r \mid \{g_i^r\}_{i=1}^{I_r})}_{\text{regression}} \underbrace{p(\mathbf{y}^c, \boldsymbol{\omega}^c \mid \{g_i^c\}_{i=1}^{I_c})}_{\text{augmented classification}} \underbrace{p(\mathbf{x}^p, \boldsymbol{\omega}^p, \Pi \mid \bar{\lambda}, \{g_i^p\}_{i=1}^{I_p})}_{\text{augmented Cox process}} \underbrace{p(g)}_{\text{MOGP}} p(\bar{\lambda}).
\end{aligned} \tag{6}
$$

Finally, our efforts are rewarded: after data augmentation, the model likelihood is conditionally conjugate to the prior and a simple Gibbs sampler can be derived to sample from the *exact posterior* by drawing a sample from each conditional distribution alternately. The samples of latent functions and intensity upper-bounds will be from the true posterior asymptotically. However, the sampling approach has a prohibitive computational cost and does not scale to large datasets. The comparison of efficiency between Gibbs sampler and variational inference is outside of the scope of this paper. Here we adopt the augmented model to derive an efficient mean-field approximation, which has closed-form iterative updates.

Following the standard derivation of mean-field approximation, we assume the posterior $p(\boldsymbol{\omega}^c, \boldsymbol{\omega}^p, \Pi, g, \bar{\lambda} \mid \mathbf{y}^r, \mathbf{y}^c, \mathbf{x}^p)$ is approximated by a variational posterior:

$$
q(\boldsymbol{\omega}^c, \boldsymbol{\omega}^p, \Pi, g, \bar{\lambda}) = q_1(\boldsymbol{\omega}^c, \boldsymbol{\omega}^p, \Pi) q_2(g, \bar{\lambda}).
$$

The independence of two sets of variables is the only assumption of the variational posterior. To minimize the Kullback–Leibler (KL) divergence between q and p, it can be proved that the optimal distribution of each factor is the expectation of the logarithm of Eq. (6) taken over variables in the other factor (Bishop, 2006; Blei et al., 2017):

$$
\begin{aligned}
q_1^*(\boldsymbol{\omega}^c, \boldsymbol{\omega}^p, \Pi) &\propto e^{\mathbb{E}_{q_2}[\log p(\mathbf{y}^r, \mathbf{y}^c, \mathbf{x}^p, \boldsymbol{\omega}^c, \boldsymbol{\omega}^p, \Pi, g, \bar{\lambda})]}, \\
q_2^*(g, \bar{\lambda}) &\propto e^{\mathbb{E}_{q_1}[\log p(\mathbf{y}^r, \mathbf{y}^c, \mathbf{x}^p, \boldsymbol{\omega}^c, \boldsymbol{\omega}^p, \Pi, g, \bar{\lambda})]}.
\end{aligned} \tag{7}
$$

A prominent weakness of GP is that it suffers from a cubic complexity w.r.t. the number of samples. In multi-task scenario, although the samples in a single task can be few, the total number of samples in all tasks can be large. To make our mean-field approximation scalable, we employ the inducing points formalism (Alvarez and Lawrence, 2008; Titsias, 2009). We denote M inducing inputs $[\mathbf{x}_1 \ \ldots, \mathbf{x}_M]^\top$ on the domain $\mathcal{X}$ for each task. The function values of basis function f_q at these inducing inputs are defined as $\mathbf{f}_{q,\mathbf{x}_m}$. Then we can obtain the i-th task latent function g_i at these inducing inputs $\mathbf{g}_{\mathbf{x}_m}^i = \sum_{q=1}^{Q} w_{i,q} \mathbf{f}_{q,\mathbf{x}_m}$.[3] If we define $\mathbf{g}_{\mathbf{x}_m} = [\mathbf{g}_{\mathbf{x}_m}^{1\top}, \ldots, \mathbf{g}_{\mathbf{x}_m}^{I\top}]^\top$, $\mathbf{g}_{\mathbf{x}_m} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{x}_m\mathbf{x}_m})$ where $\mathbf{K}_{\mathbf{x}_m\mathbf{x}_m}$ is the MOGP covariance on $\mathbf{x}_m$ for all tasks and $\mathbf{g}_{\mathbf{x}_m}^i \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{x}_m\mathbf{x}_m}^i)$ where $\mathbf{K}_{\mathbf{x}_m\mathbf{x}_m}^i$ is i-th diagonal block of $\mathbf{K}_{\mathbf{x}_m\mathbf{x}_m}$. Given $\mathbf{g}_{\mathbf{x}_m}^i$, we assume $p(g_i(\mathbf{x}) \mid \mathbf{g}_{\mathbf{x}_m}^i) = \mathcal{N}(\mathbf{k}_{\mathbf{x}_m\mathbf{x}}^{i\top} \mathbf{K}_{\mathbf{x}_m\mathbf{x}_m}^{i^{-1}} \mathbf{g}_{\mathbf{x}_m}^i, k_{\mathbf{xx}}^i - \mathbf{k}_{\mathbf{x}_m\mathbf{x}}^{i\top} \mathbf{K}_{\mathbf{x}_m\mathbf{x}_m}^{i^{-1}} \mathbf{k}_{\mathbf{x}_m\mathbf{x}}^i)$ where $\mathbf{k}_{\mathbf{x}_m\mathbf{x}}^i$ is the kernel w.r.t. inducing points and the predictive point, $k_{\mathbf{xx}}^i$ is the kernel w.r.t. the predictive point for i-th task.

[3] For the compactness of notation, the task index i is sometimes moved from subscript to superscript, which does not cause confusion because we use i consistently.

After substituting Eq. (6) into Eq. (7) and introducing the inducing points, we can obtain the optimal variational distribution of each factor in the following closed-form expressions (derivation provided in Section C).

4.4 The optimal density of Pólya-Gamma latent variables

The optimal variational posteriors of $\boldsymbol{\omega}^c$ and $\boldsymbol{\omega}^p$ are:

$$q_1(\boldsymbol{\omega}^c) = \prod_{i=1}^{I_c} \prod_{n=1}^{N_i^c} p_{\mathrm{PG}}(\omega_{i,n}^c \mid 1, \tilde{g}_{i,n}^c), \tag{8}$$

$$q_1(\boldsymbol{\omega}^p) = \prod_{i=1}^{I_p} \prod_{n=1}^{N_i^p} p_{\mathrm{PG}}(\omega_{i,n}^p \mid 1, \tilde{g}_{i,n}^p), \tag{9}$$

where $\tilde{g}_{i,n}^{\cdot} = \sqrt{\mathbb{E}[{g_{i,n}^{\cdot}}^2]}$.

4.5 The optimal intensity of marked Poisson processes

The optimal variational posterior intensity of $\Pi = \{\Pi_i\}_{i=1}^{I_p}$ is:

$$\Lambda_i^1(\mathbf{x}, \omega) = \bar{\lambda}_i^1 s(-\tilde{g}_i^p(\mathbf{x})) p_{\mathrm{PG}}(\omega \mid 1, \tilde{g}_i^p(\mathbf{x})) e^{(\tilde{g}_i^p(\mathbf{x}) - \bar{g}_i^p(\mathbf{x}))/2}, \tag{10}$$

where $\bar{\lambda}_i^1 = e^{\mathbb{E}[\log \bar{\lambda}_i]}$, $\tilde{g}_i^p(\mathbf{x}) = \sqrt{\mathbb{E}[g_i^p(\mathbf{x})^2]}$ and $\bar{g}_i^p(\mathbf{x}) = \mathbb{E}[g_i^p(\mathbf{x})]$.

4.6 The optimal density of intensity upper-bounds

The optimal variational posterior of $\bar{\lambda}$ is:

$$q_2(\bar{\lambda}) = \prod_{i=1}^{I_p} p_{\mathrm{Ga}}(\bar{\lambda}_i \mid N_i^p + R_i, |\mathcal{X}|), \tag{11}$$

where p_{Ga} is Gamma density, $R_i = \int_{\mathcal{X}} \int_0^\infty \Lambda_i^1(\mathbf{x}, \omega) d\omega d\mathbf{x}$, $|\mathcal{X}|$ is the domain size.

4.7 The optimal density of latent functions

The optimal variational posterior of $\mathbf{g}_{\mathbf{x}_m}$ is:

$$q_2(\mathbf{g}_{\mathbf{x}_m}) = \mathcal{N}(\mathbf{g}_{\mathbf{x}_m} \mid \mathbf{m}_{\mathbf{x}_m}, \boldsymbol{\Sigma}_{\mathbf{x}_m}), \tag{12}$$

where $\mathbf{g}_{\mathbf{x}_m} = [\mathbf{g}_{\mathbf{x}_m}^{r\top}, \mathbf{g}_{\mathbf{x}_m}^{c\top}, \mathbf{g}_{\mathbf{x}_m}^{p\top}]^\top$, $\mathbf{g}_{\mathbf{x}_m}^{\cdot} = [\mathbf{g}_{1,\mathbf{x}_m}^{\cdot\top}, \dots, \mathbf{g}_{I_\cdot,\mathbf{x}_m}^{\cdot\top}]^\top$ and

$$\boldsymbol{\Sigma}_{\mathbf{x}_m} = \left[\mathrm{diag}\left(\mathbf{H}_{\mathbf{x}_m}^r, \mathbf{H}_{\mathbf{x}_m}^c, \mathbf{H}_{\mathbf{x}_m}^p\right) + \mathbf{K}_{\mathbf{x}_m\mathbf{x}_m}^{-1}\right]^{-1}, \mathbf{m}_{\mathbf{x}_m} = \boldsymbol{\Sigma}_{\mathbf{x}_m}[\mathbf{v}_{\mathbf{x}_m}^{r\top}, \mathbf{v}_{\mathbf{x}_m}^{c\top}, \mathbf{v}_{\mathbf{x}_m}^{p\top}]^\top,$$

where $\mathbf{H}_{\mathbf{x}_m}^{\cdot} = \mathrm{diag}(\mathbf{H}_{1,\mathbf{x}_m}^{\cdot}, \dots, \mathbf{H}_{I_\cdot,\mathbf{x}_m}^{\cdot})$, $\mathbf{v}_{\mathbf{x}_m}^{\cdot} = [\mathbf{v}_{1,\mathbf{x}_m}^{\cdot\top}, \dots, \mathbf{v}_{I_\cdot,\mathbf{x}_m}^{\cdot\top}]^\top$ and

$$
\begin{aligned}
\mathbf{H}^{r}_{i,\mathbf{x}_m} &= \mathbf{K}^{r,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m}\mathbf{K}^{r,i}_{\mathbf{x}_m\mathbf{x}_n}\mathbf{D}^{r}_{i}\mathbf{K}^{r,i\top}_{\mathbf{x}_m\mathbf{x}_n}\mathbf{K}^{r,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m}, \mathbf{v}^{r}_{i,\mathbf{x}_m} = \mathbf{K}^{r,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m}\mathbf{K}^{r,i}_{\mathbf{x}_m\mathbf{x}_n}\frac{\mathbf{y}^{r}_{i}}{\sigma^{2}_{i}},\\
\mathbf{H}^{c}_{i,\mathbf{x}_m} &= \mathbf{K}^{c,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m}\mathbf{K}^{c,i}_{\mathbf{x}_m\mathbf{x}_n}\mathbf{D}^{c}_{i}\mathbf{K}^{c,i\top}_{\mathbf{x}_m\mathbf{x}_n}\mathbf{K}^{c,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m}, \mathbf{v}^{c}_{i,\mathbf{x}_m} = \mathbf{K}^{c,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m}\mathbf{K}^{c,i}_{\mathbf{x}_m\mathbf{x}_n}\frac{\mathbf{y}^{c}_{i}}{2},\\
\mathbf{H}^{p}_{i,\mathbf{x}_m} &= \mathbf{K}^{p,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m}\int_{\mathcal{X}} A_i(\mathbf{x})\mathbf{k}^{p,i}_{\mathbf{x}_m\mathbf{x}}\mathbf{k}^{p,i\top}_{\mathbf{x}_m\mathbf{x}}d\mathbf{x}\mathbf{K}^{p,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m},\\
\mathbf{v}^{p}_{i,\mathbf{x}_m} &= \mathbf{K}^{p,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m}\int_{\mathcal{X}} B_i(\mathbf{x})\mathbf{k}^{p,i}_{\mathbf{x}_m\mathbf{x}}d\mathbf{x},
\end{aligned}
$$

where $\mathbf{D}^{r}_{i} = \operatorname{diag}(1/\sigma^{2}_{i})$, $\mathbf{D}^{c}_{i} = \operatorname{diag}(\mathbb{E}[\boldsymbol{\omega}^{c}_{i}])$ and

$$
\begin{aligned}
A_i(\mathbf{x}) &= \sum_{n=1}^{N^{p}_{i}} \mathbb{E}[\omega^{p}_{i,n}]\delta(\mathbf{x}-\mathbf{x}^{p}_{i,n}) + \int_{0}^{\infty} \omega\Lambda^{1}_{i}(\mathbf{x},\omega)d\omega,\\
B_i(\mathbf{x}) &= \frac{1}{2}\sum_{n=1}^{N^{p}_{i}} \delta(\mathbf{x}-\mathbf{x}^{p}_{i,n}) - \frac{1}{2}\int_{0}^{\infty} \Lambda^{1}_{i}(\mathbf{x},\omega)d\omega.
\end{aligned}
$$

