Spatial Logistic Gaussian Process (SLGP): Implementing a nonparametric approach to conditional density estimation on heterogeneous data

A Preprint

Athénaïs Gautier
DTIS, ONERA, Université Paris-Saclay, 91120, Palaiseau, France
E-mail: `athenais.gautier@onera.fr`

February 11, 2026

Abstract

Spatial Logistic Gaussian Processes (SLGPs) provide a flexible and probabilistic framework for spatially dependent density estimation in heterogeneous settings, particularly when data sampes are sparsely or unevenly distributed across space. By applying a logistic density transformation to a Gaussian Process indexed over a joint index–response domain, SLGPs yield fields of probability density functions that can adapt to complex variations in shape, location, and modality.

This paper offers a practical introduction to SLGP-based modeling, emphasizing implementation choices such as basis function representations and numerical quadrature schemes for integral approximations. Key aspects of model specification, computational stability, and tuning are discussed in detail. These concepts are then illustrated through an R implementation with code snippets and examples. In particular, we demonstrate the ability of SLGPs to recover conditional distributions without relying on strong distributional assumptions or replicated data-using the Boston Housing dataset as a case study.

1 Introduction

Precise statistical inference in complex systems with spatial dependence often necessitates either a substantial computational budget for evaluating the system at numerous input variables $\mathbf{x}$ or the utilization of assumptions about the model output $T_{\mathbf{x}}$. The latter approach is commonly preferred and widely employed in Uncertainty Quantification (UQ), falling under the umbrella term of Surrogate Modeling. Surrogate modeling entails approximating either the response variable or its distribution in the case of a stochastic system. These models are not only valuable for general inference purposes, such as learning and predicting moments, quantiles or excursion sets, but also for facilitating more intricate forms of inference, including Bayesian optimization or solving stochastic inverse problems.

However, the task of learning the distribution of $T_{\mathbf{x}}$ becomes particularly challenging when the spatial dependence involves not only the mean and variance but also other characteristics such as shape, uni-modal versus multi-modal nature, and more. Furthermore, real-world applications often present heterogeneously scattered data or lack replicates, further exacerbating the difficulty of the problem. Figure 1 below illustrates a typical scenario encountered in such applications, where we seek to estimate spatially dependent probability densities (represented as red curves) based on localised potentially sparse samples thereof (summarised as histograms).

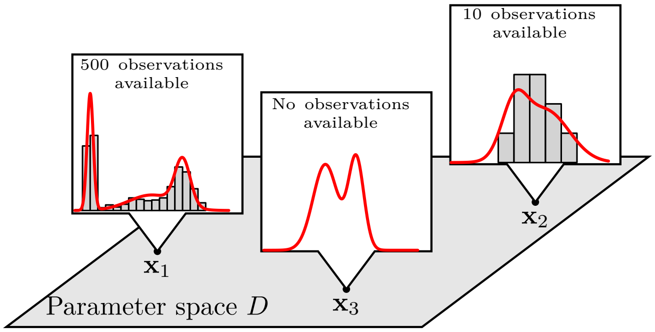
500 observations available
No observations available
10 observations available
$\mathbf{x}_1$
$\mathbf{x}_3$
$\mathbf{x}_2$
Parameter space D

Figure 1: One challenging setting: Estimating the underlying probability distributions (red curve) by integrating data that are heterogeneously scattered across space (histogram).

Numerous approaches exist for estimating spatially dependent probability distributions, each addressing different modeling aspects. But, to the best of our knowledge, none of the current approaches allow for simultaneously tackling all the challenges at hand. Therefore, we proposed Spatial Logistic Gaussian Process (SLGP) modeling, which provides a non-parametric and probabilistic framework for modeling such objects based on scattered data, even in the absence of replicates. SLGP models are obtained by transforming a suitably regular Gaussian Process (GP) through a composition of exponentiation and normalization steps, resulting in fields of probability distributions. The mathematical details underlying these stochastic processes are elaborated in **?** and are not the main focus here; instead, we emphasize implementation aspects and practical recommendations.

In line with the objective to demonstrate the potential of SLGP modeling, we introduce the Boston Housing dataset **?** as a running example in this paper. This dataset, classically used in regression, contains information about the housing values in suburbs of Boston. We shall focus on predicting the distribution of the median value of owner-occupied homes, denoted as the variable *med*, based on the proportion of owner-occupied units constructed before 1940, represented by the variable age. As depicted in Figure 2, the dataset reveals a general trend where older houses often exhibit a decline in value, with a few notable exceptions. This trend can be intuitively interpreted as a result of survivor bias, where older properties that remain are those of higher structural quality, thereby maintaining or increasing in value over time.

This dataset offers a compactly supported, one-dimensional domain for the indexing variable $\mathbf{x}$, allowing for a clear visualisation of distributional changes and providing challenging setting, in which data are unequally distributed in space, contain few replicates, and exhibit significant variation in shape and modality.

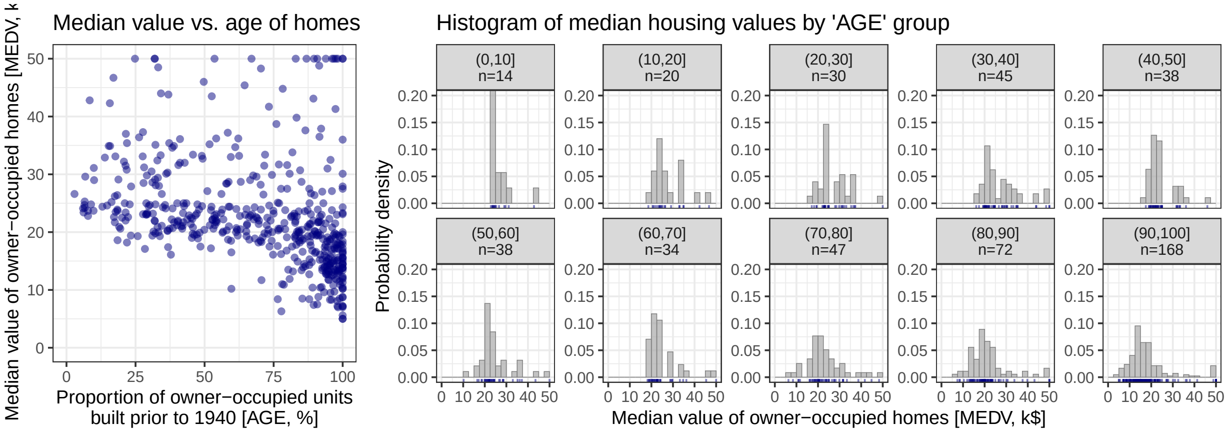

Figure 2: A visual representation of the dependency of the median value of owner-occupied homes on proportion of owner-occupied units constructed before 1940 in the Boston Housing dataset.

The remainder of this paper is structured as follows. In Section 2, we present the class of model considered, a visual proof of concept of the modeling flexibility of it and the formulae to condition on data. In Section 3, we discuss implementation choices and practical considerations to perform SLGP-based density estimation. In Section 4, we show the main functionalities of the package, through an application to the Boston Housing dataset with code chunks. Further illustrations and technical details are provided in the appendices. Appendices A, B and C.

2 The Spatial Logistic Gaussian Process, or SLGP

2.1 A prior over indexed probability density functions

The SLGP model, related to (**?**) central to the present contribution, as a continuation of (**???**), is itself a spatial generalization of the Logistic Gaussian Process models, which were established and studied in (**???**).

The SLGP for spatially dependent density estimation builds upon a *well-behaved* GP $(Z_{\mathbf{x},t})_{(\mathbf{x},t)\in D\times\mathcal{T}}$ and defines the stochastic process obtained from applying the *spatial logistic density* transformation to Z as follows:

$$\Psi[Z](\mathbf{x},t) = \frac{e^{Z_{\mathbf{x},t}}}{\int_{\mathcal{T}} e^{Z_{\mathbf{x},u}}\,du} \text{ for all } (\mathbf{x},t)\in D\times\mathcal{T} \tag{1}$$

Here and throughout the document, we consider a compact and convex response space $\mathcal{T}\subset\mathbb{R}^{d_t}$ with $d_t\geq 1$ and we further assume that $\mathcal{T}$ has a positive Lebesgue measure. We also consider that $D\subset\mathbb{R}^{d_x}$ with $d_x\geq 1$.

At any fixed $\mathbf{x}$, $\Psi[Z](\mathbf{x},\cdot)$ hence returns a random function that is, by construction, positive and integrates to one. Hence, a SLGP can be (informally) seen as a field of random pdfs. The precise mathematical nature of the objects involved-whether in terms of random measures or densities, and fields thereof-requires careful treatment, for which we refer the reader to (**?**).

Here, we adopt a more intuitive approach and assume that the objects are well-behaved enough to speak of SLGP as random fields of densities. We also show in Figure 7 three draws from a SLGP prior. Since a SLGP models directly the field as a whole, drawing one realisation of it yields a collection of pdfs indexed by $\mathbf{x}$. Under mild conditions on the transformed GP, these pdfs will vary smoothly when $\mathbf{x}$ changes.

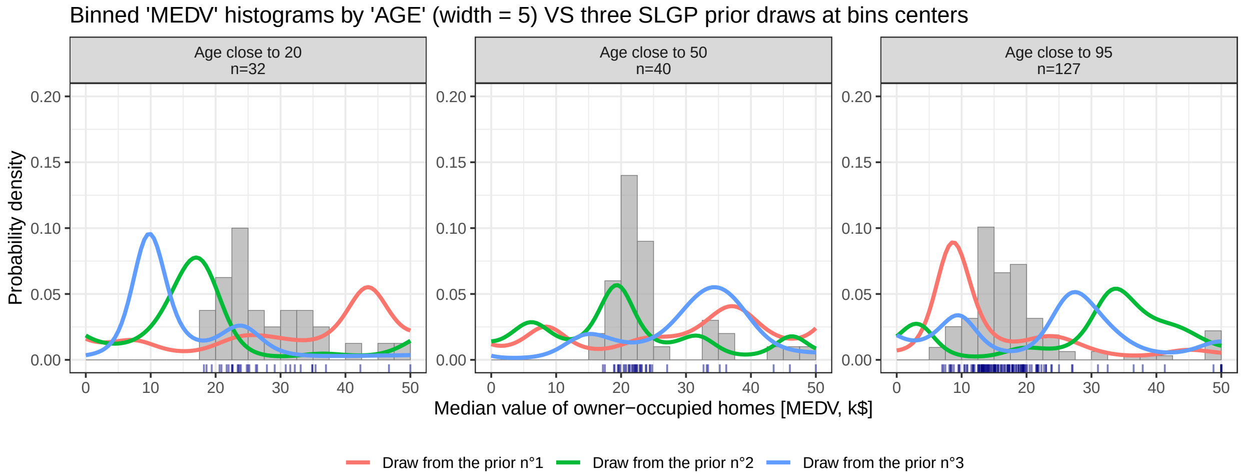

Figure 3: Prior densities of `medv` given `age` at three representative slices, estimated using MCMC samples from the SLGP model. Binned histograms of nearby observations are shown due to insufficient pointwise data to form standard histograms.

Since these are draws from the prior distribution, they do not yet incorporate information from the actual data. Therefore, it is natural that they do not match the empirical histograms. However, the prior samples display a reasonable level of variability and structure, suggesting that the model would be well-suited for density estimation after incorporating data through posterior inference.

2.2 Conditioning on data

We focus on implementation and practicalities. For this purpose, we consider that our observations are obtained by independent sampling of a reference probability density field p_0. By that, we mean that our dataset consists of n couples of locations and observations $\{(\mathbf{x}_i, t_i)\}_{1\leq i\leq n}$, where the $\mathbf{x}_i$ are in $[0,1]^{d_x}$. Moreover, we assume the t_i's are obtained by independent sampling of random variables T_i with respective densities $p_0(\mathbf{x}_i, \cdot)$. The random vector of observations is denoted by $\mathbf{T} = (T_i)_{1\leq i\leq n}$, while we use $\mathbf{t} = (t_i)_{1\leq i\leq n}$ for a realisation of $\mathbf{T}$.

Under the assumption that the observations follow the SLGP model and are independent, we obtain a density of observations conditioned on the underlying GP:

$$\pi\left(\mathbf{T} = t|Z\right) = \prod_{i=1}^{n} \Psi[Z](\mathbf{x}_i, t_i) = \prod_{i=1}^{n} \frac{e^{Z_{\mathbf{x}_i, t_i}}}{\int_{\mathcal{T}} e^{Z_{\mathbf{x}_i, u}}\, du} \tag{2}$$

We insist on the fact that this equation derives naturally from the independence assumption, and that we made no assumptions relative to the index $\mathbf{x}$. Importantly, this likelihood remains valid even when the sample locations are irregularly spaced, with small sample sizes and possibly few or no replicates.

Remark. Note that Equation 2 also applies when the indexing variable $\mathbf{x}$ is not compactly supported, or even not numeric, as long as one can define a covariance kernel on its domain (and hence a GP and an SLGP).

This formulation allows us to incorporate observed data into the SLGP framework and carry out Bayesian inference. Details of the inference and implementation choices will be discussed in the upcoming section. For now, we simply present in figure 4 an illustrative result of such an estimation to highlight the model's practical utility.

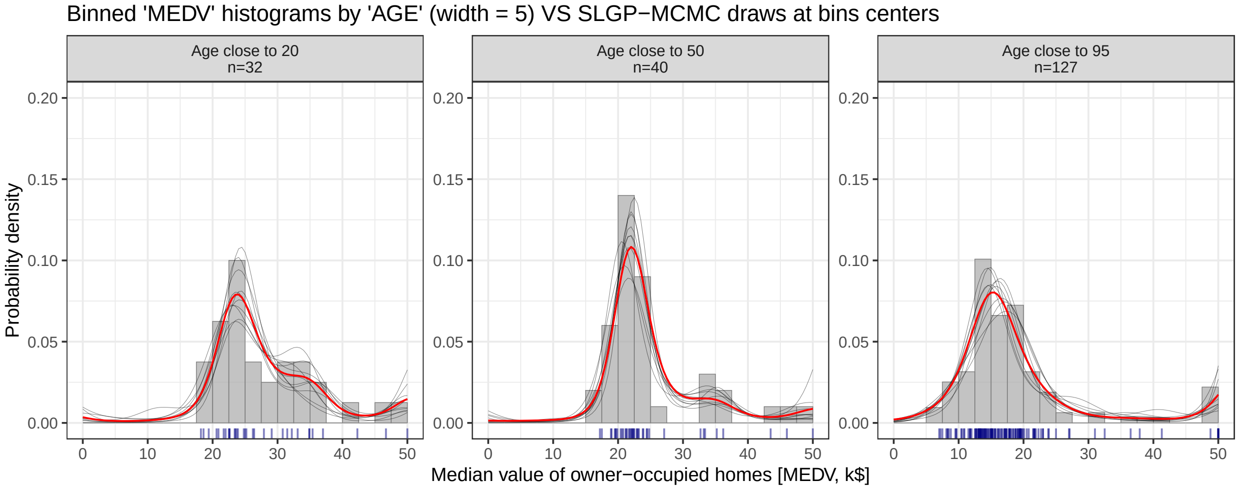

Figure 4: Posterior predictive densities of `medv` given `age` at three representative slices, estimated using MCMC samples from the SLGP model. Binned histograms of nearby observations are shown due to insufficient pointwise data to form standard histograms.