4.8 Predictive distribution

The posterior distribution of the task-specific latent function g_i at a predictive point $\mathbf{x}$ is approximated by

$$
q(g_i(\mathbf{x})) = \int p(g_i(\mathbf{x}) \mid \mathbf{g}^{i}_{\mathbf{x}_m})q(\mathbf{g}^{i}_{\mathbf{x}_m})d\mathbf{g}^{i}_{\mathbf{x}_m} = \mathcal{N}(g_i(\mathbf{x}) \mid \mu, \sigma^2),
$$

where $\mu = \mathbf{k}^{i\top}_{\mathbf{x}_m\mathbf{x}}\mathbf{K}^{i^{-1}}_{\mathbf{x}_m\mathbf{x}_m}\mathbf{m}^{i}_{\mathbf{x}_m}$, $\sigma^2 = k^{i}_{\mathbf{x}\mathbf{x}} - \mathbf{k}^{i\top}_{\mathbf{x}_m\mathbf{x}}\mathbf{K}^{i^{-1}}_{\mathbf{x}_m\mathbf{x}_m}\mathbf{k}^{i}_{\mathbf{x}_m\mathbf{x}} + \mathbf{k}^{i\top}_{\mathbf{x}_m\mathbf{x}}\mathbf{K}^{i^{-1}}_{\mathbf{x}_m\mathbf{x}_m}\boldsymbol{\Sigma}^{i}_{\mathbf{x}_m}\mathbf{K}^{i^{-1}}_{\mathbf{x}_m\mathbf{x}_m}\mathbf{k}^{i}_{\mathbf{x}_m\mathbf{x}}$. Therefore, $\tilde{g}_i(\mathbf{x}) = \sqrt{\mu^2+\sigma^2}$, $\bar{g}_i(\mathbf{x}) = \mu$, $\mathbb{E}[\omega] = \frac{b}{2c}\tanh\frac{c}{2}$ for $p_{\mathrm{PG}}(\omega \mid b, c)$ (Polson et al., 2013), $\mathbb{E}[\log\bar{\lambda}_i] = \psi(N^{p}_{i} + R_i) - \log(|\mathcal{X}|)$ where $\psi(\cdot)$ is digamma function. The intractable integral over $\mathcal{X}$ is solved by numerical quadrature. Updating the variational posterior of each factor alternately by Eqs. (8), (9), (10), (11) and (12), we obtain approximate posteriors of $\bar{\lambda}$ and $\mathbf{g}_{\mathbf{x}_m}$.

4.9 Hyperparameters and computation complexity

The model hyperparameter $\boldsymbol{\Theta}$ comprises the kernel hyperparameters $\{\boldsymbol{\theta}_q\}_{q=1}^{Q}$ associated to the covariance functions $\{k_q\}_{q=1}^{Q}$, the mixing weights $\{\mathbf{w}_q\}_{q=1}^{Q}$, the inducing inputs $\{\mathbf{x}_m\}_{m=1}^{M}$ and the noise variance $\{\sigma^2_i\}_{i=1}^{I_r}$ in regression tasks. In this work, the inducing points are uniformly located on the domain, which means the kernel matrix has Toeplitz structure (Cunningham et al., 2008) and this can lead to more efficient matrix inversion. In the implementation, we do not apply this method and instead use the naive matrix inversion. $\{\boldsymbol{\theta}_q\}_{q=1}^{Q}$, $\{\mathbf{w}_q\}_{q=1}^{Q}$ and $\{\sigma^2_i\}_{i=1}^{I_r}$ are optimized by maximizing the marginal likelihood, which is also called the empirical Bayes. Due to the intractability of marginal likelihood, we adopt an approximate approach: maximize the ELBO as a function of hyperparameters by alternating between updating variational parameters and hyperparameters. In the following, we derive the ELBO:

$$
\begin{aligned}
&\log p(\mathbf{y}^r, \mathbf{y}^c, \mathbf{x}^p) \geq \\
&\mathbb{E}_q[\log p(\mathbf{y}^r, \mathbf{y}^c, \mathbf{x}^p \mid \boldsymbol{\omega}^c, \boldsymbol{\omega}^p, \Pi, g, \bar{\lambda})] - \mathrm{KL}(q(\boldsymbol{\omega}^c, \boldsymbol{\omega}^p, \Pi, g, \bar{\lambda}) \| p(\boldsymbol{\omega}^c, \boldsymbol{\omega}^p, \Pi, g, \bar{\lambda})) \\
&= \mathbb{E}_q[\log p(\mathbf{y}^r \mid \{g_i^r\}_{i=1}^{I_r})] + \mathbb{E}_q[\log p(\mathbf{y}^c \mid \boldsymbol{\omega}^c, \{g_i^c\}_{i=1}^{I_c})] \\
&\quad + \mathbb{E}_q[\log p(\mathbf{x}^p \mid \boldsymbol{\omega}^p, \Pi, \{g_i^p\}_{i=1}^{I_p}, \bar{\lambda})] - \mathrm{KL}(q(g) \| p(g)) \\
&\quad - \mathrm{KL}(q(\boldsymbol{\omega}^c, \boldsymbol{\omega}^p, \Pi, \bar{\lambda}) \| p(\boldsymbol{\omega}^c, \boldsymbol{\omega}^p, \Pi, \bar{\lambda})),
\end{aligned}
$$

where we omit the conditioning on hyperparameters. It is straightforward to see that, given variational posteriors, only the first term includes the noise variance $\{\sigma_i^2\}_{i=1}^{I_r}$ and only the fourth term includes the kernel hyperparameters $\{\boldsymbol{\theta}_q\}_{q=1}^{Q}$ and the mixing weights $\{\mathbf{w}_q\}_{q=1}^{Q}$. All other terms are constant w.r.t. hyperparameters. After introducing the inducing points on g, we obtain the inducing points version:

$$
\mathbb{E}_q[\log p(\mathbf{y}^r \mid \{g_i^r\}_{i=1}^{I_r})] = \sum_{i=1}^{I_r} \sum_{n=1}^{N_i^r} -\log(\sigma_i \sqrt{2\pi}) - \frac{1}{2\sigma_i^2}(y_{i,n}^{r2} - 2y_{i,n}^r \bar{g}_{i,n}^r + \tilde{g}_{i,n}^{r2}) \tag{13a}
$$

$$
\begin{aligned}
&\mathrm{KL}(q(\mathbf{g}_{\mathbf{x}_m}) \| p(\mathbf{g}_{\mathbf{x}_m})) \\
&\quad = \frac{1}{2}\Big(\log |\mathbf{K}_{\mathbf{x}_m \mathbf{x}_m}| - \log |\boldsymbol{\Sigma}_{\mathbf{x}_m}| - M \cdot I + \mathrm{Tr}[\mathbf{K}_{\mathbf{x}_m \mathbf{x}_m}^{-1} \boldsymbol{\Sigma}_{\mathbf{x}_m}] + \mathbf{m}_{\mathbf{x}_m}^{\top} \mathbf{K}_{\mathbf{x}_m \mathbf{x}_m}^{-1} \mathbf{m}_{\mathbf{x}_m}\Big),
\end{aligned} \tag{13b}
$$

where we assume $p(\mathbf{g}_{\mathbf{x}_m}) = \mathcal{N}(\mathbf{g}_{\mathbf{x}_m} \mid \mathbf{0}, \mathbf{K}_{\mathbf{x}_m \mathbf{x}_m})$. Maximizing Eq. (13a), we obtain the optimal noise variance:

$$
\sigma_i^{2*} = \left(\sum_{n=1}^{N_i^r} y_{i,n}^{r2} - 2y_{i,n}^r \bar{g}_{i,n}^r + \tilde{g}_{i,n}^{r2} \right) / N_i^r. \tag{14}
$$

Minimizing Eq. 13b, we obtain the optimal kernel hyperparameters $\{\boldsymbol{\theta}_q\}_{q=1}^{Q}$ and mixing weights $\{\mathbf{w}_q\}_{q=1}^{Q}$, which has no closed-form solution and we resort to the automatic differentiation technique. The pseudocode of mean-field approximation is provided in Algorithm 1.

Algorithm 1 Mean-Field Approximation

```
Initialize hyperparameters and variational parameters.
repeat
    Update the optimal variational distribution of Pólya-Gamma variables for classification tasks in Eq. (8);
    Update the optimal variational distribution of Pólya-Gamma variables for Cox process tasks in Eq. (9);
    Update the optimal variational intensity of marked Poisson processes for Cox process tasks in Eq. (10);
    Update the optimal variational distribution of intensity upper-bounds for Cox process tasks in Eq. (11);
    Update the optimal variational distribution of latent functions for all tasks in Eq. (12);
    Update the hyperparameters $\{\boldsymbol{\theta}_q, \mathbf{w}_q\}_{q=1}^{Q}$ by minimizing Eq. (13b);
    Update the hyperparameter $\sigma^2$ by Eq. (14).
until convergence
return $g_i^r(\mathbf{x})$ for regression task, $s(g_i^c(\mathbf{x}))$ for classification task and $\bar{\lambda}_i s(g_i^p(\mathbf{x}))$ for Cox process task.
```

Defining S as the number of quadrature nodes on all point process tasks, the computational complexity of our mean-field approximation is dominated by the matrix inversion $O(M^3 I^3)$ and product $O(M^2(N^r + N^c + N^p + S))$ where $N^{\cdot}$ is the number of samples in the corresponding tasks.

4.10 Convergence and minibatch

The theoretical analysis in Hoffman et al. (2013) shows that performing the mean-field iteration for a conditionally conjugate model is equivalent to updating parameters by the natural gradient descent (Amari, 1998) with a step size of one. Therefore, our proposed mean-field approximation has inherently a faster convergence than the standard gradient descent.

The mean-field algorithm above uses all data. For further acceleration, we can resort to the stochastic variational inference (Hoffman et al., 2013) by subsampling the tasks, and samples in regression and classification tasks.

5 Experiments

In this section, we analyze our model and inference on synthetic and real-world datasets to demonstrate the performance in terms of transfer capability, efficiency and convergence. For all experiments, we use the RBF kernel $k(\mathbf{x}, \mathbf{x}') = \theta_0 \exp(-\frac{\theta_1}{2}\|\mathbf{x} - \mathbf{x}'\|^2)$ as covariance functions, and the usage of other kernels is outside of the scope of this paper. The implementation code is publicly available at https://github.com/zhoufeng6288/HGCox.

Baselines To show the superiority of our approach, we compare our model HMGCP against the single-task Cox process model: variational LGCP (Nguyen & Bonilla, 2014), and the multi-task models: MLGCP (Taylor et al., 2015) and MCPM (Aglietti et al., 2019).