3 Implementation in practice

Applying Spatial Logistic Gaussian Process (SLGP) models to real-world inference comes with several challenges. First, the SLGP's infinite-dimensional nature makes it tricky to formulate the problem. Second, computing or approximating the integrals in Equation 2 requires careful handling. Third, since SLGPs are based on transforming GPs, choosing the right hyperparameters is essential. Addressing these challenges is crucial for applying SLGP models in practical scenarios.

3.1 Addressing the dimensionality of the problem

The first challenge for our model lies in the fact that the integrals in Equations 2 involve values of Z over the whole response domain. This infinite dimensional object makes likelihood-based computations cumbersome.

3.1.1 Finite rank GPs and their use for SLGP modelling

We propose a way to reduce the dimensionality by considering only finite rank GPs. A GP Z on a generic domain S is called a *finite rank* GP if there exists $p \in \mathbb{N}$, and a family of functions $(f_i)_{1\leq i\leq p}$ on S such that:

$$Z(s) = \sum_{j=1}^{p} f_j(s)\varepsilon_j, \ \forall s \in S \tag{3}$$

where $\boldsymbol{\varepsilon} = (\varepsilon_1, ..., \varepsilon_p)^\top$ is a random vector of p i.i.d. $\mathcal{N}(0, \sigma^2)$ random variables.
Remark. Note that from now on, we will use the notation ε (resp. $\boldsymbol{\varepsilon}$) to denote the random variable (resp. random vector), and ϵ (resp. $\boldsymbol{\epsilon}$) for fixed values and realisations thereof.

In particular, we will be interested in finite-rank GPs defined over $D \times \mathcal{T} = [0,1]^{d_x+d_t}$, and as such consider function $(f_i)_{1\leq i\leq p}$ on $[0,1]^{d_x+d_t}$. This yields finite-rank GPs that we can write as follows:

$$Z(\mathbf{x}, t) = \sum_{j=1}^{p} f_j(\mathbf{x}, t)\varepsilon_j = \boldsymbol{\varepsilon}^\top \boldsymbol{F}(\mathbf{x}, t), \ \forall \mathbf{x} \in D, \ t \in \mathcal{T}, \tag{4}$$

where $\boldsymbol{F}(\mathbf{x}, t) := (f_j(\mathbf{x}, t))_{1\leq j\leq p}$ is the vector of basis functions evaluated at $(\mathbf{x}, t)$ and $\boldsymbol{\varepsilon}$ is a random vector of p i.i.d. $\mathcal{N}(0, \sigma^2)$ random variables.
Remark. When the f_i are L^2 orthonormal, this coincides with the Karhunen-Loève expansion of the GP. However, for most kernels, this expansion is not analytically known, and our application does not benefit from the basis functions being orthonormal.
Remark. A finite rank GP defined as in Equation 4 has the following covariance kernel:

$$\mathrm{Cov}\left(Z(\mathbf{x}, t), Z(\mathbf{x}', t')\right) = \sigma^2 \sum_{j=1}^{p} f_j(\mathbf{x}, t) f_j(\mathbf{x}', t') \tag{5}$$

The posterior distribution of $\boldsymbol{\varepsilon}$, given data $\{(\mathbf{x}_i, t_i)\}_{i=1}^n$ can be derived from Equation 2. Denoting by ϕ_p the pdf of the p-variate standard normal distribution, we have:

$$\pi[\boldsymbol{\epsilon}|\mathbf{T} = \mathbf{t}] \propto \phi_p(\boldsymbol{\epsilon}) \prod_{i=1}^{n} \frac{e^{\sum_{j=1}^{p} \epsilon_j f_j(\mathbf{x}_i, t_i)}}{\int_{\mathcal{T}} e^{\sum_{j=1}^{p} \epsilon_j f_j(\mathbf{x}_i, u)} du} \tag{6}$$

3.1.2 Basis functions considered

Theoretically, any basis function could be used in the implementation. However, in practice, due to the exponential transformation attention must be given to the numerical stability of their model. For instance, Figure 5 illustrates why Legendre polynomials are ill-advised as basis functions, as they give processes that are too sensitive to the number of basis functions and display poor numerical stability.

Through repeated numerical experiments, we identified some families of functions that seem to display sufficiently tame behaviours and make the implementation less prone to numerical overflow. From here on, we review such approaches.

Leveraging inducing points In **?**, the authors focus on density estimation. To this extent, they rely on LGPs (the non-spatialised counterpart of SLGPs). In their work, they propose to address the dimensionality by considering a reasonable number of inducing points. Instead of working with a GP $Z = (Z(t))_{t\in\mathcal{T}} \sim \mathcal{GP}(0, k)$, their models approximate it by $W(t) := \mathbb{E}[Z(t)|Z(t_1), ..., Z(t_p)]$ (for $p \geq 1$, $t, t_1, ..., t_p \in \mathcal{T}$).
Proposition 3.1. *The GP $W(t) := \mathbb{E}[Z(t)|Z(t_1), ..., Z(t_p)]$ (for $p \geq 1$, $t, t_1, ..., t_p \in \mathcal{T}$) is a finite rank GP over $\mathcal{T}$.*

Proof. Let us consider a Gaussian random vector of size p, $\mathbf{Z}_p := (Z(t_1), ..., Z(t_p))^\top$. Let us also denote $K = (k(t_i, t_j))_{1\leq i,j\leq p}$ the covariance matrix of the chosen design and $k_p(t) = (k(t_i, t))_{1\leq i\leq p}(t \in \mathcal{T})$. Then, assuming L is positive definite, for any $t \in \mathcal{T}$:

$$W(t) = k_p(t)^\top K^{-1}\mathbf{Z}_p \tag{7}$$

$$= k_p(t)^\top K^{-1/2} X_p \tag{8}$$

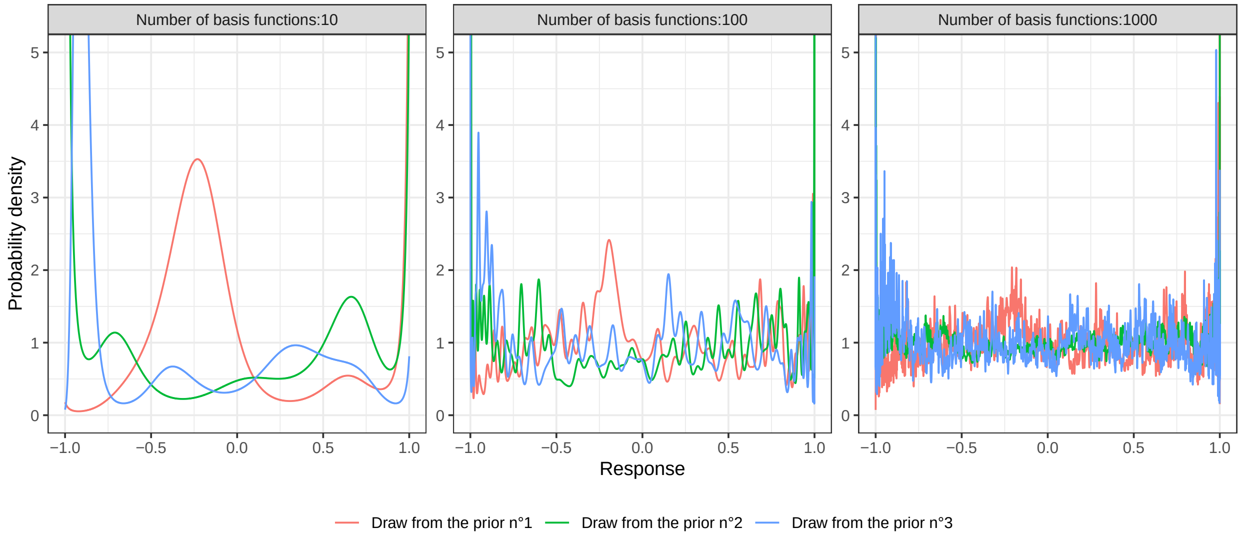

Figure 5: Three draws of a finite-rank SLGP indexed at the same value of $\mathbf{x}$, with basis functions being based on the Legendre Polynomials of order up to $p-1$.

where $\mathbf{X}_p = K^{-1/2}\mathbf{Z}_p$ is multivariate standard normal.

Therefore, setting $f_i(t)$ to be the i-th coordinate of the vector $k_p(t)^\top K^{-1/2}$ yields that $W(t) = \sum_{i=1}^p X_i f_i(t)$ where the X_i's are i.i.d. $\mathcal{N}(0,1)$. □

We implemented this approach to defining finite-rank SLGPs but we believe that conditioning GPs does not scale well, as the number of inducing points required would blow-up with dimensionality increase. Since the approach requires to compute the inverse square root of a covariance matrix, the numerical stability of the approach would suffer from an increase of the size of this matrix.

Instead, we leverage Fourier-type basis, i.e. collections of: $(\cos(\boldsymbol{\omega}^\top[\mathbf{x}, t]), \sin(\boldsymbol{\omega}^\top[\mathbf{x}, t]))$, for varying $\boldsymbol{\omega}$'s in $\mathbb{R}^{d_x+1}$. We implement different ways to select the angular frequencies $\boldsymbol{\omega}$. The first one being inspired from the discrete Fourier basis, while the next two are variations within the Random Fourier Feature framework. We also let users provide their custom set of frequencies to be used. Finally, to simplify notations we will mostly focus on the setting where $\mathcal{T} = [0,1]$.

Multivariate adaptation of the discrete Fourier basis For this approach, that is a multivariate extension of the unidimensional discrete Fourier basis, we consider that the domain D is $[0,1]^{d_x}$. We specify two parameters, noted $p_{\mathbf{x}}$ and p_t that determine the frequencies considered in $\mathbf{x}$ and t. Then, we consider all the $\boldsymbol{\omega} \in \left\{-\frac{2\pi(p_{\mathbf{x}}-1)}{p_{\mathbf{x}}}, ..., \frac{2\pi(p_{\mathbf{x}}-1)}{p_{\mathbf{x}}}\right\}^{d_x} \otimes \left\{2\pi, ..., \frac{2\pi(p_t-1)}{p_t}\right\}$. This approach yields $(2p_{\mathbf{x}}-1)^d(p_t-1)$ frequencies and therefore twice as many basis functions. This family of functions excludes non-positive frequencies in t for two reasons. First, excluding negative values allows for avoiding redundancy by ensuring that one can not have both $\boldsymbol{\omega}$ and$-\boldsymbol{\omega}$ as frequency. Second, we also avoid functions independent of t, as they would be cancelled-out within the normalisation step. When using a finite rank GP relying on these basis functions, we recommand to weight each function depending on its frequency, but to estimate only one hyperparameter: a common variance parameter. Indeed, one could imagine having one variance parameter per function, but it would lead to estimating $(2p_{\mathbf{x}}-1)^d(p_t-1)$ hyper-parameters, which would prove numerically costly.

Random Fourier Features The framework of Random Fourier Features (**??**) yields another way to construct finite rank GPs that “resemble” GPs with a prescribed (stationary and continuous) covariance kernel k. Indeed, any such kernel satisfy the hypothesis for the Bochner theorem, a result that relates positive definite, continuous, stationary kernels to the Fourier transform of a finite, positive measure. Whenever the considered measure admits a density $\mathfrak{s}(\boldsymbol{\omega})$ with respect to Lebesgue measure, this gives rise to the so-called Fourier duality of spectral densities and covariance functions. This result is also known as the

Wiener-Khinchin theorem (**?**):

$$k_0([\mathbf{x}, t] - [\mathbf{x}', t']) = \int_{R^d} s(\boldsymbol{\xi}) e^{i2\pi\boldsymbol{\xi}^\top [\mathbf{x}-\mathbf{x}', t-t']} \, d\boldsymbol{\xi} \tag{9}$$

$$s(\boldsymbol{\xi}) = \int_{R^d} k_0(\boldsymbol{r}) e^{-i2\pi\boldsymbol{\xi}^\top \boldsymbol{r}} \, d\boldsymbol{r} \tag{10}$$

The Wiener-Khinchin integral (Equation 9) can be re-written as:

$$k_0([\mathbf{x}, t] - [\mathbf{x}', t']) = \int_{R^d} s(\boldsymbol{\xi}) \left[\cos 2\pi\boldsymbol{\xi}^\top [\mathbf{x}, t] \cos 2\pi\boldsymbol{\xi}^\top [\mathbf{x}', t'] + \sin 2\pi\boldsymbol{\xi}^\top [\mathbf{x}, t] \sin 2\pi\boldsymbol{\xi}^\top [\mathbf{x}', t']\right] \, d\boldsymbol{\xi} \tag{11}$$

The essential element of the approach of Random Fourier Features (**??**) is the realisation that the Wiener-Khintchin integral (9) can be approximated by Monte Carlo sums.

Indeed, we can consider p draws $\boldsymbol{\xi}_i$ of independent random variables that have a density equal to the spectral density associated to k_0. We define a finite basis expansions as:

$$\boldsymbol{\varphi}([\mathbf{x}, t]) = \sqrt{\frac{1}{p}} \left[\cos(\boldsymbol{\xi}_1^\top [\mathbf{x}, t]), ..., \cos(\boldsymbol{\xi}_p^\top [\mathbf{x}, t]), \sin(\boldsymbol{\xi}_1^\top [\mathbf{x}, t]), ..., \sin(\boldsymbol{\xi}_p^\top [\mathbf{x}, t])\right] \tag{12}$$

Then, based on Equation 11, we have $\mathbb{E}_{\xi_1, ... \xi_p} \left[\boldsymbol{\varphi}([\mathbf{x}, t])^\top \boldsymbol{\varphi}([\mathbf{x}', t'])\right] = k_0([\mathbf{x}, t] - [\mathbf{x}', t'])$, which gives:

$$k([\mathbf{x}, t], [\mathbf{x}', t']) = \sigma^2 k_0([\mathbf{x}, t] - [\mathbf{x}', t']) \approx k_{RFF}([\mathbf{x}, t], [\mathbf{x}', t']) := \sigma^2 \boldsymbol{\varphi}([\mathbf{x}, t])^\top \boldsymbol{\varphi}([\mathbf{x}', t']) \tag{13}$$

Moreover, for $\boldsymbol{\varepsilon}$ a $2p$-variate standard normal vector, the process:

$$Z_{RFF, \mathbf{x}, t} = \sigma \boldsymbol{\varphi}([\mathbf{x}, t])^\top \boldsymbol{\varepsilon} \tag{14}$$

is a GP with mean zero and covariance kernel $k_{RFF}([\mathbf{x}, t], [\mathbf{x}', t'])$.