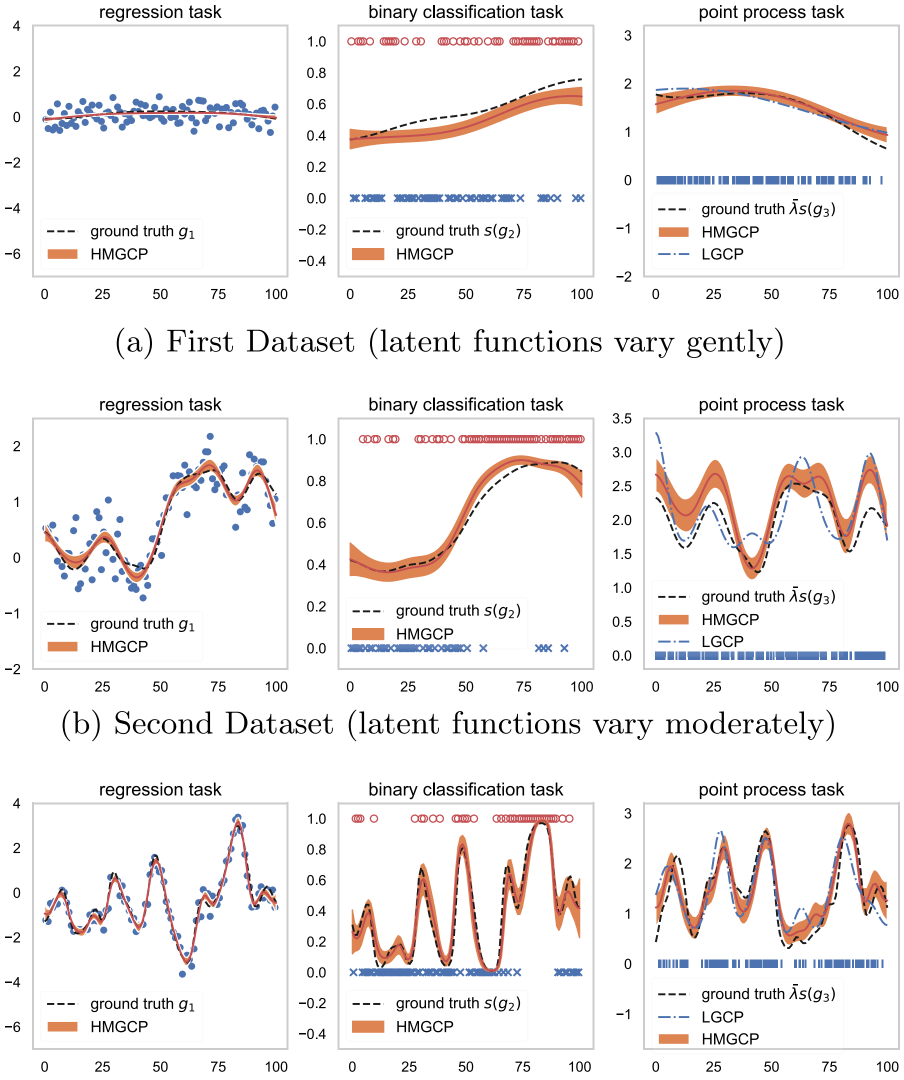
Metrics We provide the comparison result of our model with baselines in terms of *estimation error (EE)*, *test log-likelihood (TLL)*, *running time (RT)* and *convergence rate (CR)*. EE is the root mean square error (RMSE) between the estimated parameter and the ground truth. It is worth noting that EE is only applicable to synthetic data because the ground truth is required. TLL is the log-likelihood on test data using the posterior mean of parameters estimated from training data. RT is the running time of the inference algorithm. CR is the convergence rate of training log-likelihood w.r.t. the number of iterations.

5.1 Synthetic data: complete

To illustrate the performance of transfer capability, efficiency and convergence of our approach, we simulate three heterogeneous correlated tasks (one regression, one binary classification and one Cox process) by sampling three latent functions from a MOGP prior and using them to simulate the observed samples in regression, classification and Cox process tasks. We simulate three sets of synthetic data using three different sets of hyperparameters where latent functions vary from gently to drastically; each synthetic dataset contains both training and test data. We use two basis functions. The hyperparameters are $\sigma^2 = 0.1$, $\boldsymbol{\theta}_1 = [1, 0.001]$, $\boldsymbol{\theta}_2 = [1, 0.001]$, $\mathbf{w}_1 = [0.9, 0.5, 0.1]$ and $\mathbf{w}_2 = [0.1, 0.5, 0.9]$ for the first dataset; $\sigma^2 = 0.1$, $\boldsymbol{\theta}_1 = [1, 0.02]$, $\boldsymbol{\theta}_2 = [2, 0.001]$, $\mathbf{w}_1 = [0.9, 0.5, 0.1]$ and $\mathbf{w}_2 = [0.1, 0.5, 0.9]$ for the second dataset; $\sigma^2 = 0.1$, $\boldsymbol{\theta}_1 = [1, 0.1]$, $\boldsymbol{\theta}_2 = [2, 0.1]$, $\mathbf{w}_1 = [0.9, 0.5, 0.1]$ and $\mathbf{w}_2 = [0.1, 0.5, 0.9]$ for the third dataset.

For each dataset, we draw two basis functions $\{f_q\}_{q=1}^2$ on the domain [0, 100] from two independent zero-mean GP priors with the corresponding kernel hyperparameters. The task-specific latent functions are $\{g_i = \sum_{q=1}^2 w_{i,q} f_q\}_{i=1}^3$. g_1 is used as the mean of a Gaussian distribution $\mathcal{N}(g_1(\mathbf{x}), \sigma^2)$ to draw samples for the regression task. g_2 is passed through a sigmoid function and then used as the parameter of a Bernoulli distribution to draw samples for the binary classification task. g_3 is passed through a sigmoid function and then scaled by $\bar{\lambda}$ to serve as the intensity for simulating a Cox process. For regression and classification tasks, we assume the samples are uniformly distributed on the domain.

Our goal is to recover the intensity upper-bound $\bar{\lambda}$ and latent functions $\{g_i\}_{i=1}^3$. We use 30 inducing points that are uniformly distributed on the domain and 100 Gaussian quadrature nodes for the intractable integral. For initialization, the initial hyperparameters σ^2, $\{\boldsymbol{\theta}_q, \mathbf{w}_q\}_{q=1}^2$ are set to the ground-truth hyperparameters and the variational parameters are initialized randomly. In the training process, the variational parameters and hyperparameters are updated concurrently. Specifically, the variational parameters are updated by the mean-filed iteration, the kernel hyperparameters $\{\boldsymbol{\theta}_q, \mathbf{w}_q\}_{q=1}^2$ are updated by minimizing Eq. (13b) using the 'SLSQP' method, and the noise variance σ^2 is updated by Eq. (14). Figure 1 represents the estimated result for three datasets where we can see HMGCP is able to recover the ground truth. For convergence, HMGCP only takes 2-3 steps to converge in terms of training log-likelihood, which is much faster than the first-order gradient-based LGCP requiring more than 500 steps. More importantly, HMGCP has the better EE and TLL (Table 1) than the single-task LGCP that is trained independently and not able to transfer information to help recover the intensity of Cox process. For a fair comparison of



(a) First Dataset (latent functions vary gently)

(b) Second Dataset (latent functions vary moderately)

(c) Third Dataset (latent functions vary drastically)

Fig. 1 HMGCP recovers the latent functions g_1, $s(g_2)$ and $\bar{\lambda}s(g_3)$ in three datasets whose posterior is constructed by 100 samples of g and $\bar{\lambda}$ from the corresponding variational posterior. The shading area indicates one standard deviation. Blue dots are samples in regression task; red circles and blue crosses are positive and negative samples in classification task; blue bars are samples in Cox process task. For LGCP, we show the posterior mean intensity for the Cox process task

efficiency, we run both HMGCP and LGCP on a single Cox process task with 400 iterations, and our inference is at least twice as fast as LGCP (Table 1) demonstrating its outstanding efficiency.

Table 1 The performance of EE, TLL and RT for HMGCP and LGCP on three synthetic datasets.

	Model	EE(reg)	EE(cla)	EE(Cox)	TLL(reg)	TLL(cla)	TLL(Cox)	RT
1	HMGCP	0.046	0.074	**0.114**	−33.17	−63.57	**−89.05**	**0.73**
	LGCP	×	×	0.147	×	×	−90.23	2.70
2	HMGCP	0.098	0.048	**0.319**	−28.54	−55.23	**−63.54**	**1.09**
	LGCP	×	×	0.385	×	×	−65.19	2.73
3	HMGCP	0.167	0.067	**0.272**	−42.43	−56.14	**−72.75**	**0.69**
	LGCP	×	×	0.433	×	×	−79.17	2.71

EE is the RMSE between posterior mean and ground truth. Time in seconds

5.2 Synthetic data: missing

As far as we know, all current multi-task Cox process models exclusively focus on *homogeneous* scenarios. This does not apply to the more general *heterogeneous* multi-task setup where we need to transfer knowledge between multiple heterogeneous correlated tasks. In this section, we compare HMGCP against homogeneous multi-task baselines: MLGCP and MCPM. We construct four heterogeneous correlated tasks (one regression, one binary classification and two Cox processes) using the same method as in Sect. 5.1. We simulate one set of synthetic data that contains both training and test data. We use two basis functions. The hyperparameters are $\sigma^2 = 0.1$, $\boldsymbol{\theta}_1 = [1, 0.02]$, $\boldsymbol{\theta}_2 = [2, 0.001]$, $\mathbf{w}_1 = [0.9, 0.1, 0.3, 1.0]$ and $\mathbf{w}_2 = [0.1, 0.9, 0.5, 1.0]$. To further illustrate the heterogeneous transfer capability of our approach, in addition to the complete data, we follow the experimental setup of Aglietti et al. (2019): we create some *missing gaps* by evenly partitioning the domain into several regions and randomly masking four non-overlapping regions on four tasks (one for each task). To demonstrate the transfer capability on problems with different levels of difficulty, we experiment with two missing-gap widths: 5 and 10, where a wider missing gap means a more difficult transfer problem. For each missing-gap width, we experiment with ten random configurations of missing gaps.

We use 10 inducing points which are uniformly distributed on the domain. All the other experimental settings are the same as in Sect. 5.1. HMGCP successfully transfers knowledge between heterogeneous tasks by exploiting commonalities between them to recover the missing-gap latent functions for all tasks (Fig. 2), whereas MLGCP and MCPM exhibit the inferior generalization capability since they can only share information between Cox processes. Figure 2 shows the estimated latent functions for several configurations with 3 different missing-gap widths across tasks. Generally, the transfer of knowledge in regression and classification tasks is easier than that in Cox process tasks. This is because the likelihood of regression and classification only considers observed points, the function in the missing gap is entirely determined by the smoothness induced by prior. However, in addition to observed points, the Cox process likelihood also considers the domain where no points appear, so the function in the missing gap is determined by both prior and likelihood (zero-valued intensity). This makes the estimated intensity in the missing gap generally lower than the ground truth. For each missing-gap width, we report the statistics of EE and TLL for HMGCP, MLGCP and MCPM over ten random configurations of missing gaps in Table 2 where HMGCP outperforms alternatives in all experiments. The reason is HMGCP extracts useful information from regression, classification and other Cox processes to improve the estimation of intensity for the current Cox process, while MLGCP and MCPM cannot incorporate the information existing in heterogeneous tasks. As in Sect. 5.1, we run HMGCP, MLGCP and MCPM only on the complete Cox process data for a fair comparison

Table 2 The performance of EE and TLL for HMGCP, MLGCP and MCPM over ten random configurations of missing gaps with three different missing-gap widths (0 means complete data).

Gap Width	Model	EE(reg)	EE(cla)	EE(Cox)	TLL(reg)	TLL(cla)	TLL(Cox)
0	HMGCP	0.093	0.066	**0.390**	−50.61	−56.67	**−120.55**
	MLGCP	×	×	0.535	×	×	−136.28
	MCPM	×	×	0.676	×	×	−126.73
5	HMGCP	0.095(0.006)	0.066(0.005)	**0.461**(0.056)	−50.76(0.92)	−56.74(0.51)	**−122.94**(2.27)
	MLGCP	×	×	0.601(0.051)	×	×	−126.24(3.39)
	MCPM	×	×	0.725(0.035)	×	×	−129.82(2.53)
10	HMGCP	0.111(0.006)	0.072(0.008)	**0.664**(0.071)	−52.14(1.94)	−56.82(0.69)	**−128.49**(5.74)
	MLGCP	×	×	0.791(0.070)	×	×	−128.59(5.33)
	MCPM	×	×	0.765(0.024)	×	×	−131.59(1.99)

The mean and standard deviation (in brackets) are provided. EE(Cox)/TLL(Cox) is the sum of EEs/TLLs of two Cox processes

of efficiency: HMGCP consumes 3.68 seconds, while MLGCP and MCPM consume 12.15 and 21.36 seconds, respectively (2000 iterations).

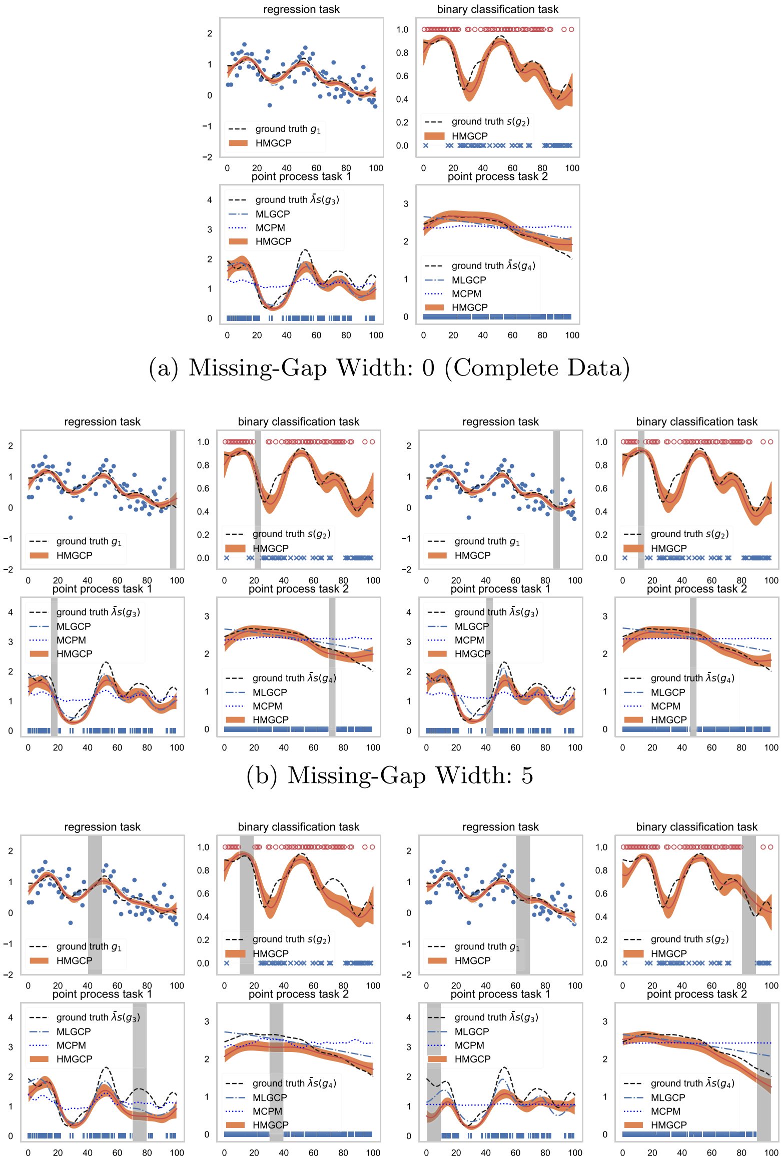
5.3 Real data

In this section, we demonstrate the superiority of HMGCP in terms of heterogeneous knowledge transfer, efficiency and convergence on a real-world 2D urban data of Vancouver. The dataset[4] contains four parts of data (Fig. 3): **(1)** *Employment income in Vancouver*: the median employment income for full-year full-time workers in 2015 in the neighbourhoods of Vancouver; **(2)** *Education in Vancouver*: the number of population holding university certificate, diploma or degree at bachelor level or above in the neighbourhoods of Vancouver; **(3)** *Crime in Vancouver*: the recording of miscellaneous crimes (type, neighbourhood, latitude, longitude) in 2015 in Vancouver; **(4)** *Non-market housing in Vancouver*: the information of non-market housing projects (name, address, neighbourhood, latitude, longitude) that is for low and moderate income singles and families.

For the first dataset, we formulate it as a regression task, and use the centroid of each neighbourhood as the input, the median income as the output; for the second dataset, we formulate it as a binary classification task according to the degree of education: we divide the 22 neighbourhoods into '+1' if there are more people holding university certificate, diploma or degree at bachelor level or above, and '−1' if not; for the third and fourth datasets, we extract the locations of 'Theft of Vehicle' records in 2015 and non-market housing projects respectively, and formulate them as two Cox process tasks. On the basis of common sense, the income level, education degree, crime rate and non-market housing are closely correlated. Therefore, their integrative analysis offers more advantages compared to learning multiple tasks independently, which is susceptible to overfitting.

To show the heterogeneous transfer capability of our approach, we compare HMGCP against MLGCP and MCPM. Due to lack of ground-truth latent functions, we cannot compare them in terms of EE but only TLL. We scale the area of Vancouver between longitude

[4] The income, education and non-market housing data is from the Vancouver Open Data Catalog (https://opendata.vancouver.ca/pages/home/). The crime data is from Kaggle (https://www.kaggle.com/datasets/wosaku/crime-in-vancouver).



(a) Missing-Gap Width: 0 (Complete Data)

(b) Missing-Gap Width: 5

(c) Missing-Gap Width: 10

Fig. 2 The estimated posterior of latent functions g_1, $s(g_2)$, $\bar{\lambda}_3 s(g_3)$ and $\bar{\lambda}_4 s(g_4)$ from HMGCP with missing-gap width being (**a**) 0, **b** 5 and **c** 10. For missing-gap widths 5 and 10, we show two configurations of missing gaps across tasks. The grey areas indicate the masked missing gaps. For MLGCP and MCPM, we show the posterior mean intensities for two Cox process tasks. The posterior variance in the missing gap does not increase significantly meaning HMGCP successfully transfers heterogeneous knowledge

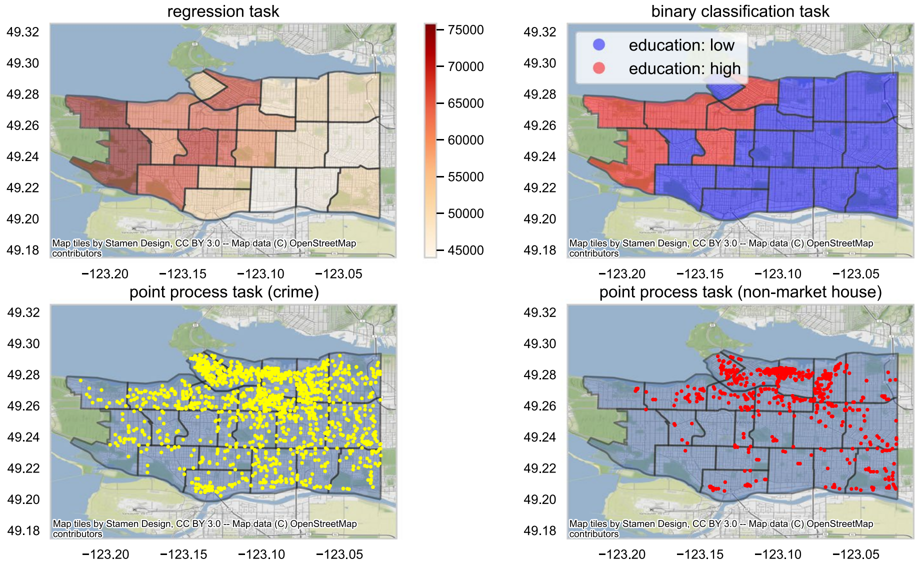
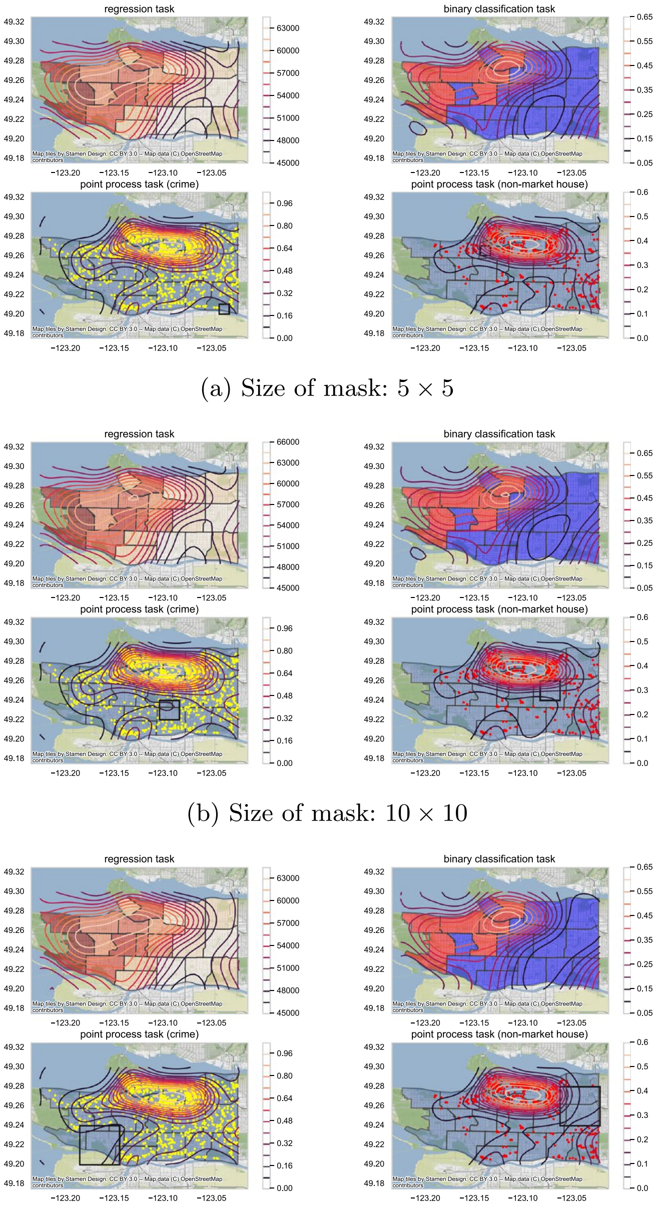

regression task
binary classification task
education: low
education: high
point process task (crime)
point process task (non-market house)
49.32
49.30
49.28
49.26
49.24
49.22
49.20
49.18
−123.20
−123.15
−123.10
−123.05
75000
70000
65000
60000
55000
50000
45000
Map tiles by Stamen Design, CC BY 3.0 -- Map data (C) OpenStreetMap contributors

Fig. 3 The median employment income (top left), education degree (top right), theft of vehicle (bottom left) and non-market house (bottom right) in 22 neighbourhoods of Vancouver

$[-123.226, -123.022]$ and latitude $[49.20, 49.30]$ to the domain $[0, 100] \times [0, 50]$. We choose three basis functions by trial and error: we gradually increase the number of basis functions and find that using three basis functions can achieve excellent performance. Using more basis functions only has a slight impact on the performance on the test data, but leads to longer training time. The initial hyperparameters are set to $\sigma^2 = 0.1$, $\boldsymbol{\theta}_1 = [1, 0.01]$, $\boldsymbol{\theta}_2 = [1, 0.005]$, $\boldsymbol{\theta}_3 = [1, 0.001]$, $\mathbf{w}_1 = [0.5, 0.5, 0.1, 0.1]$, $\mathbf{w}_2 = [0.1, 0.5, 0.2, 0.5]$ and $\mathbf{w}_3 = [0.5, 0.1, 0.5, 0.2]$, and the variational parameters are initialized randomly. In the training process, the variational parameters and hyperparameters are updated concurrently. Specifically, the variational parameters are updated by the mean-filed iteration, the kernel hyperparameters $\{\boldsymbol{\theta}_q, \mathbf{w}_q\}_{q=1}^{3}$ are updated by minimizing Eq. (13b) using the ‘SLSQP’ method, and the noise variance σ^2 is updated by Eq. (14). To assess the transfer capability with different levels of difficulty, we follow the experimental setup in Sect. 5.2: we randomly mask two non-overlapping regions on *Crime in Vancouver* and *Non-market housing in Vancouver*, one for each task, with three different mask sizes: $5 \times 5, 10 \times 10$ and 20×20. A larger mask indicates a more difficult transfer problem. For each mask size, we experiment with ten random configurations of masks.