Remark. In the literature it is common to encounter another basis expansion. Indeed, it is also possible to samples p random phases $U_i \sim \mathcal{U}([0, 2\pi])$ and to use a basis using only cosines:

$$\tilde{\varphi}(\mathbf{y}) = \sqrt{\frac{2\pi}{p}} \left[\cos(\boldsymbol{\xi}_1^\top \mathbf{y} + U_1), \cos(\boldsymbol{\xi}_2^\top \mathbf{y} + U_2), ..., \cos(\boldsymbol{\xi}_p^\top \mathbf{y} + U_p)\right], \tag{15}$$

Philosophically, there is no fundamental difference between these two basis, since $\sin(t + \pi/2) = \cos(t)$. Here, selected the parametrisation in sines/cosines due to its improved behaviour compared to collections of the type $(\cos(\boldsymbol{\xi}^\top \mathbf{y} + U)$ (**?**).

Space-filling Random Fourier Features Building on the RFF framework, we propose selecting the frequencies $\boldsymbol{\xi}_i$ using a space-filling design with respect to the spectral density, rather than relying on purely random sampling. This approach yields a Fourier-type feature family whose frequencies are more evenly distributed across the spectral domain, potentially avoiding overlap between considered basis functions.

3.2 Estimation schemes considered and implemented

Once a family of basis function is selected, we need to select a suitable estimation paradigm. We propose and implemented three prominent estimation strategies, each with its unique merits and suitability for different scenarios within the context of our model. These methods' sanity are underlined by the following result:

Theorem 3.2. *For fixed data* $\mathbf{t}$ *the likelihood function* $\boldsymbol{\epsilon} \mapsto \mathcal{L}(\boldsymbol{\epsilon}|\mathbf{t}) = \prod_{j=1}^{n} \frac{e^{\sum_{i=1}^{p} \epsilon_j f_i(\mathbf{x}_j, t_j)}}{\int_{\mathcal{T}} e^{\sum_{i=1}^{p} \epsilon_i f_i(\mathbf{x}_j, u)} \, du}$ *is log-concave.*

Equivalently, the negative log-likelihood function $\boldsymbol{\epsilon} \mapsto \ell(\boldsymbol{\epsilon}|\mathbf{t})$ *is convex.*

Proof. To prove this statement, we compute the gradient and Hessian of the negative log likelihood. The negative log-likelihood is:

$$\ell(\boldsymbol{\epsilon}|\mathbf{t}) = \sum_{j=1}^{n} \left(-\sum_{i=1}^{p} f_i(\mathbf{x}_j, t_j)\epsilon_i + \log\left(\int_{\mathcal{T}} e^{\sum_{i=1}^{p} f_i(\mathbf{x}_j, u)\epsilon_i} \, du \right) \right) \tag{16}$$

Its gradients write:

$$\frac{\partial \ell(\boldsymbol{\epsilon}|\mathbf{t})}{\partial \epsilon_i} = -\sum_{j=1}^{n} f_i(\mathbf{x}_j, t_j) + \sum_{j=1}^{n} \int_{\mathcal{T}} f_i(\mathbf{x}_j, u) \frac{e^{\boldsymbol{\epsilon}^T F(\mathbf{x}_k, u)}}{\int_{\mathcal{T}} e^{\boldsymbol{\epsilon}^T F(\mathbf{x}_k, v)} \, dv} \, du \tag{17}$$

The second order derivatives are:

$$\begin{aligned} \frac{\partial^2 \ell(\boldsymbol{\epsilon}|\mathbf{t})}{\partial \epsilon_i \partial \epsilon_{i'}} &= \sum_{j=1}^{n} \int_{\mathcal{T}} f_i(\mathbf{x}_j, u) f_{i'}(\mathbf{x}_j, u) \frac{e^{\boldsymbol{\epsilon}^T F(\mathbf{x}_j, u)}}{\int_{\mathcal{T}} e^{\boldsymbol{\epsilon}^T F(\mathbf{x}_j, v)} \, dv} \, du \\ - \sum_{j=1}^{n} &\left(\int_{\mathcal{T}} f_i(\mathbf{x}_j, u) \frac{e^{\boldsymbol{\epsilon}^T F(\mathbf{x}_j, u)}}{\int_{\mathcal{T}} e^{\boldsymbol{\epsilon}^T F(\mathbf{x}_j, v)} \, dv} \, du \right) \left(\int_{\mathcal{T}} f_{i'}(\mathbf{x}_j, u) \frac{e^{\boldsymbol{\epsilon}^T F(\mathbf{x}_j, u)}}{\int_{\mathcal{T}} e^{\boldsymbol{\epsilon}^T F(\mathbf{x}_j, v)} \, dv} \, du \right) \end{aligned} \tag{18}$$

First, observe that only the first term in Equation 17 depends on the values of the t_is, and that all the other terms can be computed only once per distinct value of the $\mathbf{x}_i$s. Let us assume there are $K \leq n$ unique values of the $\mathbf{x}_i$s. We denote $\tilde{\mathbf{x}}_1, ..., \tilde{\mathbf{x}}_K$ these values, and assume that $\tilde{\mathbf{x}}_k$ appears n_k times in $\{\mathbf{x}_i\}_{1 \leq i \leq n}$. Then:

$$\frac{\partial \ell(\boldsymbol{\epsilon}|\mathbf{t})}{\partial \epsilon_i} = -\sum_{j=1}^{n} f_i(\mathbf{x}_j, t_j) + \sum_{k=1}^{K} n_k \int_{\mathcal{T}} f_i(\tilde{\mathbf{x}}_k, u) \frac{e^{\boldsymbol{\epsilon}^T F(\tilde{\mathbf{x}}_k, u)}}{\int_{\mathcal{T}} e^{\boldsymbol{\epsilon}^T F(\tilde{\mathbf{x}}_k, v)} \, dv} \, du \tag{19}$$

and

$$\begin{aligned} \frac{\partial^2 \ell(\boldsymbol{\epsilon}|\mathbf{t})}{\partial \epsilon_i \partial \epsilon_{i'}} &= \sum_{k=1}^{K} n_k \int_{\mathcal{T}} f_i(\tilde{\mathbf{x}}_k, u) f_{i'}(\tilde{\mathbf{x}}_k, u) \frac{e^{\boldsymbol{\epsilon}^T F(\tilde{\mathbf{x}}_k, u)}}{\int_{\mathcal{T}} e^{\boldsymbol{\epsilon}^T F(\tilde{\mathbf{x}}_k, v)} \, dv} \, du \\ - \sum_{k=1}^{K} n_k &\left(\int_{\mathcal{T}} f_i(\tilde{\mathbf{x}}_k, u) \frac{e^{\boldsymbol{\epsilon}^T F(\tilde{\mathbf{x}}_k, u)}}{\int_{\mathcal{T}} e^{\boldsymbol{\epsilon}^T F(\tilde{\mathbf{x}}_k, v)} \, dv} \, du \right) \left(\int_{\mathcal{T}} f_{i'}(\tilde{\mathbf{x}}_k, u) \frac{e^{\boldsymbol{\epsilon}^T F(\tilde{\mathbf{x}}_k, u)}}{\int_{\mathcal{T}} e^{\boldsymbol{\epsilon}^T F(\tilde{\mathbf{x}}_k, v)} \, dv} \, du \right) \end{aligned} \tag{20}$$

For a given $\boldsymbol{\epsilon}$, let us introduce $Y_1, ...Y_K$, K independent random variables where each of the Y_i has a probability density of $\frac{e^{\boldsymbol{\epsilon}^T F(\tilde{\mathbf{x}}_i, u)}}{\int_{\mathcal{T}} e^{\boldsymbol{\epsilon}^T F(\tilde{\mathbf{x}}_i, v)} \, dv}$. Then:

$$\begin{aligned} \frac{\partial^2 \ell(\boldsymbol{\epsilon}|\mathbf{t})}{\partial \epsilon_i \partial \epsilon_{i'}} &= \sum_{k=1}^{K} n_k \left(\mathbb{E}\left[f_i(\tilde{\mathbf{x}}_k, Y_i) f_{i'}(\tilde{\mathbf{x}}_k, Y_{i'}) \right] - \mathbb{E}\left[f_i(\tilde{\mathbf{x}}_k, Y_i) \right] \mathbb{E}\left[f_{i'}(\tilde{\mathbf{x}}_k, Y_{i'}) \right] \right) \\ &= \sum_{k=1}^{K} n_k \mathrm{Cov}\left(f_i(\tilde{\mathbf{x}}_k, Y_i), f_{i'}(\tilde{\mathbf{x}}_k, Y_{i'}) \right) \end{aligned} \tag{21}$$

Since the Hessian matrix is a (sum of) covariance matrices, it inherits their symmetry and positive-definiteness. This suffices to prove that the negative log-likelihood is convex, and therefore that the likelihood is log-concave. □

In light of this result, we propose three approaches.

Markov Chain Monte Carlo (MCMC) This method allows us to explore the posterior distribution by drawing samples from it. It enables precise, exact inference of the posterior distribution of $\boldsymbol{\varepsilon}$ knowing the data. The main drawback of this approach is typically the need to tune hyper-parameters of the MCMC estimation scheme and carefully monitor outputs to ensure correct mixing in the chain, hence resulting in a generally high computational cost, due to the need to evaluate many instances of the posterior in Equation 6.

Fortunately, our scenario benefits from the posterior being a log-concave function of $\boldsymbol{\varepsilon}$, which ensures a rapid convergence of the considered algorithms (**?**). To address the tuning challenge, our implementation leverages the RStan package, our implementation relies on the RStan package (**?**) for MCMC sampling, utilizing the No-U-Turn Sampler (NUTS) algorithm (**?**). NUTS is designed to efficiently sample from complex posterior distributions without requiring manual adjustment of sampling parameters. This innovation significantly reduces the computational burden for practitioners. However, it still incurs costs associated with gradient evaluations necessary for navigating the posterior landscape. This lead us to also propose other, faster inference techniques.

The MCMC returns N draws $(\boldsymbol{\epsilon}_i)_{1 \leq i \leq N}$ from the posterior in Equation 6. This, in turns yields a probabilistic prediction of the density field, since we have N draws of the SLGP:

$$\hat{Y}_{\mathbf{x},t}^{\text{MCMC},i} = \frac{e^{\sigma \sum_{j=1}^{p} f_j(x_1,\dots,x_d,t)\epsilon_{i,j}}}{\int_{\mathcal{T}} e^{\sigma \sum_{j=1}^{p} f_j(x_1,\dots,x_d,u)\epsilon_{i,j}} \, du} \tag{22}$$

Maximum A Posteriori (MAP) For a quicker approximation, MAP estimation provides a point estimate, offering a balance between computational efficiency and the depth of inference. It is the fastest estimation scheme we propose, however MAP does not facilitate uncertainty quantification because it yields a non-probabilistic estimate of $\boldsymbol{\varepsilon}$, focusing instead on identifying the mode of the posterior distribution. For consistency with the previous implementation, we also perform this optimisation through the RStan package (**?**), and use straightforward yet effective Newton's method, enjoying to the fullest our knowledge of the posterior's properties.

The MAP approach returns the mode $\boldsymbol{\epsilon}^*$ from the posterior in Equation 6. This in turns yields a prediction of the density field:

$$\hat{Y}_{\mathbf{x},t}^{\text{MAP}} = \frac{e^{\sigma \sum_{j=1}^{p} f_j(x_1,\dots,x_d,t)\epsilon_j^*}}{\int_{\mathcal{T}} e^{\sigma \sum_{j=1}^{p} f_j(x_1,\dots,x_d,u)\epsilon_j^*} \, du} \tag{23}$$

MAP coupled with Laplace approximation By integrating the MAP approach with Laplace approximation, we enhance our estimation strategy to approximate the posterior distribution using a multivariate Gaussian model. This combination offers a nuanced middle ground between the exact Bayesian inference provided by MCMC and the rapid estimation capabilities of MAP. It achieves this by utilizing an approximation that leverages both the gradient and Hessian of the actual posterior, thereby informing the approximation with critical insights into the shape and curvature of the posterior landscape.

The Laplace approximation returns N draws $(\boldsymbol{\epsilon}_i)_{1 \leq i \leq N}$ from the multivariate normal distribution centred at the mode of the posterior in Equation 6, and precision matrix being the observed Fisher information of this same posterior. This in turns yields a probabilistic prediction of the density field, since we have N draws of the SLGP:

$$\hat{Y}_{\mathbf{x},t}^{\text{Laplace},i} = \frac{e^{\sigma \sum_{j=1}^{p} f_j(x_1,\dots,x_d,t)\epsilon_{i,j}}}{\int_{\mathcal{T}} e^{\sigma \sum_{j=1}^{p} f_j(x_1,\dots,x_d,u)\epsilon_{i,j}} \, du} \tag{24}$$

We summarise in Table 1 the main advantages and limitations of all three methods, to help practitioners select the one most suitable to their needs. We also point out here that in most use cases, MAP with Laplace proves to be a reasonable approach, with good capacities for uncertainty quantification, and a notable speed-up compared to MCMC.

	MCMC	**MAP**	**MAP with Laplace**
Probabilistic predictions	✓ Exact	✗ None	✓ With normal approximation
Estimation speed	Slowest	Fastest	Fast
Suitable for large p	✓	✓	✗ Needs to invert Hessian
Typical use case	For uncertainty quantification and exact inference.	Get a point estimate quickly. Get a start point for MCMC.	Compromise between the other two methods.

Table 1: Summarising the specifics and suggested use cases of each estimation method.

3.3 Approximation paradigms for the integrals in the SLGP

In all the aforementioned estimation schemes, inferring the posterior distribution necessitates the computation of multiple integrals as delineated by Equation 6. These integrals, essential in obtaining the posterior of $\boldsymbol{\varepsilon}$ given the data, present a computational challenge. To address this, we introduce a numerical quadrature approach that approximates these integrals. By utilizing the finite rank representation, we are able to pre-calculate the basis functions at designated nodes. This preparation streamlines the computational process, allowing for rapid substitution of $\boldsymbol{\varepsilon}$ values and subsequent calculation of the requisite integrals.