We use 10×5 uniformly distributed inducing points horizontally and vertically on each task and 50×25 Gaussian quadrature nodes for the intractable integral. We randomly mask regions as explained above, and use the remaining data for training and the masked data for testing. Figure 4 shows several examples of estimated latent functions from HMGCP with 3 different mask sizes (two examples for each size), while Fig. 5 shows the corresponding estimated intensity functions from MLGCP and MCPM. The black boxes in Fig. 4 represent several possible configuration of masked regions on two Cox process tasks. It is easily observed in the data that in terms of income level and education degree, the west is significantly higher than the east; while for crime rate and non-market housing, it is the other way around. HMGCP successfully transfers knowledge existing in regression and



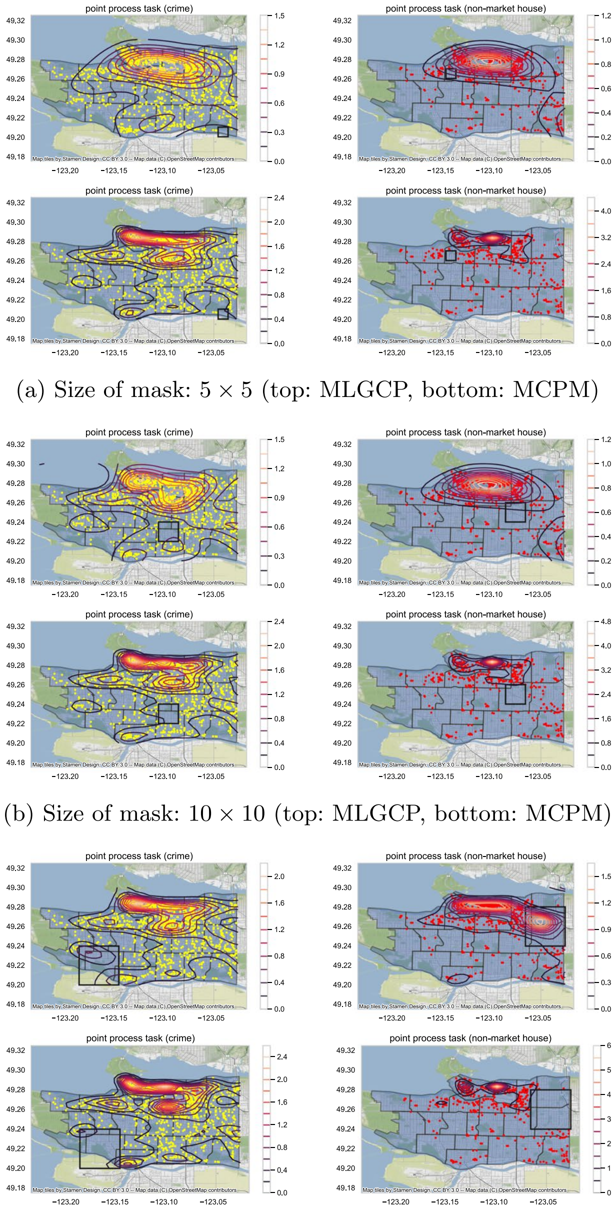
(a) Size of mask: 5 × 5

(b) Size of mask: 10 × 10

(c) Size of mask: 20 × 20

Fig. 4 The estimated posterior mean latent functions g_1, $s(g_2)$, $\bar{\lambda}_3 s(g_3)$ and $\bar{\lambda}_4 s(g_4)$ from HMGCP with the mask size being (a) 5 × 5, (b) 10 × 10 and (c) 20 × 20 on each Cox process task. We show one configuration of masked regions across Cox process tasks. The black boxes indicate the masked regions

classification tasks to help recover the intensity functions in masked regions for Cox process tasks (Fig. 4), while MLGCP and MCPM are prone to overfitting because they can only transfer homogeneous knowledge (Fig. 5). Therefore, HMGCP defeats the competing



(a) Size of mask: 5 × 5 (top: MLGCP, bottom: MCPM)

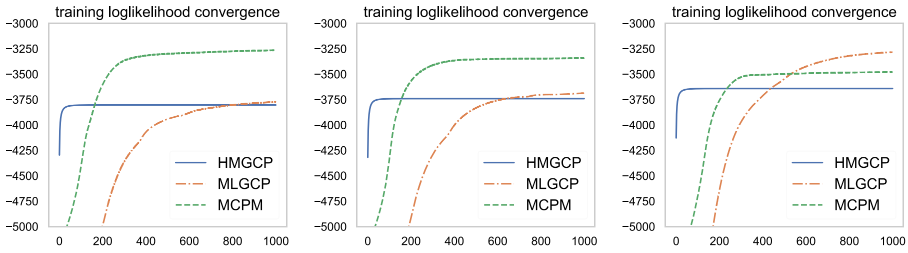
(b) Size of mask: 10 × 10 (top: MLGCP, bottom: MCPM)

(c) Size of mask: 20 × 20 (top: MLGCP, bottom: MCPM)

Fig. 5 The estimated posterior mean intensity functions for two Cox process tasks from MLGCP and MCPM with the mask size being (a) 5×5, (b) 10×10 and (c) 20×20 on each Cox process task. We show one configuration of masked regions across Cox process tasks. The black boxes indicate the masked regions

Table 3 The performance of TLL and RT for HMGCP, MLGCP and MCPM on the real data over ten random configurations of masked regions with three different sizes of mask. The mean and standard deviation (in brackets) are provided. Time in seconds

Size of Mask	Model	TLL (crime)	TLL (non-market house)	RT (per step)
5 × 5	HMGCP	**−14.20**(12.14)	**−14.22**(9.71)	**2.70**
	MLGCP	−22.71(22.67)	−23.67(21.67)	7.82
	MCPM	−24.40(23.32)	−20.08(13.93)	12.02
10 × 10	HMGCP	**−66.58**(28.91)	**−33.55**(22.90)	**2.67**
	MLGCP	−111.55(70.76)	−48.54(16.59)	7.39
	MCPM	−115.18(64.25)	−47.76(14.12)	11.81
20 × 20	HMGCP	**−313.75**(133.26)	**−143.11**(82.89)	**2.49**
	MLGCP	−776.84(425.69)	−363.07(305.55)	6.13
	MCPM	−558.02(205.73)	−223.67(101.72)	11.85



(a) Size of mask: 5 × 5 (b) Size of mask: 10 × 10 (c) Size of mask: 20 × 20

Fig. 6 The training log-likelihood convergence of HMGCP, MLGCP and MCPM. HMGCP only takes 40–50 steps to converge, while MCPM and MLGCP require more than 400 and 1000 steps to converge respectively. MLGCP and MCPM achieve the higher training log-likelihood due to overfitting

baselines MLGCP and MCPM in terms of TLL in all experiments (Table 3). More importantly, HMGCP has a faster convergence, which needs 40-50 steps to converge in terms of training log-likelihood, than the first-order gradient-based MLGCP and MCPM requiring more than 400 and 1000 steps respectively (Fig. 6). Besides, HMGCP significantly outperforms MLGCP and MCPM in terms of efficiency (Table 3, only on two Cox process tasks for a fair comparison).

6 Conclusion

The main objective of this study is to provide a heterogeneous multi-task learning framework for the analysis of multivariate inhomogeneous Poisson processes data with correlated regression and classification tasks. We adopt the MOGP prior to provide a shared representation to allow the transfer of knowledge between heterogeneous tasks. To circumvent the non-conjugate Bayesian inference, we employ the data augmentation technique to derive a closed-form mean-field approximation. Experimental results on synthetic and real data demonstrate that our model successfully shares the heterogeneous information to enhance the generalization capability and our inference approach has the predominant efficiency and convergence.

We adopted the LMC based MOGP to incorporate the correlation between multiple heterogeneous tasks. An interesting research track in the future may be the extension to MOGP based on process convolution, which may bring more benefits on computation efficiency. Moreover, we only consider three kinds of heterogeneous tasks: regression, classification and Cox process in this work; other kinds of unsupervised tasks, such as clustering, can also be attempted to be introduced to the multi-task framework.

Appendix A. Proof of augmented likelihood for classification

Substituting Eq. (2) in the paper into the classification likelihood Eq. (1b) in the paper, we can obtain

$$p(\mathbf{y}^c \mid \{g_i^c\}_{i=1}^{I_c}) = \prod_{i=1}^{I_c} \prod_{n=1}^{N_i^c} \int_0^\infty e^{h(\omega_{i,n}^c, y_{i,n}^c g_{i,n}^c)} p_{\mathrm{PG}}(\omega_{i,n}^c \mid 1, 0) d\omega_{i,n}^c, \tag{A1}$$

where the integrand is the augmented likelihood:

$$p(\mathbf{y}^c, \boldsymbol{\omega}^c \mid \{g_i^c\}_{i=1}^{I_c}) = \prod_{i=1}^{I_c} \prod_{n=1}^{N_i^c} e^{h(\omega_{i,n}^c, y_{i,n}^c g_{i,n}^c)} p_{\mathrm{PG}}(\omega_{i,n}^c \mid 1, 0). \tag{A2}$$

Appendix B. Proof of augmented likelihood for Cox process

Substituting Eqs. (2) and (4) in the paper into the product and exponential integral terms respectively in the Cox process likelihood Eq. (1c) in the paper, we can obtain

$$\begin{aligned} p(\mathbf{x}^p \mid \{\bar{\lambda}_i, g_i^p\}_{i=1}^{I_p}) = \prod_{i=1}^{I_p} \prod_{n=1}^{N_i^p} \int_0^\infty \Lambda_i(\mathbf{x}_{i,n}^p, \omega_{i,n}^p) e^{h(\omega_{i,n}^p, g_{i,n}^p)} d\omega_{i,n}^p \\ \int_{\mathcal{X}} \int_0^\infty p_{\Lambda_i}(\Pi_i \mid \bar{\lambda}_i) \prod_{(\omega, \mathbf{x}) \in \Pi_i} e^{h(\omega, -g_i^p(\mathbf{x}))} d\omega d\mathbf{x}, \end{aligned} \tag{B3}$$

where the integrand is the augmented likelihood:

$$\begin{aligned} p(\mathbf{x}^p, \boldsymbol{\omega}^p, \Pi \mid \bar{\lambda}, \{g_i^p\}_{i=1}^{I_p}) = \prod_{i=1}^{I_p} \prod_{n=1}^{N_i^p} \Lambda_i(\mathbf{x}_{i,n}^p, \omega_{i,n}^p) e^{h(\omega_{i,n}^p, g_{i,n}^p)} p_{\Lambda_i}(\Pi_i \mid \bar{\lambda}_i) \\ \prod_{(\omega, \mathbf{x}) \in \Pi_i} e^{h(\omega, -g_i^p(\mathbf{x}))}. \end{aligned} \tag{B4}$$

Appendix C. Proof of mean-field approximation

The augmented joint distribution can be written as:

$$
\begin{aligned}
&p(\mathbf{y}^r, \mathbf{y}^c, \mathbf{x}^p, \boldsymbol{\omega}^c, \boldsymbol{\omega}^p, \Pi, g, \bar{\lambda}) \\
=&\underbrace{p(\mathbf{y}^r \mid \{g_i^r\}_{i=1}^{I_r})}_{\text{regression}} \underbrace{p(\mathbf{y}^c, \boldsymbol{\omega}^c \mid \{g_i^c\}_{i=1}^{I_c})}_{\text{augmented classification}} \underbrace{p(\mathbf{x}^p, \boldsymbol{\omega}^p, \Pi \mid \bar{\lambda}, \{g_i^p\}_{i=1}^{I_p})}_{\text{augmented Cox process}} \underbrace{p(g)}_{\text{MOGP}} p(\bar{\lambda}) \\
=&\prod_{i=1}^{I_r} \prod_{n=1}^{N_i^r} \mathcal{N}(y_{i,n}^r \mid g_{i,n}^r, \sigma_i^2) \prod_{i=1}^{I_c} \prod_{n=1}^{N_i^c} e^{h(\omega_{i,n}^c, y_{i,n}^c g_{i,n}^c)} p_{\mathrm{PG}}(\omega_{i,n}^c \mid 1, 0) \\
&\prod_{i=1}^{I_p} \prod_{n=1}^{N_i^p} \Lambda_i(\mathbf{x}_{i,n}^p, \omega_{i,n}^p) e^{h(\omega_{i,n}^p, g_{i,n}^p)} p_{\Lambda_i}(\Pi_i \mid \bar{\lambda}_i) \prod_{(\omega,\mathbf{x}) \in \Pi_i} e^{h(\omega, -g_i^p(\mathbf{x}))} p(g) p(\bar{\lambda}).
\end{aligned}
\tag{C5}
$$

Here, we assume the variational posterior $q(\boldsymbol{\omega}^c, \boldsymbol{\omega}^p, \Pi, g, \bar{\lambda}) = q_1(\boldsymbol{\omega}^c, \boldsymbol{\omega}^p, \Pi) q_2(g, \bar{\lambda})$. To minimize the KL divergence between variational posterior and true posterior, it can be proved that the optimal distribution of each factor is the expectation of the logarithm of the joint distribution taken over variables in the other factor (Bishop, 2006):

$$
\begin{aligned}
q_1^*(\boldsymbol{\omega}^c, \boldsymbol{\omega}^p, \Pi) &\propto e^{\mathbb{E}_{q_2}[\log p(\mathbf{y}^r, \mathbf{y}^c, \mathbf{x}^p, \boldsymbol{\omega}^c, \boldsymbol{\omega}^p, \Pi, g, \bar{\lambda})]}, \\
q_2^*(g, \bar{\lambda}) &\propto e^{\mathbb{E}_{q_1}[\log p(\mathbf{y}^r, \mathbf{y}^c, \mathbf{x}^p, \boldsymbol{\omega}^c, \boldsymbol{\omega}^p, \Pi, g, \bar{\lambda})]}.
\end{aligned}
\tag{C6}
$$

Substituting Eq. (C5) into Eq. (C6), we can obtain the optimal variational distributions. The process of deriving variational posteriors for $\boldsymbol{\omega}^c$, $\boldsymbol{\omega}^p$, Π, and $\bar{\lambda}$ is similar to that in Donner and Opper (2018). The primary distinction lies in the treatment of the latent function g. Further details are provided below.