For the sake of brevity and given that the multivariate normal distribution is well known, we choose to center the remaining of our discussion on the intricacies of the likelihood function. The likelihood, which underpins the posterior distribution, is:

$$\mathcal{L}(\boldsymbol{\epsilon}|D) = \prod_{i=1}^{n} \frac{\exp\{\boldsymbol{\epsilon}^\top \boldsymbol{F}(\mathbf{x}_i, t_i)\}}{\int_0^1 \exp\{\boldsymbol{\epsilon}^\top \boldsymbol{F}(\mathbf{x}_i, u)\}\, du} \tag{25}$$

Assuming that $(u_i)_{i=1}^m$ are nodes of a regular grid of $[0,1]$, our proposed quadrature is the following:

$$\mathcal{L}(\boldsymbol{\epsilon}|D) \approx \prod_{i=1}^{n} \frac{\exp\{\boldsymbol{\epsilon}^\top \boldsymbol{F}(\mathbf{x}_i, t_i)\}}{\frac{1}{m}\sum_{j=1}^{m} \exp\{\boldsymbol{\epsilon}^\top \boldsymbol{F}(\mathbf{x}_i, u_j)\}} \tag{26}$$

A first step in ensuring numerical stability of our implementation is to work with the negative log-likelihood rather than with the likelihood itself, as is standard in Bayesian inference. Following our approach and notations from the proof of Theorem 3.2, we need to evaluate:

$$\ell(\boldsymbol{\epsilon}|\mathbf{t}) \approx -\boldsymbol{\epsilon}^\top \left(\sum_{i=1}^{n} \boldsymbol{F}(x_i, t_i) \right) + \sum_{k=1}^{K} n_k \log \left(\frac{1}{m} \sum_{j=1}^{m} \exp\{\boldsymbol{\epsilon}^\top \boldsymbol{F}(\tilde{\mathbf{x}}_k, u_j)\} \right) \tag{27}$$

Evaluating each of the K terms in the second sum can be computationally expensive, particularly when dealing with large datasets. To balance accuracy and efficiency, we propose three different implementation strategies:

- **Exact evaluation:** Computes the full sum $\frac{1}{m}\sum_{j=1}^{m} \exp\{\boldsymbol{\epsilon}^\top \boldsymbol{F}(\tilde{\mathbf{x}}_k, u_j)\}$ for each instance of $\tilde{\mathbf{x}}_k$, ensuring the highest accuracy but at the cost of increased computational time.
- **Nearest neighbor approximation:** Discretised the space of $\mathbf{x}$ and for each $\tilde{\mathbf{x}}_k$, uses the integral value from the closest precomputed node. This is the fastest method but introduces the largest approximation error.

- **Interpolated approximation (recommended):** Discretised the space of $\mathbf{x}$ and for each $\tilde{\mathbf{x}}_k$, interpolates the values at the vertices of the hypercube surrounding it. This provides a good trade-off between accuracy and computational efficiency.

These three approaches allow users to adapt computations to their needs, depending on whether precision or speed is the priority. For more details regarding these approximation schemes, we refer the reader to Appendix A.

	No approximation	**NN approximation**	**WNN approximation**
Estimation quality	✓ Exact	✗ Piecewise constant	$\sim$ Interpolation
Estimation speed	Slowest	Fastest	Fast
Typical use case	With few different indices $\mathbf{x}$	To get the fastest estimate	Compromise between the other two methods (recommended).

Table 2: Summarising the specifics and suggested use cases of each integral approximation scheme.

3.4 Tackling the hyper-parameters in SLGP estimation.

Yet another problem posed is that in almost all realistic cases, GPs depend on some unknown hyperparameters that need to be estimated. This issue could be addressed in a Bayesian way, by specifying a prior on the hyperparameters. While a principled approach to selecting suitable prior have been proposed and studied, (**??**), these generally do not allow for Theorem 3.2 to hold. As this log-concavity is crucial to the efficiency of our estimation procedure, we rather focus on heuristics and simple estimation procedures.

Assuming that the GP is parametrised by a variance parameter σ^2 as well as $d+1$ length-scale parameters $\rho_t, \rho_1, ..., \rho_d$ through:

$$Z_{\mathbf{x},t} = \sigma \sum_{j=1}^{p} f_j(x_1/\rho_1, ..., x_d/\rho_d, t/\rho_t)\varepsilon_j, \ \forall \mathbf{x} \in D, \ t \in \mathcal{T} \tag{28}$$

we propose using a heuristic for σ^2 and either another heuristic or a grid search for the ρ's.

Variance heuristic: This part consists in numerically selecting a variance that ensures numerical stability of the SLGP. In our case, we typically select σ^2 such that $\mathbb{E}[\max_{\mathbf{x}\in D}(\max_{t\in\mathcal{T}} Z_{\mathbf{x},t} - \min_{t\in\mathcal{T}} Z_{\mathbf{x},t})] \leq 5$. This heuristic controls the range of values that the SLGP can take, typically restricting it to $[0, e^5 \approx 148]$. This step is not performed conditionally on any data, and is here to ensure that we specified a model that does not present numeric overflow.

Motivating examples underlying this heuristic are available in Figure 6, where the impact of σ on the prior's expressiveness and stability can be visualised. Smaller values of σ yield distribution fields that resemble the uniform, while the larger values produce sharply peaked fields that might produce numerical instabilities. It appears that ensuring $\max_{\mathbf{x}\in D}(\max_{t\in\mathcal{T}} Z_{\mathbf{x},t} - \min_{t\in\mathcal{T}} Z_{\mathbf{x},t}) \approx 5$ yields a compromise between expressiveness and stability.

Length-scale selection: As for the length-scales, we propose two approaches: a heuristic one, motivated by extensive model fitting over varied datasets, and MAP approach.

The former is motivated by the fact that in practice, and especially in small data regime, a useful rule of thumb is to set each length-scale to about 10–15% of the corresponding domain size.

Otherwise, a grid search over candidate values, followed by selection based on the marginal posterior is advised. To this end, we specify a prior π over $\boldsymbol{\rho}$, and focus on the joint posterior:

$$\pi(\boldsymbol{\epsilon}; \boldsymbol{\rho}|\mathbf{T}_n = \mathbf{t}_n) \propto \pi(\boldsymbol{\rho})\phi(\boldsymbol{\epsilon}) \prod_{i=1}^{n} \frac{e^{\sigma \sum_{j=1}^{p} f_j(x_{1,i}/\rho_1, ..., x_{d,i}/\rho_{d_x}, t_i/\rho_{d_x+1})\epsilon_j}}{\int_{\mathcal{T}} e^{\sigma \sum_{j=1}^{p} f_j(x_{1,i}/\rho_1, ..., x_{d,i}/\rho_{d_x}, u/\rho_{d_x+1})\epsilon_j} du} \tag{29}$$

where, in light of the results from **?**, we suggest an inverse Gamma prior for $\boldsymbol{\rho}$.

Optimising the joint prior for a fixed $\boldsymbol{\rho}$ is a convex problem that yields a unique optimal predictor $\boldsymbol{\epsilon}^*(\boldsymbol{\rho}) = \arg\max_{\boldsymbol{\epsilon}\in\mathbb{R}^p} \pi(\boldsymbol{\epsilon}; \boldsymbol{\rho}|\mathbf{T}_n = \mathbf{t}_n)$. We then identify the optimal length-scales as the maximisers of the profile posterior: $\boldsymbol{\rho}^* \in \arg\max_{\boldsymbol{\rho}\in\mathbb{R}^{d_x+1}} \pi(\boldsymbol{\epsilon}^*(\boldsymbol{\rho}); \boldsymbol{\rho}|\mathbf{T}_n = \mathbf{t}_n)$. In practice, this outer optimisation over $\boldsymbol{\rho}$ is carried out via a simple grid search.

We further illustrate the impact of length-scale selection in the Appendix B.

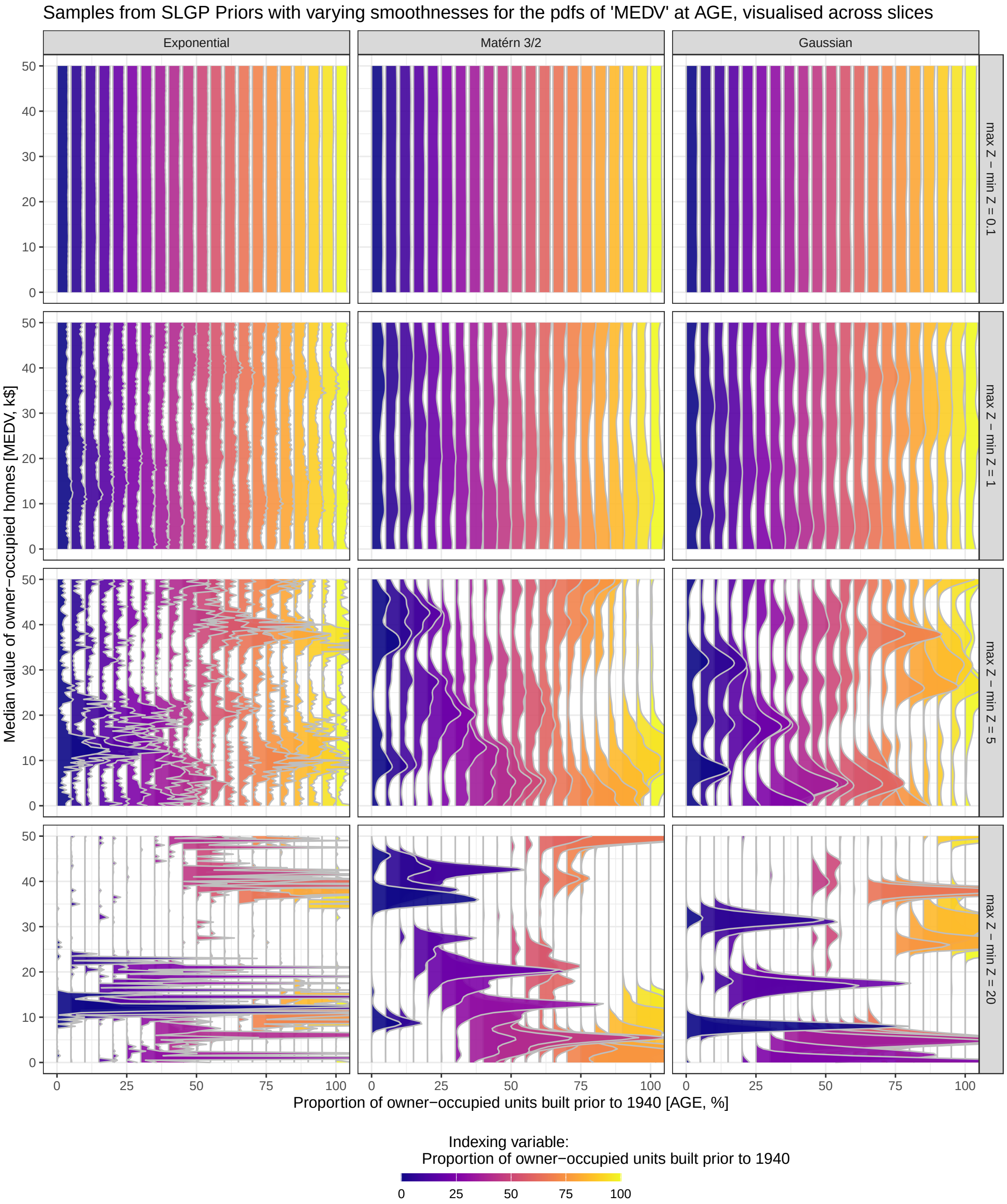

Figure 6: Four realisations of SLGPs, with varying ranges of values of the underlying GP. The corresponding GP's realisations can be seen in Appendix, Figure A-5.

4 Guidelines for the users, application to the Boston Housing dataset

All subsequent codes are executed relying on the R SLGP package (?), and a detailed markdown document to reproduced all the Figures and numerical development in this document is available on the author's Github page (?).

4.1 Density field estimation with a SLGP

4.1.1 Prior induced by a SLGP model

To build intuition about the behavior of a SLGP prior, we begin by visualising some realisations. Specifically, we approximate a Matérn 5/2 kernel using 200 sine and 200 cosine basis functions in a RFF framework. The length-scales are set to 15% of the range of the inputs, ensuring a reasonable degree of smoothness while capturing variability across the domain. The variance is automatically determined based on the heuristic we proposed earlier.

Below is an R code snippet illustrating how to generate such a prior, followed by the resulting illustration in Figure 7:

```r
library(SLGP)

modelPrior <- slgp(medv~age, # Use a formula to specify predictors VS response
                   data=df,
                   method="none", # SLGP prior
                   basisFunctionsUsed = "RFF",
                   hyperparams = list(lengthscale=c(0.15, 0.15), sigma2=1),
                   # Applied to normalised data, so 15% of the range of values
                   # Sigma will be re-selected with sigmaEstimationMethod
                   sigmaEstimationMethod = "heuristic",
                   predictorsLower= c(range_x[1]),
                   predictorsUpper= c(range_x[2]),
                   responseRange= range_response,
                   opts_BasisFun = list(nFreq=200, MatParam=5/2), seed=1)
```

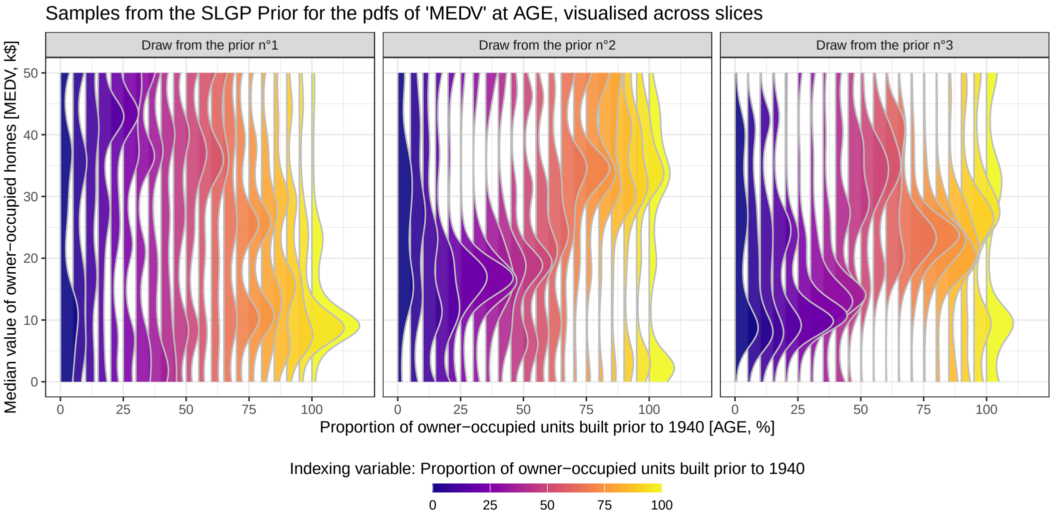

Figure 7: Three draws from the SLGP prior using RFF with a Matérn 5/2 kernel.

Representing the pdfs along slices as we do allows for visualising the changes in shape and modalities in the generated field, by displaying pdfs generated for a SLGP for various values of $\mathbf{x}$. For instance, the third SLGP draw that is represented in Figure 7 displays right-skewed unimodal distributions for $\mathbf{x} =$ age close to 0, flatter pdfs with most of the probability mass around medv $= 35$ for $\mathbf{x}$ close to 50%, a sharper uni-modal distributions that places most of its probability mass around `medv` $= 25$ for `x` close to 75%, then bi-modal distributions with a mode around `medv` $= 10$ and another around `medv` $= 32$ for `x` close to 100%. These changes give encouraging insight into the SLGP's abilities to model a wide range of distributions. Other draws of SLGP's prior exhibit different changes in shape or modalities, illustrating the flexibility of our model.

Since the GP used in this construction is centered, the corresponding SLGP prior has a uniform mean. If a different mean behavior is desired (e.g., to reflect expert knowledge or known trends), practitioners can incorporate this by modifying the mean function of the underlying GP.

We also chose the Matérn 5/2 kernel to strike a balance between smoothness and variability. Other kernels induce different degrees of regularity, which influence the types of densities favored by the SLGP. This is illustrated in Figure 8.

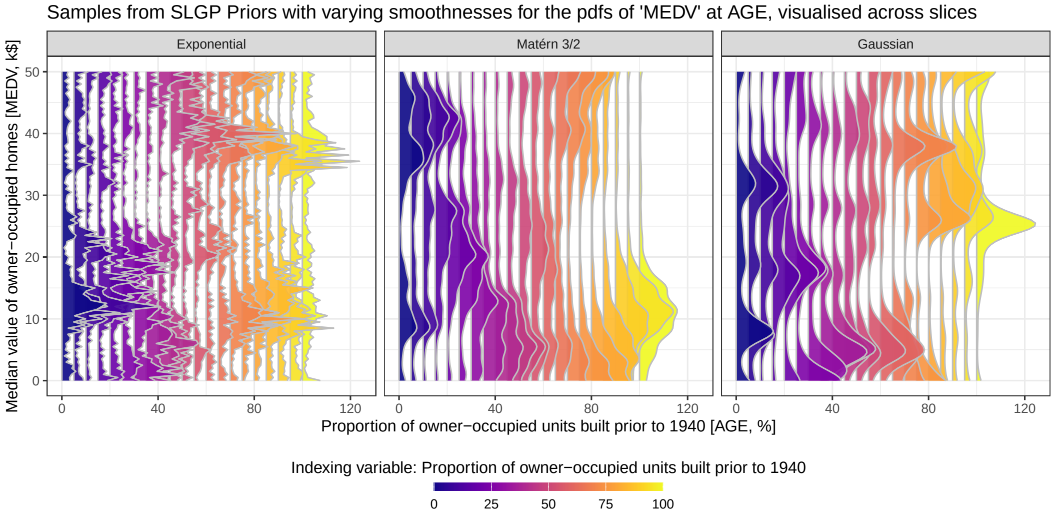

Figure 8: Three draws of SLGPs induced by GPs with various kernels (and thus, spatial regularity).

4.1.2 Maximum a Posteriori estimation

Now that the prior has been defined, we can proceed with estimating the spatially varying density field. The fastest estimation scheme provided in our package is the MAP estimation procedure. MAP estimation yields a single point estimate. However, this comes at the cost of limited inferential depth: MAP does not support uncertainty quantification, as it focuses solely on identifying the mode of the posterior distribution rather than characterizing its full shape.

The following code demonstrates how to train an SLGP model using the MAP scheme:

```
modelMAP <- slgp(medv~age, # Use a formula to specify predictors VS response
                 data=df,
                 method="MAP", #Maximum a posteriori estimation scheme
                 basisFunctionsUsed = "RFF",
                 interpolateBasisFun="WNN", # Accelerate inference
                 hyperparams = list(lengthscale=c(0.15, 0.15), sigma2=1),
                 sigmaEstimationMethod = "heuristic",
                 predictorsLower= c(range_x[1]),
                 predictorsUpper= c(range_x[2]),
                 responseRange= range_response,
                 opts_BasisFun = list(nFreq=200, MatParam=5/2), seed=1)

# Or re-train the model we generated earlier
modelMAP <- retrainSLGP(SLGPmodel=modelPrior, newdata = df, method="MAP")

# Predict the values to prepare the Figure
```

```
dfGrid <- data.frame(expand.grid(seq(range_x[1], range_x[2],, 101),
                                 seq(range_response[1], range_response[2], 1)))
colnames(dfGrid) <- c("age", "medv")
pred <- predictSLGP_newNode(SLGPmodel=modelMAP, newNodes = dfGrid)
```

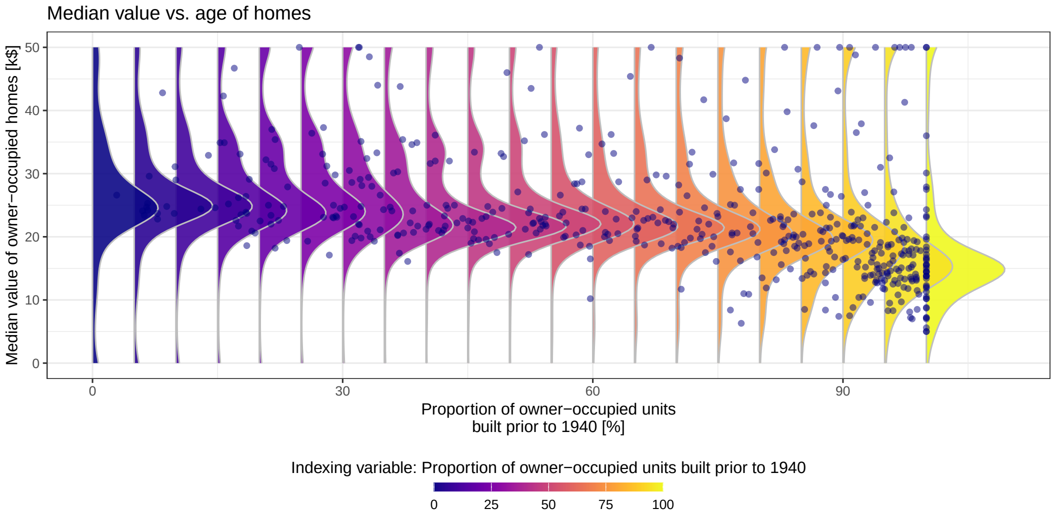

Figure 9: MAP-based SLGP predictive densities for `medv` across values of `age` .

Figure 9 presents the predictive densities produced by the SLGP model, based on a heuristic length-scale set to 15% of the input range. In Appendix B, we explore a more principled approach by optimizing this length-scale through grid search, under an inverse gamma prior.

While MAP inference provides only a single estimate, its outputs remain interpretable and useful. Figure 10 compares the MAP-predicted density slices to empirical histograms computed from the raw data. Note that this comparison is provided for illustrative purposes only: one of the key motivations behind the development of SLGP modeling is to avoid the need for arbitrary binning of data prior to density estimation.

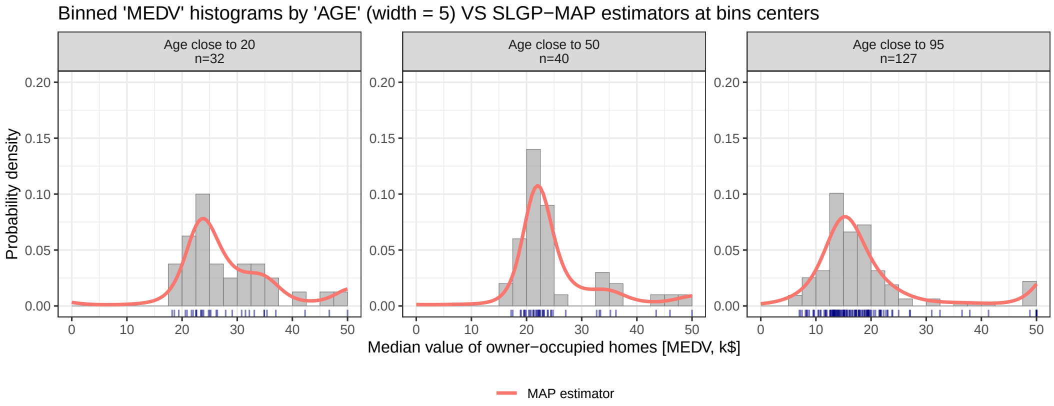

Figure 10: SLGP MAP estimates versus empirical histograms for binned `medv` data at several `age` slices.

It is nonetheless encouraging to observe that the SLGP model succeeds in reproducing several key features of the data, effectively capturing both multi-modal behavior and the increase density concentration for high values of the conditioning variable (`age` close to 95). This illustrates the effectiveness of SLGPs even under a fast, point-estimate-based inference strategy.

We can also extract summary statistics from the estimated SLGP, such as conditional quantiles, as shown in Figure 11. In this figure, we conduct a sanity check by comparing SLGP-based quantile estimates with those obtained using classical quantile regression (?), as implemented in the R package quantreg (?).

```
probsL <- c(5, 25, 50, 75, 95)/100
dfX <- data.frame(age=seq(range_x[1], range_x[2], 1))
pred <- predictSLGP_quantiles(SLGPmodel=modelMAP, newNodes = dfX, probs=probsL)
```

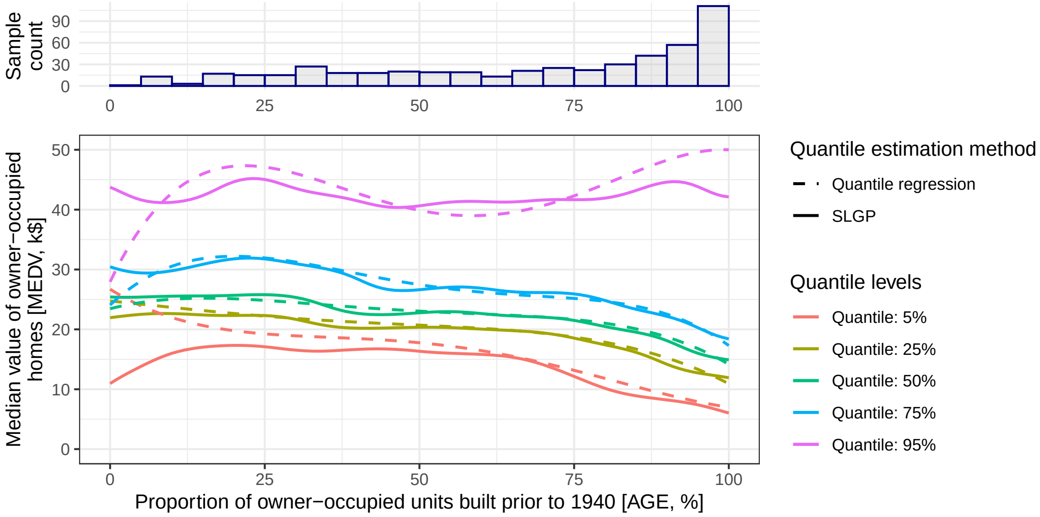

Figure 11: [Top] Histogram of the number of observations across age intervals. [Bottom] Simultaneous quantile estimates from SLGP (MAP-based) and classical quantile regression.

The two methods are broadly consistent, especially in regions of the input space (age) with dense data coverage. One notable advantage of SLGP is its ability to avoid quantile crossing, a pathological behavior visible in the quantile regression curves in regions of data sparsity (e.g., age ¡ 10).

A visible divergence between both approaches appears for the 95% quantile when age exceeds 75. To assess this, we compare both models to empirical quantiles computed over bins of width 5 (Figure 12). In this high-age region, the quantile regression curve tends to overshoot, while the SLGP estimates remain better aligned with the observed variability, suggesting the SLGP's prediction is more faithful to the underlying data.

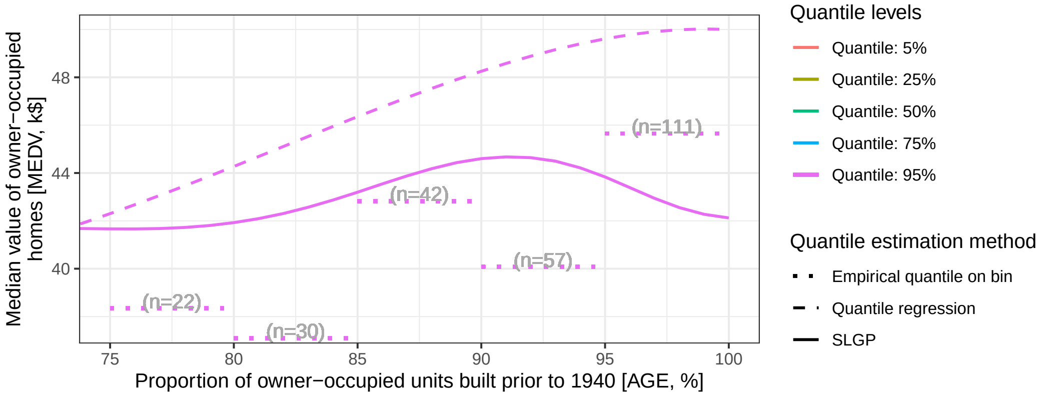

Figure 12: Comparison of conditional quantile estimates obtained via SLGP (MAP-based), classical quantile regression, and empirical quantile over arbitrary bins for the 95% quantile when age exceeds 75.

4.1.3 Uncertainty quantification: Laplace approximation and Markov Chain Monte Carlo-based estimation

We do not pursue more quantitative comparisons here and instead rely on these visual checks as qualitative sanity checks to assess the model's adequacy.

To enable uncertainty quantification, we must move beyond MAP estimation. We provide two alternatives: Laplace approximation and MCMC. Both approaches yield probabilistic predictions.

The Laplace approximation offers a fast and tractable method for capturing posterior uncertainty. Figure 13 shows posterior draws obtained from the Laplace approximation overlaid with data histograms, showcasing the model's ability to capture both variability and uncertainty.