The optimal density for Pólya-Gamma latent variables

The optimal variational posteriors of $\boldsymbol{\omega}^c$ and $\boldsymbol{\omega}^p$ are

$$
q_1(\boldsymbol{\omega}^c) = \prod_{i=1}^{I_c} \prod_{n=1}^{N_i^c} p_{\mathrm{PG}}(\omega_{i,n}^c \mid 1, \tilde{g}_{i,n}^c), \quad q_1(\boldsymbol{\omega}^p) = \prod_{i=1}^{I_p} \prod_{n=1}^{N_i^p} p_{\mathrm{PG}}(\omega_{i,n}^p \mid 1, \tilde{g}_{i,n}^p), \tag{C7}
$$

where $\tilde{g}_{i,n}^{\cdot} = \sqrt{\mathbb{E}[g_{i,n}^{\cdot\,2}]}$ and we adopt the tilted Pólya-Gamma distribution $p_{\mathrm{PG}}(\omega \mid b, c) \propto e^{-c^2\omega/2} p_{\mathrm{PG}}(\omega \mid b, 0)$ (Polson et al., 2013).

The optimal intensity for marked Poisson processes

The derivation of optimal variational posterior of $\Pi = \{\Pi_i\}_{i=1}^{I_p}$ is challenging, so we provide some details below. After taking expectation, we can obtain

$$
\begin{aligned}
q_1(\Pi_i) =& \frac{p_{\tilde{\Lambda}_i}(\Pi_i \mid \bar{\lambda}_i^1)\prod_{(\omega,\mathbf{x})\in\Pi_i} e^{-\frac{\mathbb{E}[g_i^p(\mathbf{x})]}{2}-\frac{\mathbb{E}[g_i^p(\mathbf{x})^2]}{2}\omega-\log 2}}{\iint p_{\tilde{\Lambda}_i}(\Pi_i \mid \bar{\lambda}_i^1)\prod_{(\omega,\mathbf{x})\in\Pi_i} e^{-\frac{\mathbb{E}[g_i^p(\mathbf{x})]}{2}-\frac{\mathbb{E}[g_i^p(\mathbf{x})^2]}{2}\omega-\log 2}d\omega d\mathbf{x}} \\
=& p_{\tilde{\Lambda}_i}(\Pi_i \mid \bar{\lambda}_i^1)\prod_{(\omega,\mathbf{x})\in\Pi_i} e^{-\frac{\mathbb{E}[g_i^p(\mathbf{x})]}{2}-\frac{\mathbb{E}[g_i^p(\mathbf{x})^2]}{2}\omega-\log 2} \\
& \exp\left(\int_{\mathcal{X}}\int_0^{\infty}(1-e^{-\frac{\mathbb{E}[g_i^p(\mathbf{x})]}{2}-\frac{\mathbb{E}[g_i^p(\mathbf{x})^2]}{2}\omega-\log 2})\bar{\lambda}_i^1 p_{\mathrm{PG}}(\omega \mid 1,0)d\omega d\mathbf{x}\right) \\
=& \prod_{(\omega,\mathbf{x})\in\Pi_i}\bar{\lambda}_i^1 p_{\mathrm{PG}}(\omega \mid 1,0)e^{-\frac{\mathbb{E}[g_i^p(\mathbf{x})]}{2}-\frac{\mathbb{E}[g_i^p(\mathbf{x})^2]}{2}\omega-\log 2} \\
& \exp\left(-\int_{\mathcal{X}}\int_0^{\infty}\bar{\lambda}_i^1 p_{\mathrm{PG}}(\omega \mid 1,0)e^{-\frac{\mathbb{E}[g_i^p(\mathbf{x})]}{2}-\frac{\mathbb{E}[g_i^p(\mathbf{x})^2]}{2}\omega-\log 2}d\omega d\mathbf{x}\right),
\end{aligned}
\tag{C8}
$$

where $\bar{\lambda}_i^1 = e^{\mathbb{E}[\log\bar{\lambda}_i]}$ and $\tilde{\Lambda}_i(\mathbf{x},\omega) = \bar{\lambda}_i^1 p_{\mathrm{PG}}(\omega \mid 1,0)$. The second line of Eq. (C8) used Campbell's theorem $\mathbb{E}_{\Pi_i}\left[\exp\left(\sum_{(\mathbf{x},\omega)\in\Pi_i} h(\mathbf{x},\omega)\right)\right] = \exp\left[\iint\left(e^{h(\mathbf{x},\omega)}-1\right)\tilde{\Lambda}_i(\mathbf{x},\omega)d\omega d\mathbf{x}\right]$. It is easy to see the posterior intensity of Π_i is

$$
\begin{aligned}
\Lambda_i^1(\mathbf{x},\omega) &= \bar{\lambda}_i^1 p_{\mathrm{PG}}(\omega \mid 1,0)e^{-\frac{\mathbb{E}[g_i^p(\mathbf{x})]}{2}-\frac{\mathbb{E}[g_i^p(\mathbf{x})^2]}{2}\omega-\log 2} \\
&= \bar{\lambda}_i^1 s(-\tilde{g}_i^p(\mathbf{x}))p_{\mathrm{PG}}(\omega \mid 1,\tilde{g}_i^p(\mathbf{x}))e^{(\tilde{g}_i^p(\mathbf{x})-\bar{g}_i^p(\mathbf{x}))/2},
\end{aligned}
\tag{C9}
$$

where we adopt $e^{-c^2\omega/2}p_{\mathrm{PG}}(\omega \mid b,0) = 2\,s(-c)e^{c/2}p_{\mathrm{PG}}(\omega \mid b,c)$ (Polson et al., 2013), $\tilde{g}_i^p(\mathbf{x}) = \sqrt{\mathbb{E}[g_i^p(\mathbf{x})^2]}$, $\bar{g}_i^p(\mathbf{x}) = \mathbb{E}[g_i^p(\mathbf{x})]$.

The optimal density for intensity upper-bounds

The optimal variational posterior of $\bar{\lambda}$ is

$$
q_2(\bar{\lambda}) = \prod_{i=1}^{I_p} p_{\mathrm{Ga}}(\bar{\lambda}_i \mid N_i^p + R_i, |\mathcal{X}|),
\tag{C10}
$$

where p_{Ga} is Gamma density, $R_i = \int_{\mathcal{X}}\int_0^{\infty}\Lambda_i^1(\mathbf{x},\omega)d\omega d\mathbf{x}$, $|\mathcal{X}|$ is the domain size.

The optimal density for latent functions

The derivation of optimal variational posterior of g is challenging, so we provide some details below. After taking expectation, we can obtain

$$\begin{aligned}
\log q_2(g) = {} & \sum_{i=1}^{I_r}\sum_{n=1}^{N_i^r}\log\mathcal{N}(y_{i,n}^r \mid g_{i,n}^r, \sigma_i^2) + \sum_{i=1}^{I_c}\sum_{n=1}^{N_i^c}\left[\frac{y_{i.n}^c g_{i,n}^c}{2} - \frac{{g_{i,n}^c}^2}{2}\mathbb{E}[\omega_{i,n}^c]\right] + \\
& \sum_{i=1}^{I_p}\left[\sum_{n=1}^{N_i^p}\frac{g_{i,n}^p}{2} - \frac{{g_{i,n}^p}^2}{2}\mathbb{E}[\omega_{i,n}^p] - \mathbb{E}_{\Pi_i}\sum_{(\omega,\mathbf{x})\in\Pi_i}\frac{g_i^p(\mathbf{x})}{2} + \frac{g_i^p(\mathbf{x})^2}{2}\omega\right] + \log p(g) + C \\
= {} & \sum_{i=1}^{I_r}\sum_{n=1}^{N_i^r}\log\mathcal{N}(g_{i,n}^r \mid y_{i,n}^r, \sigma_i^2) + \sum_{i=1}^{I_c}\sum_{n=1}^{N_i^c}\log\mathcal{N}(g_{i,n}^c \mid \frac{y_{i,n}^c}{2\mathbb{E}[\omega_{i,n}^c]}, \frac{1}{\mathbb{E}[\omega_{i,n}^c]}) \\
& + \sum_{i=1}^{I_p}\left[\int_{\mathcal{X}}\sum_{n=1}^{N_i^p}\left(\frac{g_i^p(\mathbf{x})}{2} - \frac{g_i^p(\mathbf{x})^2}{2}\mathbb{E}[\omega_{i,n}^p]\right)\delta(\mathbf{x}-\mathbf{x}_{i,n}^p)d\mathbf{x}\right. \\
& \left. - \int_{\mathcal{X}}\int_0^\infty\left(\frac{g_i^p(\mathbf{x})}{2} + \frac{g_i^p(\mathbf{x})^2}{2}\omega\right)\Lambda_i^1(\mathbf{x},\omega)d\omega d\mathbf{x}\right] + \log p(g) + C \\
= {} & \sum_{i=1}^{I_r}\sum_{n=1}^{N_i^r}\log\mathcal{N}(g_{i,n}^r \mid y_{i,n}^r, \sigma_i^2) + \sum_{i=1}^{I_c}\sum_{n=1}^{N_i^c}\log\mathcal{N}(g_{i,n}^c \mid \frac{y_{i,n}^c}{2\mathbb{E}[\omega_{i,n}^c]}, \frac{1}{\mathbb{E}[\omega_{i,n}^c]}) \\
& + \sum_{i=1}^{I_p}\left[\int_{\mathcal{X}}\left(\frac{1}{2}\sum_{n=1}^{N_i^p}\delta(\mathbf{x}-\mathbf{x}_{i,n}^p) - \frac{1}{2}\int_0^\infty\Lambda_i^1(\mathbf{x},\omega)d\omega\right)g_i^p(\mathbf{x})d\mathbf{x} + \log p(g)\right. \\
& \left. -\frac{1}{2}\int_{\mathcal{X}}\left(\sum_{n=1}^{N_i^p}\mathbb{E}[\omega_{i,n}^p]\delta(\mathbf{x}-\mathbf{x}_{i,n}^p) + \int_0^\infty\omega\Lambda_i^1(\mathbf{x},\omega)d\omega\right)g_i^p(\mathbf{x})^2 d\mathbf{x}\right] + C \\
= {} & \sum_{i=1}^{I_r}\sum_{n=1}^{N_i^r}\log\mathcal{N}(g_{i,n}^r \mid y_{i,n}^r, \sigma_i^2) + \sum_{i=1}^{I_c}\sum_{n=1}^{N_i^c}\log\mathcal{N}(g_{i,n}^c \mid \frac{y_{i,n}^c}{2\mathbb{E}[\omega_{i,n}^c]}, \frac{1}{\mathbb{E}[\omega_{i,n}^c]}) \\
& + \sum_{i=1}^{I_p}\left[\int_{\mathcal{X}}B_i(\mathbf{x})g_i^p(\mathbf{x})d\mathbf{x} - \frac{1}{2}\int_{\mathcal{X}}A_i(\mathbf{x})g_i^p(\mathbf{x})^2 d\mathbf{x}\right] + \log p(g) + C,
\end{aligned} \tag{C11}$$

where $A_i(\mathbf{x}) = \sum_{n=1}^{N_i^p}\mathbb{E}[\omega_{i,n}^p]\delta(\mathbf{x}-\mathbf{x}_{i,n}^p) + \int_0^\infty\omega\Lambda_i^1(\mathbf{x},\omega)d\omega$ and $B_i(\mathbf{x}) = \frac{1}{2}\sum_{n=1}^{N_i^p}\delta(\mathbf{x}-\mathbf{x}_{i,n}^p) - \frac{1}{2}\int_0^\infty\Lambda_i^1(\mathbf{x},\omega)d\omega$.