The following code demonstrates how to re-train our SLGP model using the Laplace estimation-based scheme:

```
modelLaplace <- retrainSLGP(SLGPmodel=modelPrior, newdata = df, method="Laplace")
```

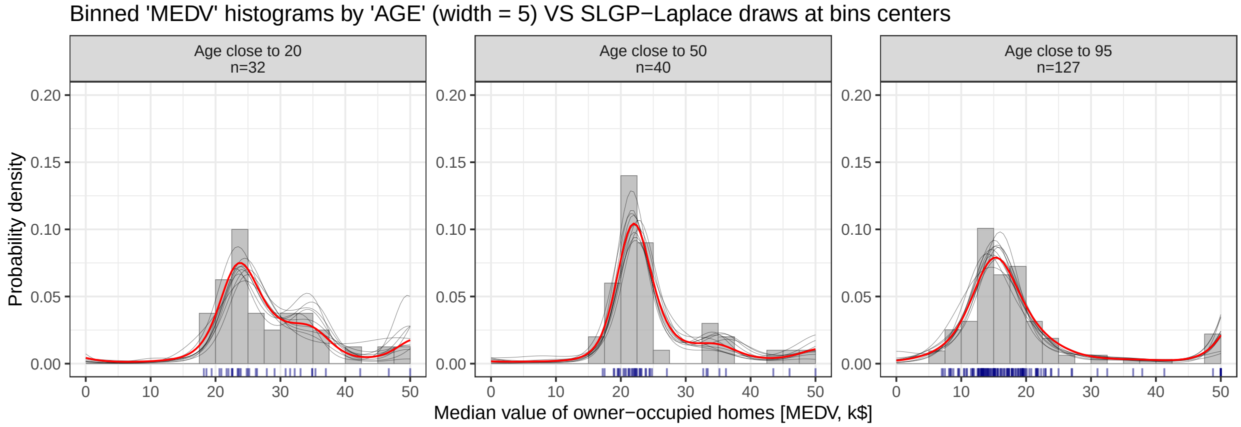

Figure 13: Predictive probability density (Laplace approximation) of `medv` at selected `age` values, with posterior draws superimposed.

Using MCMC provides access to full Bayesian inference, offering posterior samples from the distribution of the SLGP model. This is the approach previously used in the paper to produce Figure 4. The following code demonstrates how to re-train our SLGP model using the MCMC estimation-based scheme:

```
modelMCMC  <- retrainSLGP(SLGPmodel=modelPrior, newdata = df, method="MCMC",
                          opts = list(stan_chains=2, stan_iter=1000))
```

In addition to posterior draws, we also estimate conditional quantiles using MCMC (Figure 14). These can be compared to the quantile regression baseline as before.

```
probsL <- c(5, 25, 50, 75, 95)/100
pred <- predictSLGP_quantiles(SLGPmodel=modelMCMC, newNodes = dfX, probs=probsL)
```

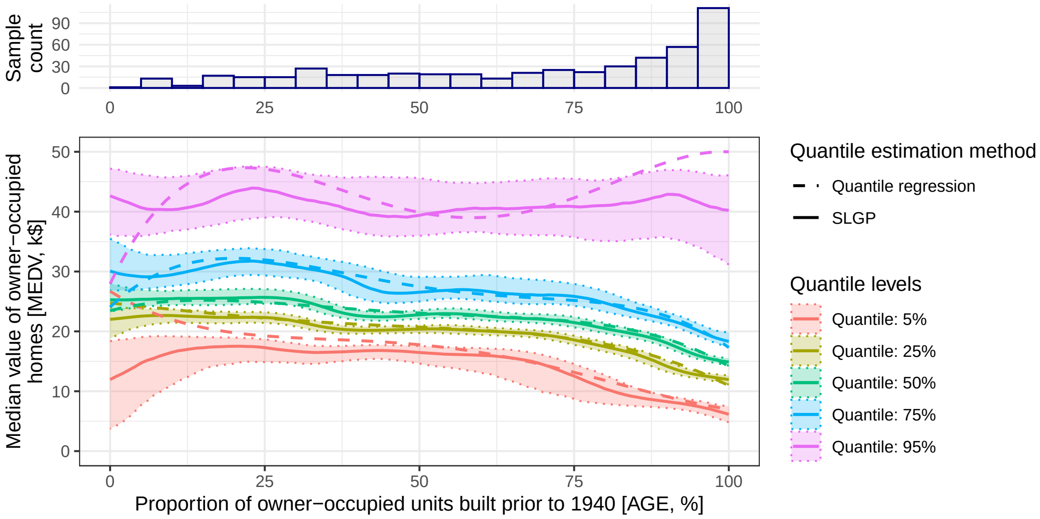

Figure 14: [Top] Histogram of the number of observations across intervals of `age` . [Bottom] Simultaneous quantile predictions using the SLGP model estimated via MCMC, compared to standard quantile regression.

Finally, we demonstrate the interpretability of the SLGP framework by estimating posterior moments (mean, standard deviation, skewness, and kurtosis) of the conditional densities using MCMC samples (Figure 15). These higher-order statistics provide a concise yet informative summary of the predictive distributions and further illustrate the variety of by-products yielded by our model.

```
dfX <- data.frame(age=seq(range_x[1], range_x[2], 1))

predMean <- predictSLGP_moments(SLGPmodel=modelMCMC,
                                newNodes = dfX, power=c(1),centered=FALSE)
                                # Uncentered moments for the mean
predVar <- predictSLGP_moments(SLGPmodel=modelMCMC,
                               newNodes = dfX, power=c(2, 3, 4), centered=T)
                               # For the Variance, Kurtosis and Skewness

pred <- rbind(predMean, predVar)
pred <- pred %>%
  pivot_longer(-c("age", "power"))%>%
  mutate(value=ifelse(power==2, sqrt(value), value))%>% # Define std
  pivot_wider(values_from = value,
              names_from = power)%>%
  mutate(`3`=`3`/`2`^2,
         `4`=`4`/`2`^4)%>% # Kurtosis and Skewness
  pivot_longer(-c("age", "name"), names_to = "power")%>%
  data.frame()

pred$power <- factor(c("Expected value", "Standard deviation",
                       "Skewness","Kurtosis")[as.numeric(pred$power)],
                     levels=c("Expected value", "Standard deviation",
                              "Skewness", "Kurtosis"))
```

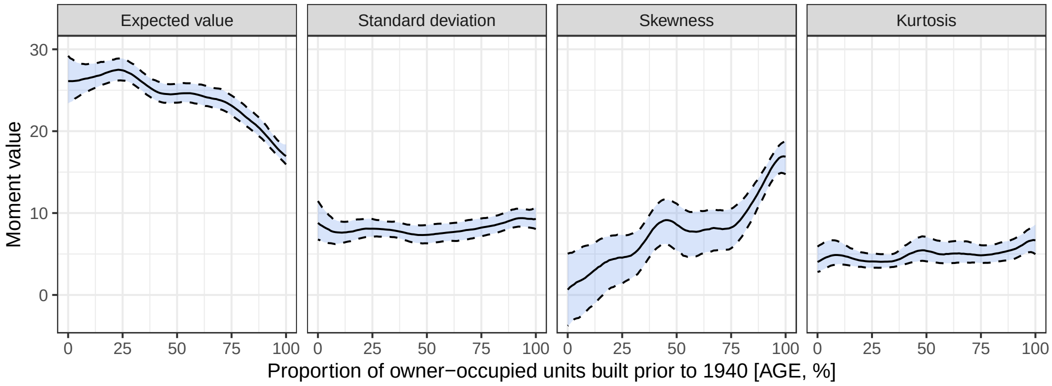

Figure 15: Posterior moments (mean, standard deviation, skewness, and kurtosis) of the conditional density field estimated with a MCMC scheme.

Depending on the application, users may opt for a faster point estimate or a more comprehensive probabilistic predictor. In all cases, the SLGP model has demonstrated its ability to capture complex variations in shape and to provide uncertainty estimates where relevant. Furthermore, estimating the full conditional distribution yields a variety of useful functionals-such as quantiles, moments, and probabilities-without relying on strong parametric assumptions, restrictive data structures, or arbitrary binning and pooling of the data.

We now demonstrate another application of our model to the Boston Housing dataset.

4.2 Discrete probability estimation with a SLGP model

A particularly useful feature of the SLGP framework is its ability to handle discrete response variables, thanks to the quadrature-based integration method introduced in Equation 26. While SLGPs naturally model continuous densities, this approach extends their applicability to discrete outcomes in a principled way.

We demonstrate this flexibility using the Boston Housing dataset by modeling the distribution of the index of accessibility to radial highways (denoted `rad`), conditioned on `age` . The variable `rad` takes discrete integer values from 1 to 24, where 24 indicates highest accessibility and 1 the lowest.

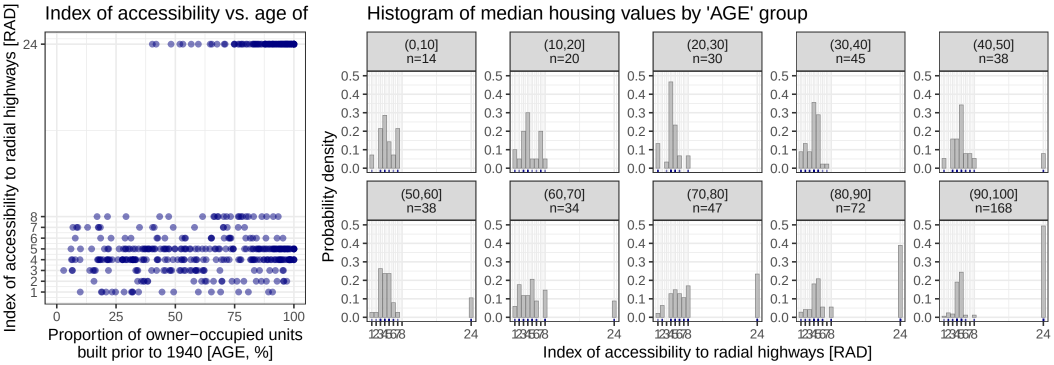

Figure 16: A visual representation of the dependency of the index of accessibility to radial highways on proportion of owner-occupied units constructed before 1940 in the Boston Housing dataset.

We fit an SLGP using MAP estimation and Random Fourier Features (RFF). To account for the discrete nature of the response, we set the quadrature grid to exactly match the support of `rad` . The code below illustrates this setup:

```
modelMAPdisc <- slgp(rad~age, # Use a formula with two indexing variables
                     data=df,
                     method="MAP", #Maximum a posteriori estimation scheme
```

```
                              basisFunctionsUsed = "RFF",
                              interpolateBasisFun="WNN",
                              hyperparams = list(lengthscale=c(0.15, 0.15), sigma2=1),
                              nIntegral =  24, # Important, to adjust the quadrature
                              sigmaEstimationMethod = "heuristic",
                              predictorsLower= c(range_x[1]),
                              predictorsUpper= c(range_x[2]),
                              responseRange=c(1, 24), # Important, to adjust the quadrature
                              opts_BasisFun = list(nFreq=200, MatParam=5/2),
                              seed=1)
```

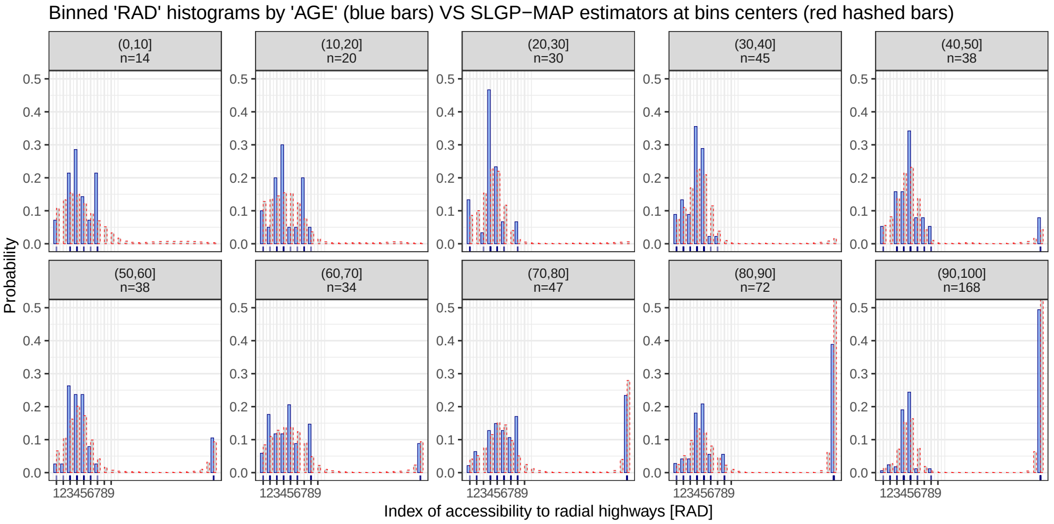

Figure 17: SLGP MAP estimation with heuristic length-scale selection versus histograms of values at bins centered at several `age` values.

The MAP estimation captures the sharp mode around `rad` $= 24$ very well, especially for higher values of `age` . However, in age groups under 20, where the empirical distribution can appear bimodal or more dispersed, the SLGP struggles to fully represent the structure. This limitation may stem from two main causes: data sparsity in the low-`age` region, or insufficient model flexibility in the `rad` direction due to a fixed length-scale. While the first issue cannot be addressed without additional data, the second can be tackled.

One approach is to increase model flexibility by reducing the length-scale along the response dimension (`rad`). However, this can lead to overfitting or numerical instability if not carefully controlled. An alternative strategy is to transform the response domain to reduce its range and smooth out irregularities in density.

As an illustration, we apply a simple transformation by remapping the value `rad` $= 24$ to `rad` $= 9$, thereby shortening the effective domain. This adjustment improves the alignment between the response distribution and the GP basis functions, making the model more expressive and better adapted to the structure of the data. The resulting data spread can be visualised in Figure 18.

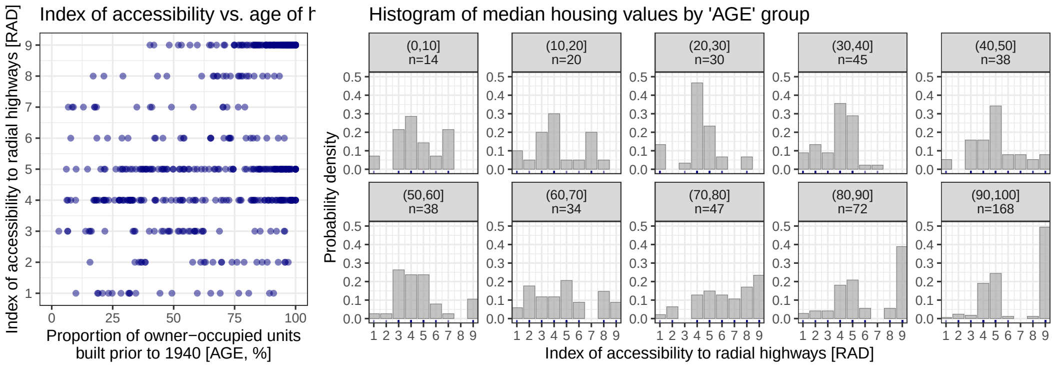

Figure 18: A visual representation of the dependency of the index of accessibility to radial highways on proportion of owner-occupied units constructed before 1940 in the Boston Housing dataset, where RAD = 24 is mapped to RAD = 9.

```
df$rad2 <- ifelse(df$rad==24, 9, df$rad)

modelMAPdisc2<- slgp(rad2~age, # Use a formula with two indexing variables
                     data=df,
                     method="MAP", #Maximum a posteriori estimation scheme
                     basisFunctionsUsed = "RFF",
                     interpolateBasisFun="WNN",
                     hyperparams = list(lengthscale=c(0.15, 0.15), sigma2=1),
                     nIntegral = 9, # Important, to adjust the quadrature
                     sigmaEstimationMethod = "heuristic",
                     predictorsLower= c(range_x[1]),
                     predictorsUpper= c(range_x[2]),
                     responseRange=c(1, 9), # Important, to adjust the quadrature
                     opts_BasisFun = list(nFreq=200, MatParam=5/2),
                     seed=1)
```

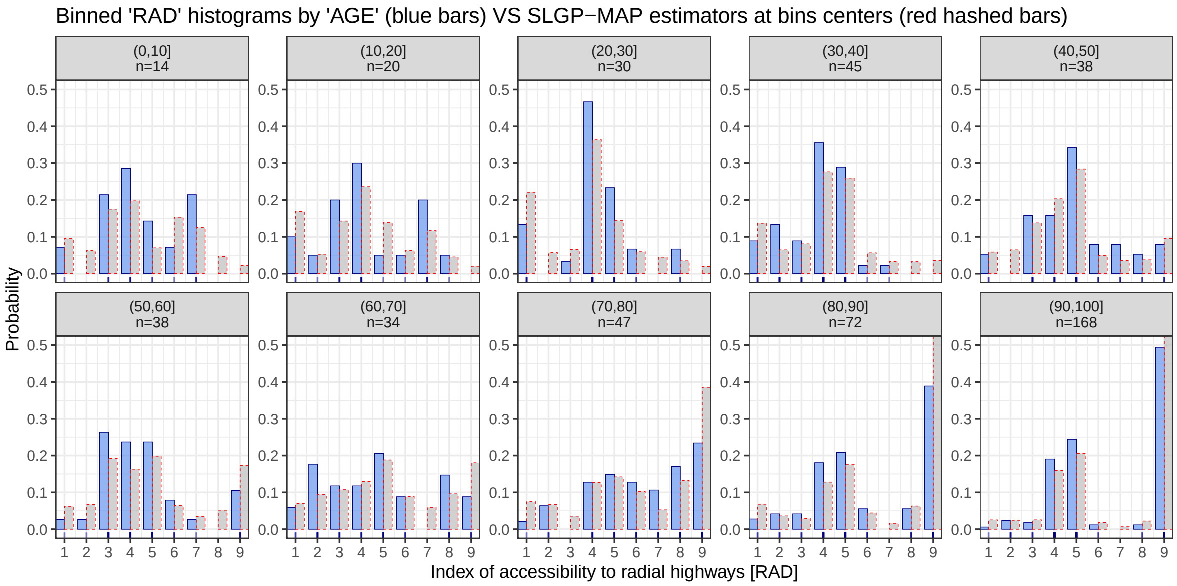

Figure 19: SLGP MAP estimation with heuristic length-scale selection versus histograms of values at bins centered at several `age` values.

As shown in Figure 19, this simple transformation leads to a marked improvement in model fit. The SLGP is now better able to capture both sharp peaks and variations in modality in the conditional distribution. While some discrepancies persist (unsurprisingly given the data sparsity and coarse histogram bins) the results remain encouraging.

This example highlights the practical utility of SLGP models in discrete settings, where traditional density estimation methods may struggle due to the lack of continuous structure or the presence of irregular support.

We now showcase the SLGP's asymptotic behaviour under various model specification and sampling strategies.

4.3 Posterior consistency of SLGP modeling

To assess the quality of SLGP-based density field estimation, we first define a reference field using the MAP estimate, which serves as the ground truth. We then generate synthetic data from this reference field and compare different estimation setups to evaluate their impact on accuracy. We explore three main experimental factors, and for each setting we replicate the experiment 25 times with several seeds to account for variability:

- Number of Basis Functions : This experiment evaluates the impact of the number of Random Fourier Features (RFFs). A larger number of basis functions is expected to improve accuracy but increase computational cost. This setup allows us to study the approximation error induced by using a finite number of basis functions and assess the trade-off between speed and expressiveness.
- Lengthscale Misspecification: This experiment, we test the sensitivity of SLGP estimation to various specification of the lengthscale hyperparameter. We train models using different lengthscales. We consider lengthscale that are too small: 0.01, 0.05; well-specified: 0.15; or too large: 0.25, 0.5. This analysis highlights the consequences of hyper-parameter selection.
- Sampling Schemes (Experiment 3) This experiment investigates how the spatial structure of the data affects the quality of estimation. We vary the sampling scheme across different types: Uniform random sampling for the $\mathbf{x}$; Points clustered around a few modes (with 5, 11, or 21 modes), Uniform random sampling with two censored regions (we remove data-points with medv in $[20, 30]$ and $[70, 90]$). These variations reveal the method's ability to generalize in poorly observed regions.

This analysis provides practical insights into how SLGP estimation behaves under different conditions, guiding model selection for real applications.

For each tested configuration, we compute the estimation error, comparing the inferred density field to the reference field. This allows us to measure the effect of different modeling choices.To assess the similarity between estimated and true distributions, we compute several integrated distances based on both the probability density functions (PDFs) and the cumulative distribution functions (CDFs).

The considered distances are defined for a distribution P_1 that admits a pdf f_1 and cdf F_1 and a distribution P_2 that admits a pdf f_2 and cdf F_2 as:

- Hellinger Distance $d_H(f_1, f_2) = \sqrt{\frac{1}{2} \int \left(\sqrt{f_1(t)} - \sqrt{f_2(t)}\right)^2 dt}$
- Total Variation Distance $TV(f_1, f_2) = \frac{1}{2} \int |f_1(t) - f_2(t)|\, dt$
- Kullback–Leibler Divergence $KL(f_1, f_2) = \int f_1(t) \log\left(\frac{f_1(t)}{f_2(t)}\right) dt$
- Cramér–von Mises Distance: $CvM(F_1, F_2) = \int (F_1(t) - F_2(t))^2 f_1(t)\, dt$

These metrics are evaluated across a grid of values and provide different perspectives on the quality of the density estimation.

4.4 Impact of the number of basis functions

We first evaluate the effect of the number of RFF on SLGP estimation, keeping the length-scale fixed at the well-specified value and the design strategy as uniform. For each configuration, we generate synthetic datasets from the reference field and fit SLGP models using MAP estimation. The resulting estimated densities are then compared to the reference field by integrating the distances defined previously and the results are displayed in Figure 20.

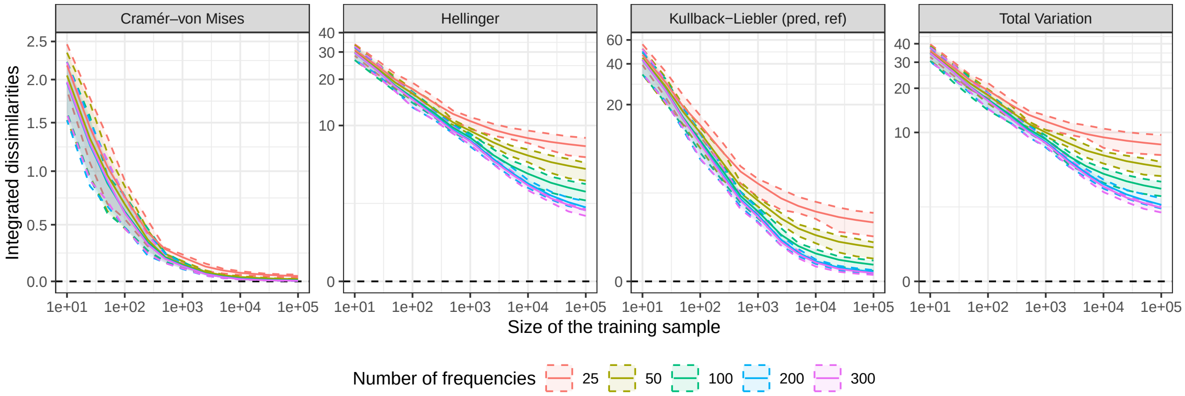

Figure 20: Effect of the number of basis functions on SLGP density estimation. Integrated distances between the MAP estimates and the reference field are shown across varying training sample sizes: mean (solid lines) and 25th–75th percentile range (shaded ribbon).

As expected, increasing the number of basis functions improves estimation accuracy. Beyond a threshold ($n_{\text{Freq}} = 200$), additional features provide negligible gains. Also, the Kullback–Leibler divergence and Cramér–von Mises distance converge rapidly toward zero, while the Hellinger distance and Total Variation distance improve more gradually. This is due to sensitivities of these metrics to local versus global discrepancies between densities.

4.5 Impact of the length-scale selection

Using the well-specified number of basis functions (200) and a uniform sampling of `age` , we next investigate how the choice of length-scale affects MAP-based SLGP estimation, with the results being displayed in Figure 21.

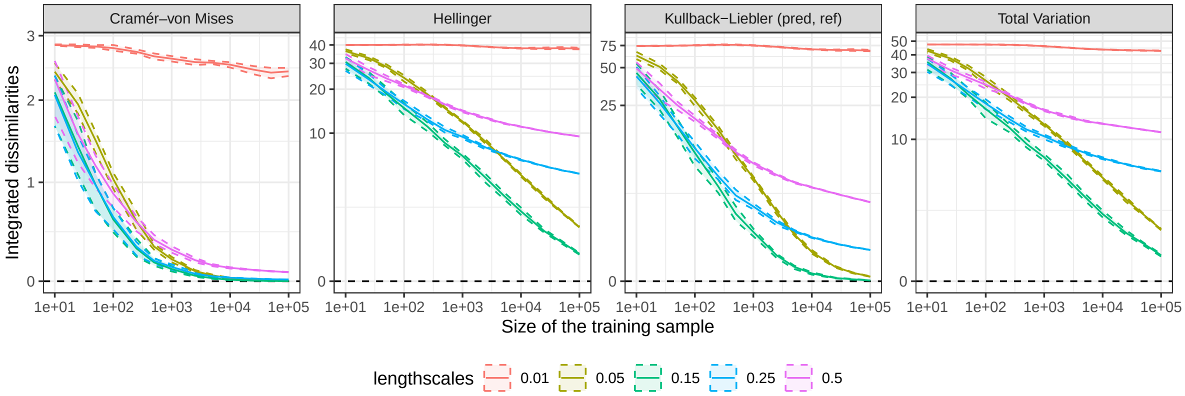

Figure 21: Effect of the lengthscale on SLGP density estimation using MAP. Integrated distances between the MAP estimates and the reference field are shown across varying training sample sizes: mean (solid lines) and 25th–75th percentile range (shaded ribbon).

The analysis indicates that the best performance across all metrics is obtained using the true or near-true lengthscale. Short lengthscales lead to overly localized predictions, failing to capture broader trends, whereas long lengthscales oversmooth the distributions, missing fine-scale features such as multi-modality. The well-specified lengthscale provides a balanced compromise, reproducing both global and local structure in the reference field.

4.6 Impact of the Sampling Scheme

Finally, we assess the influence of the sampling design on SLGP performance. We fix the number of basis functions and the length-scale to well-specified values and compare MAP estimates to the reference field across different `age` selection schemes:

- uniform on the regular grid with 5 points: $\{0, 25, 50, 75, 100\}$,
- uniform on the regular grid with 11 points: $\{0, 10, 20, \dots, 100\}$,
- uniform on the regular grid with 21 points: $\{0, 5, 10, \dots, 100\}$,
- uniform on the full segment $[0, 100]$,
- and uniform with holes where we consider a distribution uniform on $[0, 10] \cup [20, 60] \cup [80, 100]$.

We visualise the integrated dissimilarities with varying sample sizes in Figure 22

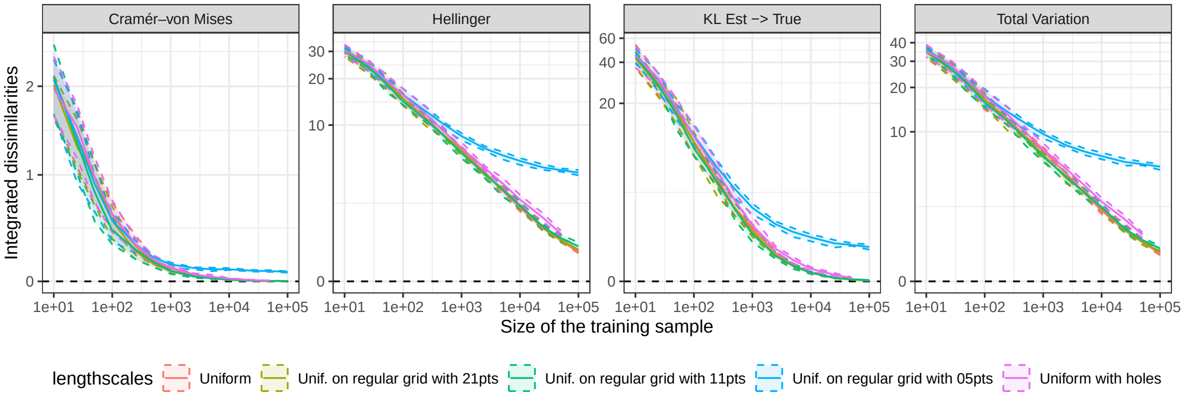

Figure 22: Impact of selection scheme for `age` in SLGP density estimation using MAP. Integrated distances between the MAP estimates and the reference field are shown across varying training sample sizes: mean (solid lines) and 25th–75th percentile range (shaded ribbon).

Estimation performance depends on the sampling scheme used for `age` . The "Uniform on a regular grid with 5 point" design consistently leads to the weakest performance, characterized by a systematic estimation bias. In comparison, the remaining designs achieve similar levels of accuracy, with the "Uniform with hole" scheme showing only a modest loss in efficiency.

To better understand these differences, we look at localised errors in addition to the integrated ones. We examine how the estimation error varies as a function of `age` . We display in Figure 23 the spatial variations of the Kullback-Liebler divergence between the estimated probability density function at a given `age` and the corresponding reference density. The same is done for Cramér–von Mises criterion, Hellinger distance, and Total variation distance in Figures A-6, A-7 and A-8 of Appendix C.

These results illustrate the impact of localisation: regions with missing or sparse data exhibit higher errors, and dense coverage in some areas does not always compensate for missing information elsewhere.

This analysis suggests a practical guideline: if the goal is to learn the full field rather than targeting specific locations, it is preferable to distribute sampling points evenly across the space rather than accumulating them at selected covariates.

However if the goal is to refine knowledge at prescribed location, intensifying sampling in the area leads to better local predictions.