The computation of Eq. (C11) suffers from a cubic complexity w.r.t. the number of data points in regression, classification and point process tasks. We use the inducing inputs formalism to make the inference scalable. We denote M inducing inputs $[\mathbf{x}_1 \ldots, \mathbf{x}_M]^\top$ on the domain $\mathcal{X}$ for each task. The function values of basis function f_q at these inducing inputs are defined as $\mathbf{f}_{q,\mathbf{x}_m}$. Then we can obtain the function values of task-specific latent function g_i at these inducing inputs $\mathbf{g}_{\mathbf{x}_m}^i = \sum_{q=1}^Q w_{i,q}\mathbf{f}_{q,\mathbf{x}_m}$. If we define $\mathbf{g}_{\mathbf{x}_m} = [\mathbf{g}_{1,\mathbf{x}_m}^\top, \ldots, \mathbf{g}_{I,\mathbf{x}_m}^\top]^\top$, $\mathbf{g}_{\mathbf{x}_m} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{x}_m\mathbf{x}_m})$ where $\mathbf{K}_{\mathbf{x}_m\mathbf{x}_m}$ is the MOGP covariance on $\mathbf{x}_m$ for all tasks and $\mathbf{g}_{\mathbf{x}_m}^i \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{x}_m\mathbf{x}_m}^i)$ where $\mathbf{K}_{\mathbf{x}_m\mathbf{x}_m}^i$ is i-th diagonal block of $\mathbf{K}_{\mathbf{x}_m\mathbf{x}_m}$. Given $\mathbf{g}_{\mathbf{x}_m}^i$, we assume the function $g_i(\mathbf{x})$ is the posterior mean function $g_i(\mathbf{x}) = \mathbf{k}_{\mathbf{x}_m\mathbf{x}}^{i\top}\mathbf{K}_{\mathbf{x}_m\mathbf{x}_m}^{i^{-1}}\mathbf{g}_{\mathbf{x}_m}^i$ where $\mathbf{k}_{\mathbf{x}_m\mathbf{x}}^i$ is the kernel w.r.t. inducing points and predictive points for i-th task. Therefore, $\{g_{i,n}^r\}_{i=1}^{I_r}$, $\{g_{i,n}^c\}_{i=1}^{I_c}$ and $\{g_i^p(\mathbf{x})\}_{i=1}^{I_p}$ can be written as

$$\mathbf{g}_i^r = \mathbf{K}_{\mathbf{x}_m\mathbf{x}_n}^{r,i\top}\mathbf{K}_{\mathbf{x}_m\mathbf{x}_m}^{r,i^{-1}}\mathbf{g}_{\mathbf{x}_m}^{r,i}, \mathbf{g}_i^c = \mathbf{K}_{\mathbf{x}_m\mathbf{x}_n}^{c,i\top}\mathbf{K}_{\mathbf{x}_m\mathbf{x}_m}^{c,i^{-1}}\mathbf{g}_{\mathbf{x}_m}^{c,i}, g_i^p(\mathbf{x}) = \mathbf{k}_{\mathbf{x}_m\mathbf{x}}^{p,i\top}\mathbf{K}_{\mathbf{x}_m\mathbf{x}_m}^{p,i^{-1}}\mathbf{g}_{\mathbf{x}_m}^{p,i}, \tag{C12}$$

where $\mathbf{g}_i^r = [g_{i,1}^r, \ldots, g_{i,N_i^r}^r]^\top$, $\mathbf{g}_i^c = [g_{i,1}^c, \ldots, g_{i,N_i^c}^c]^\top$, $g_i^p(\mathbf{x})$ is the function value of g_i^p on $\mathbf{x}$.

Substituting Eq. (C12) into Eq. (C11), we obtain the inducing points version of Eq. (C11):

$$
\begin{aligned}
q_2(\mathbf{g}_{\mathbf{x}_m}) \propto & \prod_{i=1}^{I_r} \mathcal{N}(\mathbf{K}^{r,i\top}_{\mathbf{x}_m\mathbf{x}_n} \mathbf{K}^{r,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m} \mathbf{g}^{r,i}_{\mathbf{x}_m} \mid \mathbf{y}^r_i, \mathrm{diag}(\sigma^2_i)) \\
& \cdot \prod_{i=1}^{I_c} \mathcal{N}(\mathbf{K}^{c,i\top}_{\mathbf{x}_m\mathbf{x}_n} \mathbf{K}^{c,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m} \mathbf{g}^{c,i}_{\mathbf{x}_m} \mid \frac{\mathbf{y}^c_i}{2\mathbb{E}[\boldsymbol{\omega}^c_i]}, \mathrm{diag}(\frac{1}{\mathbb{E}[\boldsymbol{\omega}^c_i]})) \\
& \cdot \prod_{i=1}^{I_p} \exp\left(\int_{\mathcal{X}} B_i(\mathbf{x}) \mathbf{k}^{p,i\top}_{\mathbf{x}_m\mathbf{x}} d\mathbf{x} \mathbf{K}^{p,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m} \mathbf{g}^{p,i}_{\mathbf{x}_m} \right. \\
& \left. -\frac{1}{2} \mathbf{g}^{p,i\top}_{\mathbf{x}_m} \mathbf{K}^{p,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m} \int_{\mathcal{X}} A_i(\mathbf{x}) \mathbf{k}^{p,i}_{\mathbf{x}_m\mathbf{x}} \mathbf{k}^{p,i\top}_{\mathbf{x}_m\mathbf{x}} d\mathbf{x} \mathbf{K}^{p,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m} \mathbf{g}^{p,i}_{\mathbf{x}_m} \right) \\
& \cdot \mathcal{N}(\mathbf{g}_{\mathbf{x}_m} \mid \mathbf{0}, \mathbf{K}_{\mathbf{x}_m\mathbf{x}_m}).
\end{aligned}
\tag{C13}
$$

It is easy to see the third line of Eq. (C13) is a multivariate Gaussian distribution of $\mathbf{g}^{p,i}_{\mathbf{x}_m}$. The likelihoods of $\mathbf{g}^{r,i}_{\mathbf{x}_m}$ for regression, $\mathbf{g}^{c,i}_{\mathbf{x}_m}$ for classification and $\mathbf{g}^{p,i}_{\mathbf{x}_m}$ for point process tasks are all Gaussian distributions, so they are conjugate to the MOGP prior and we can obtain the closed-form variational posterior for $\mathbf{g}_{\mathbf{x}_m}$:

$$
q_2(\mathbf{g}_{\mathbf{x}_m}) = \mathcal{N}(\mathbf{g}_{\mathbf{x}_m} \mid \mathbf{m}_{\mathbf{x}_m}, \boldsymbol{\Sigma}_{\mathbf{x}_m}), \tag{C14}
$$

where $\mathbf{g}_{\mathbf{x}_m} = [\mathbf{g}^{r\top}_{\mathbf{x}_m}, \mathbf{g}^{c\top}_{\mathbf{x}_m}, \mathbf{g}^{p\top}_{\mathbf{x}_m}]^\top$, $\mathbf{g}^{\cdot}_{\mathbf{x}_m} = [\mathbf{g}^{\cdot\top}_{1,\mathbf{x}_m}, \dots, \mathbf{g}^{\cdot\top}_{I_\cdot,\mathbf{x}_m}]^\top$ and

$$
\boldsymbol{\Sigma}_{\mathbf{x}_m} = \left[\mathrm{diag}\left(\mathbf{H}^r_{\mathbf{x}_m}, \mathbf{H}^c_{\mathbf{x}_m}, \mathbf{H}^p_{\mathbf{x}_m}\right) + \mathbf{K}^{-1}_{\mathbf{x}_m\mathbf{x}_m}\right]^{-1}, \mathbf{m}_{\mathbf{x}_m} = \boldsymbol{\Sigma}_{\mathbf{x}_m}[\mathbf{v}^{r\top}_{\mathbf{x}_m}, \mathbf{v}^{c\top}_{\mathbf{x}_m}, \mathbf{v}^{p\top}_{\mathbf{x}_m}]^\top,
$$

where $\mathbf{H}^{\cdot}_{\mathbf{x}_m} = \mathrm{diag}(\mathbf{H}^{\cdot}_{1,\mathbf{x}_m}, \dots, \mathbf{H}^{\cdot}_{I_\cdot,\mathbf{x}_m})$, $\mathbf{v}^{\cdot}_{\mathbf{x}_m} = [\mathbf{v}^{\cdot\top}_{1,\mathbf{x}_m}, \dots, \mathbf{v}^{\cdot\top}_{I_\cdot,\mathbf{x}_m}]^\top$ and

$$
\begin{aligned}
\mathbf{H}^r_{i,\mathbf{x}_m} &= \mathbf{K}^{r,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m} \mathbf{K}^{r,i}_{\mathbf{x}_m\mathbf{x}_n} \mathbf{D}^r_i \mathbf{K}^{r,i\top}_{\mathbf{x}_m\mathbf{x}_n} \mathbf{K}^{r,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m}, \mathbf{v}^r_{i,\mathbf{x}_m} = \mathbf{K}^{r,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m} \mathbf{K}^{r,i}_{\mathbf{x}_m\mathbf{x}_n} \frac{\mathbf{y}^r_i}{\sigma^2_i}, \\
\mathbf{H}^c_{i,\mathbf{x}_m} &= \mathbf{K}^{c,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m} \mathbf{K}^{c,i}_{\mathbf{x}_m\mathbf{x}_n} \mathbf{D}^c_i \mathbf{K}^{c,i\top}_{\mathbf{x}_m\mathbf{x}_n} \mathbf{K}^{c,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m}, \mathbf{v}^c_{i,\mathbf{x}_m} = \mathbf{K}^{c,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m} \mathbf{K}^{c,i}_{\mathbf{x}_m\mathbf{x}_n} \frac{\mathbf{y}^c_i}{2}, \\
\mathbf{H}^p_{i,\mathbf{x}_m} &= \mathbf{K}^{p,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m} \int_{\mathcal{X}} A_i(\mathbf{x}) \mathbf{k}^{p,i}_{\mathbf{x}_m\mathbf{x}} \mathbf{k}^{p,i\top}_{\mathbf{x}_m\mathbf{x}} d\mathbf{x} \mathbf{K}^{p,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m}, \\
\mathbf{v}^p_{i,\mathbf{x}_m} &= \mathbf{K}^{p,i^{-1}}_{\mathbf{x}_m\mathbf{x}_m} \int_{\mathcal{X}} B_i(\mathbf{x}) \mathbf{k}^{p,i}_{\mathbf{x}_m\mathbf{x}} d\mathbf{x},
\end{aligned}
$$

where $\mathbf{D}^r_i = \mathrm{diag}(1/\sigma^2_i)$ and $\mathbf{D}^c_i = \mathrm{diag}(\mathbb{E}[\boldsymbol{\omega}^c_i])$.

Appendix D. Multi-class classification

In the paper, we mainly focus on the binary classification problem because each binary classification task corresponds to a single latent function. This setting is consistent with the regression and point process tasks in which each task only specifies a single latent function.

For Z-class classification problem, each task corresponds to Z latent functions. The usual likelihood for multi-class classification is the softmax function:

Table 4 The performance of TLL and RT for HMGCP and HetMOGP on three synthetic datasets in Section 5.1. Time in seconds

	Model	TLL(reg)	TLL(cla)	TLL(Cox)	RT (400 iterations)
1	HMGCP	**−33.17**	**−63.57**	**−89.05**	**1.6**
	HetMOGP	−97.80	−66.22	−181.91	708.74
2	HMGCP	**−28.54**	**−55.23**	**−63.54**	**1.79**
	HetMOGP	−98.8	−58.1	−196.71	812.88
3	HMGCP	**−42.43**	**−56.14**	**−72.75**	**1.54**
	HetMOGP	−138.21	−65.01	−172.77	647.70

$$p(y^c_{i,n} = k \mid \mathbf{f}^c_{i,n}) = \frac{e^{(f^{c,k}_{i,n})}}{\sum_{z=1}^{Z} e^{(f^{c,z}_{i,n})}}, \tag{D15}$$

where $f^{c,k}_{i,n} = f^{c,k}_i(\mathbf{x}_n)$, $\mathbf{f}^c_{i,n} = [f^{c,1}_{i,n}, \dots, f^{c,Z}_{i,n}]^\top$, $k \in \{1, \dots, Z\}$. However, the Pólya-Gamma augmentation technique for binary classification can not be directly employed in the softmax function. Galy-Fajou et al. (2020) and Snell and Zemel (2021) proposed the *logistic-softmax function* and the *one-vs-each softmax approximation* respectively that enable us to employ Pólya-Gamma augmentation to obtain a conditionally conjugate model for multi-class classification tasks. Both methods mentioned above can be incorporated into our framework in the multi-class classification scenario. We refer the readers to Galy-Fajou et al. (2020); Snell and Zemel (2021) for more details.

Appendix E. Comparison with HetMOGP

One anonymous reviewer point out that an important baseline to compare against is Moreno-Muñoz et al. (2018) that can also handle regression, classification and counting data, even if the discretized Poisson distribution likelihood is used instead of the continuous point process likelihood considered in this work. Moreno-Muñoz et al. (2018) used the generic variational inference method mentioned in the introduction for parameter posterior, so this comparison can demonstrate the advantage of using data augmentation for conjugate operations.

We compare the performance of TLL and RT for HMGCP and heterogeneous multi-output Gaussian process (HetMOGP) (Moreno-Muñoz et al., 2018) on the synthetic data from Sects. 5.1 and 5.2. Since HetMOGP can only handle discrete count data, we discretize the original observation window [0, 100] into 100 bins and then calculate the number of points in each bin separately. We use the default hyperparameter settings in the demo code provided by Moreno-Muñoz et al. (2018). The results are shown in Tables 4 and 5. From Tables 4 and 5, we can see that HMGCP has the better TLL than HetMOGP that is trained on the discrete count data. For a fair comparison of efficiency, we run both HMGCP and HetMOGP on all tasks, and our inference is much faster than HetMOGP. This is because, for HetMOGP, it uses the generic variational inference, so the numerical optimization has to be performed during the variational iterations; while for our model HMGCP, the variational iterations have completely analytical expressions due to data augmentation, so it leads to the more efficient computation. It is worth noting that the running times presented in Tables 4 and 5 encompass all tasks (regression, classification, and Cox processes),

Table 5 The performance of TLL and RT for HMGCP and HetMOGP on synthetic datasets in Section 5.2 over ten random configurations of missing gaps with three different missing-gap widths (0 means complete data).

Gap Width	Model	TLL(reg)	TLL(cla)	TLL(Cox)	RT (2000 iterations)
0	HMGCP	**−50.61**	**−56.67**	**−120.55**	**12.9**
	HetMOGP	−101.19	−64.18	−380.63	4029.75
5	HMGCP	**−50.76**(0.92)	**−56.74**(0.51)	**−122.94**(2.27)	**12.5**
	HetMOGP	−105.19 (2.92)	−73.99 (12.01)	−391.38 (29.00)	3826.56
10	HMGCP	**−52.14**(1.94)	**−56.82**(0.69)	**−128.49**(5.74)	**11.7**
	HetMOGP	−104.45 (10.49)	−66.12	−414.58 (21.78)	3424.73

The mean and standard deviation (in brackets) are provided. TLL(Cox) is the sum of TLLs of two Cox processes. Time in seconds

resulting in longer duration compared to those reported in Sects. 5.1 and 5.2, which are solely based on the Cox process tasks.

Author Contributions All authors have contributed significantly to this work. The theoretical derivation was primarily performed by FZ, and the experimental validation was primarily performed by FZ, QK and ZD with support from FH and PC. All authors wrote the manuscript. JZ supervised the whole project.

Funding This work was conceived and conducted when FZ was doing post-doc at Tsinghua University. This work was supported by NSFC Projects (Nos. 62061136001, 62106121, 61621136008, U19A2081), the MOE Project of Key Research Institute of Humanities and Social Sciences (22JJD110001), the National Key Research and Development Program of China (2021ZD0110502), the major key project of PCL (No. PCL2021A12), the fund for building world-class universities (disciplines) of Renmin University of China (No. KYGJC2023012), and Tsinghua Guo Qiang Institute, and the High Performance Computing Center, Tsinghua University, and the Public Computing Cloud, Renmin University of China.

Data Availability Experiments were done on the public dataset.

Code Availability Code is public.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

Adams, R.P., Murray, I., MacKay, D.J. (2009). Tractable nonparametric Bayesian inference in Poisson processes with Gaussian process intensities. In: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, pp 9–16.

Aglietti, V., Damoulas, T., Bonilla, E.V. (2019). Efficient inference in multi-task Cox process models. In: The 22nd International Conference on Artificial Intelligence and Statistics, PMLR, pp 537–546.

Alvarez, M.A., Lawrence, N.D. (2008). Sparse convolved Gaussian processes for multi-output regression. In: NIPS, pp 57–64.

Álvarez, M. A., Rosasco, L., & Lawrence, N. D. (2012). Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning, 4*(3), 195–266.

Álvarez, M.A., Ward, W., Guarnizo, C. (2019). Non-linear process convolutions for multi-output Gaussian processes. In: The 22nd International Conference on Artificial Intelligence and Statistics, PMLR, pp 1969–1977.

Amari, S. I. (1998). Natural gradient works efficiently in learning. *Neural Computation, 10*(2), 251–276.

Besag, J. (1994). Discussion on the paper by Grenander and Miller. *Journal of the Royal Statistical Society: Series B (Methodological), 56*, 591–592.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Cham: Springer.

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association, 112*(518), 859–877.

Bonilla, E.V., Chai, K.M.A., Williams, C.K.I. (2007). Multi-task Gaussian process prediction. In: Platt JC, Koller D, Singer Y, et al (eds) Advances in neural information processing systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007. Curran Associates, Inc., pp 153–160.

Cunningham, J.P., Shenoy, K.V., Sahani, M. (2008). Fast Gaussian process methods for point process intensity estimation. In: International Conference on Machine Learning, ACM, pp 192–199.

Daley, D.J., Vere-Jones, D. (2003). An introduction to the theory of point processes. Vol. I. Probability and its applications.

Dezfouli, A., & Bonilla, E. V. (2015). Scalable inference for Gaussian process models with black-box likelihoods. *Advances in Neural Information Processing Systems, 28*, 1414–1422.

Diggle, P. J., Moraga, P., Rowlingson, B., et al. (2013). Spatial and spatio-temporal log-Gaussian Cox processes: Extending the geostatistical paradigm. *Statistical Science, 28*(4), 542–563.

Donner, C., & Opper, M. (2018). Efficient Bayesian inference of sigmoidal Gaussian Cox processes. *Journal of Machine Learning Research, 19*(1), 2710–2743.

Galy-Fajou, T., Wenzel, F., Donner, C., et al. (2020). Multi-class Gaussian process classification made conjugate: Efficient inference via data augmentation. In: Uncertainty in Artificial Intelligence, PMLR, pp 755–765.

Hensman, J., Matthews, A., Ghahramani, Z. (2015). Scalable variational Gaussian process classification. In: Artificial Intelligence and Statistics, PMLR, pp 351–360.

Hoffman, M.D., Blei, D.M,. Wang, C., et al. (2013). Stochastic variational inference. Journal of Machine Learning Research 14(5).

Jahani, S., Zhou, S., Veeramani, D., et al. (2021). Multioutput Gaussian process modulated Poisson processes for event prediction. IEEE Transactions on Reliability.

Journel, A. G., & Huijbregts, C. J. (1976). *Mining geostatistics*. London: Academic Press.

Lasko, T.A. (2014). Efficient inference of Gaussian-process-modulated renewal processes with application to medical event data. In: Uncertainty in artificial intelligence: Proceedings of the Conference on Uncertainty in Artificial Intelligence, NIH Public Access, p 469.

Li, C., Zhu, J., Chen, J. (2014). Bayesian max-margin multi-task learning with data augmentation. In: International Conference on Machine Learning, PMLR, pp 415–423.

Lian, W., Henao, R., Rao, V., et al. (2015). A multitask point process predictive model. In: International Conference on Machine Learning, PMLR, pp 2030–2038.

Lloyd, C., Gunter, T., Osborne, M., et al. (2015). Variational inference for Gaussian process modulated Poisson processes. In: International Conference on Machine Learning, pp 1814–1822.

Møller, J., Syversveen, A. R., & Waagepetersen, R. P. (1998). Log Gaussian Cox processes. *Scandinavian Journal of Statistics, 25*(3), 451–482.

Moreno-Muñoz, P., Artés, A., Álvarez, M. (2018). Heterogeneous multi-output Gaussian process prediction. Advances in Neural Information Processing Systems 31

Mutny, M., Krause, A. (2021). No-regret algorithms for capturing events in Poisson point processes. In: International Conference on Machine Learning, PMLR, pp 7894–7904.

Neal, R. M. (1993). Probabilistic inference using Markov chain Monte Carlo methods. Department of Computer Science: University of Toronto Toronto, ON, Canada.

Nguyen, T. V., & Bonilla, E. V. (2014). Automated variational inference for Gaussian process models. *Advances in Neural Information Processing Systems, 27*, 1404–1412.

Polson, N. G., Scott, J. G., & Windle, J. (2013). Bayesian inference for logistic models using Pólya-Gamma latent variables. *Journal of the American statistical Association, 108*(504), 1339–1349.

Rasmussen, C.E. (2003). Gaussian processes in machine learning. In: Summer School on Machine Learning, Springer, pp 63–71.

Shirota, S., Gelfand, A.E. (2017) Space and circular time log Gaussian Cox processes with application to crime event data. The Annals of Applied Statistics pp 481–503.

Snell, J., Zemel, R.S. (2021). Bayesian few-shot classification with one-vs-each Pólya-Gamma augmented Gaussian processes. In: International Conference on Learning Representations, ICLR 2021. OpenReview.net.

Soleimani, H., Hensman, J., & Saria, S. (2017). Scalable joint models for reliable uncertainty-aware event prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 40*(8), 1948–1963.

Taylor, B. M., Davies, T. M., Rowlingson, B. S., et al. (2015). Bayesian inference and data augmentation schemes for spatial, spatiotemporal and multivariate log-Gaussian Cox processes in R. *Journal of Statistical Software, 63*(1), 1–48.

Titsias, M. (2009). Variational learning of inducing variables in sparse Gaussian processes. In: Artificial Intelligence and Statistics, pp 567–574.

Uspensky, J. V., et al. (1937). *Introduction to mathematical probability*. New York: McGraw-Hill Book Co., Inc.

Ver Hoef, J. M., & Barry, R. P. (1998). Constructing and fitting models for cokriging and multivariable spatial prediction. *Journal of Statistical Planning and Inference, 69*(2), 275–294.

Wenzel, F., Galy-Fajou, T., Donner, C., et al. (2019). Efficient Gaussian process classification using Pólya-Gamma data augmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 5417–5424.

Williams, C. K., & Rasmussen, C. E. (2006). *Gaussian processes for machine learning* (Vol. 2). MIT Press.

Wood, F., Meent, J.W., Mansinghka, V. (2014). A new approach to probabilistic programming inference. In: Artificial intelligence and statistics, PMLR, pp 1024–1032.

Zhou, F., Li, Z., Fan, X., et al. (2020). Efficient inference for nonparametric Hawkes processes using auxiliary latent variables. *Journal of Machine Learning Research, 21*(241), 1–31.

Zhou, F., Zhang, Y., Zhu, J. (2021). Efficient inference of flexible interaction in spiking-neuron networks. In: International Conference on Learning Representations.

Zhou, F., Kong, Q., Deng, Z., et al. (2022). Efficient inference for dynamic flexible interactions of neural populations. *Journal of Machine Learning Research, 23*(211), 1–49.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.