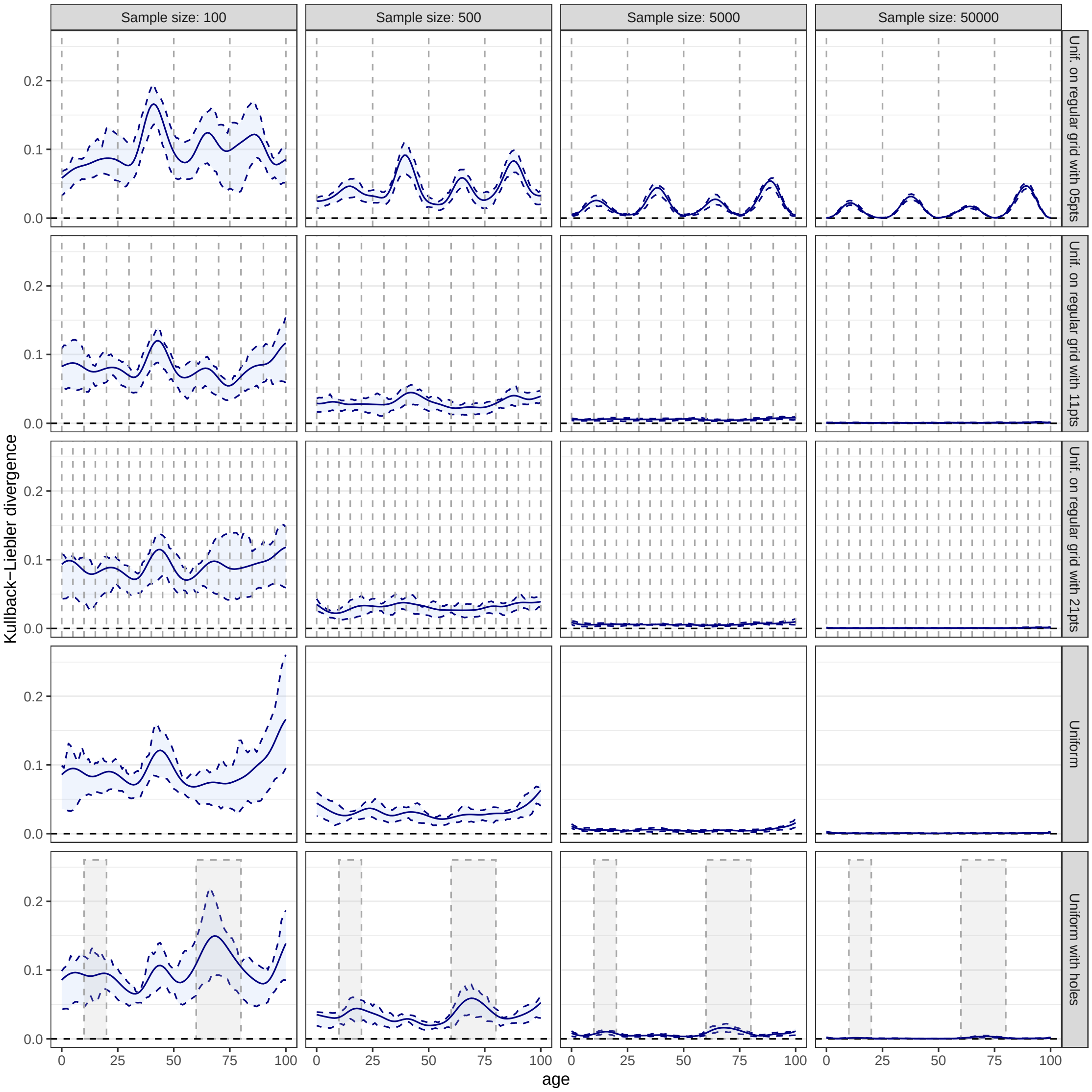

Figure 23: Impact of the `age` selection scheme on SLGP density estimation using MAP. Local Kullback–Leibler divergences between the estimated densities and the reference field are shown across varying training sample sizes for different schemes: mean (solid lines) and 25th–75th percentile range (shaded ribbon).

Finally, these observations raise the question of the interplay between sampling density and the length-scales of the underlying process. While this relationship is central to optimal design and model calibration, a systematic investigation of this trade-off lies beyond the scope of the present work and is deferred to future research.

5 Conclusion and perspective

In this work, we provided a practical introduction to Spatial Logistic Gaussian Process (SLGP) modeling, including a tutorial for implementation in `R`. We showed that SLGPs offer a flexible and expressive framework for modeling spatially dependent probability distributions, capable of capturing heterogeneity and multimodality while supporting both point estimation, via maximum a posteriori inference, and fully probabilistic inference, for instance through MCMC or Laplace approximations. Beyond accurate density estimation, SLGPs naturally yield a range of informative distributional summaries, including quantiles, moments, and other functionals of interest. Moreover, the framework extends seamlessly to discrete and ordinal outcomes, thereby broadening its applicability well beyond the continuous setting. All accompanying code is publicly available, ensuring full reproducibility and facilitating adoption in a wide range of applied contexts (**?**).

Through systematic experiments, we illustrated the practical aspects of SLGP modeling, emphasizing the influence of hyperparameter choices (e.g., length-scale, number of basis functions) and sampling design on estimation quality. These insights provide concrete guidance for configuring SLGP models and for planning data collection strategies, particularly in contexts where observations are costly or unevenly distributed.

Looking forward, SLGPs offer substantial promise for advanced statistical inference tasks, including uncertainty quantification, goal-oriented modeling, and adaptive experimental design. In particular, extensions to stochastic inversion using SLGP-based Approximate Bayesian Computation (ABC) (**?**) provide a flexible approach to tackle high-dimensional, stochastic inverse problems. This framework opens avenues for principled, data-driven learning for complex systems.

Several directions for future research naturally follow from this work. A first important avenue concerns the interplay between sampling density and the effective length-scales of the SLGP. While our numerical experiments highlight the influence of design choices on estimation quality, a systematic theoretical and empirical investigation of this trade-off—particularly in the context of adaptive experimental design—remains to be conducted. Understanding how sampling density should scale with the smoothness of the underlying distributional field would provide principled guidance for data acquisition and design strategies.

Moreover, the SLGP framework is well suited to adaptive experimental design, as it yields full probabilistic descriptions of conditional distributions. Future work will focus on developing acquisition criteria tailored to distribution-valued outputs.

Finally, another extension of the present framework is the modeling of multivariate responses, with a possible mix between discrete and continous output. Incorporating dependence structures and exploring multiple-output SLGP would substantially broaden its applicability.

Acknowledgements

This work was largely supported by the Swiss National Science Foundation (SNSF) under Grant No. *178858*. The author would like to thank David Ginsbourger for his proofreading of earlier versions of the manuscript and for numerous insightful suggestions regarding SLGP's study. The author is also particularly grateful to Yves Deville for his contributions during the early stages of the software development, as well as for methodological suggestions that helped shape the implementation choices presented in this work.

A Appendix: On the SLGP's denominator's integral

A.1 Definition of the approximation schemes for the normalising integral of the SLGP

We recall that our goal is to provide a reasonably fast and numerically stable approximation of the negative log likelihood in a SLGP model.

$$\ell(\boldsymbol{\epsilon}|D) = -\boldsymbol{\epsilon}^\top \sum_{i=1}^{m} F(x_i, t_i) + \sum_{i=1}^{m} \left(\log \int_0^1 \exp\{\boldsymbol{\epsilon}^\top F(x_i, u)\}\, du \right) \tag{30}$$

First, we note that when noting $M_i := \max_u \boldsymbol{\epsilon}^\top F(x_i, u)$, we have the equality:

$$\ell(\boldsymbol{\epsilon}|D) = -\boldsymbol{\epsilon}^\top \sum_{i=1}^{m} F(x_i, t_i) + \sum_{i=1}^{m} M_i + \sum_{i=1}^{m} \left(\log \int_0^1 \exp\{\boldsymbol{\epsilon}^\top F(x_i, u) - M_i\}\, du \right) \tag{31}$$

This simple numerical manipulations allows for greater numerical stability, ensuring that the exponentials will not go out of range, and that the Riemann's sum approximation of the integrals can be performed without issues.

$$\ell(\boldsymbol{\epsilon}|D) \approx -\boldsymbol{\epsilon}^\top \sum_{i=1}^{m} F(x_i, t_i) + \sum_{i=1}^{m} M_i + \sum_{i=1}^{m} \log \left(\frac{1}{m} \sum_{j=1}^{m} \exp\{\boldsymbol{\epsilon}^\top F(x_i, u_j) - M_i\} \right) \tag{32}$$

While numerically stable, this term remains fairly expensive to evaluate, as it involves approximating one integral (and the maximum values M_i) per distinct value of the predictor xX. However, when the family of basis functions consist of smooth elements, for any value of $\boldsymbol{\epsilon}$, the term $\int_{\mathcal{T}} e^{\boldsymbol{\epsilon}^T F(\mathbf{x},u)}\, du$ will evolve smoothly with $\mathbf{x}$, and as such, one can enjoy further approximations in the model to greatly alleviate numerical costs. We already mentioned in the main body of the paper that we propose three main approaches, we further detail them here, complete with equations and illustrations.

Approximation by nearest neighbour on a regular grid To avoid computing up to m integrals, we can introduce a grid of D and denote $\tilde{x}_i$ the closest neighbour of x_i on this grid and $\tilde{u}_i$ that of t_i.

$$\ell(\boldsymbol{\epsilon}|D) \approx \ell_{NN}(\boldsymbol{\epsilon}|D) \approx \boldsymbol{\epsilon}^\top \sum_{i=1}^{m} F(\tilde{x}_i, \tilde{u}_i) - \sum_{i=1}^{m} \log \left(\frac{1}{m} \sum_{j=1}^{m} \exp\{\boldsymbol{\epsilon}^\top F(\tilde{x}_i, u_j)\} \right) \tag{33}$$

This enables one to bound the maximum number of integrals to compute. However, it yields a piecewise constant approximation of the negative log-likelihood landscape, which can too rough of an approximation.

Weighted nearest neighbours Yet another suitable approximation is to use a weighted combination of several neighbours.

For a point $[x_i, t_i]$, we denote $\left[\check{x}_{ij}, \check{t}_{ij}\right]_{j=1}^{2d+1}$ the vertices of the smallest enclosing hypercube within a regular grid. Considering weight factors w_{ij} (with $\sum w_{ij} = 1$), we get:

$$\ell(\boldsymbol{\epsilon}|D) \approx \ell_{WNN}(\boldsymbol{\epsilon}|D) \approx \sum_{i=1}^{m} \log \left(\sum_{j=1}^{2^{d+1}} w_{ij} \frac{\exp\{\boldsymbol{\epsilon}^\top F(\check{x}_{ij}, \check{u}_{ij})\}}{\frac{1}{m}\sum_{l=1}^{m} \exp\{\boldsymbol{\epsilon}^\top F(\check{x}_{ij}, u_\ell)\}} \right) \tag{34}$$

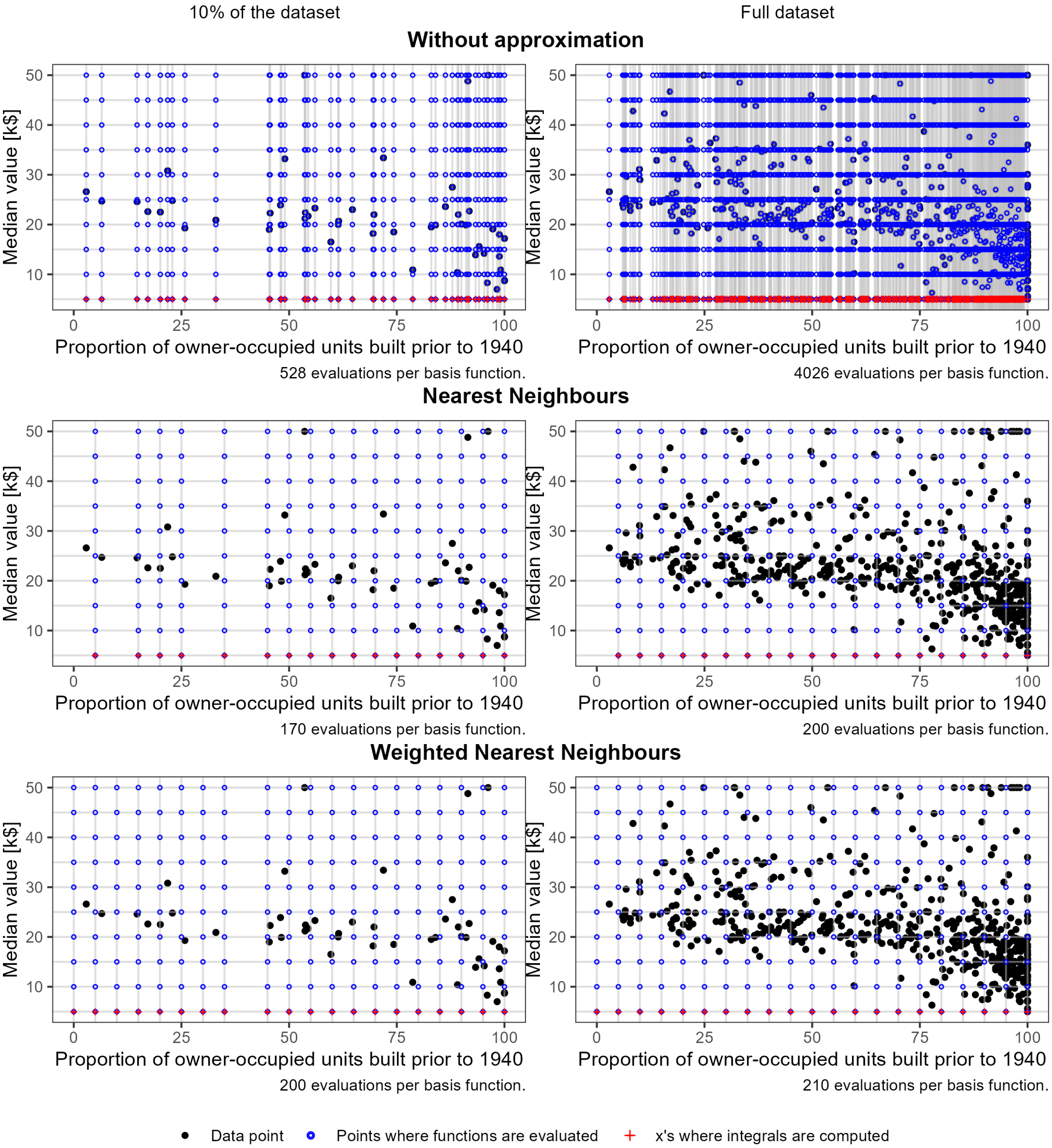

Figure A-1: Visualisation of three strategies for integral approximation applied to the Boston Housing dataset, distinguished by sample sizes. The top panels approximates separately the integral for each predictor variable **x**. In the middle panels, the 'Nearest Neighbour' approach assigns the value of the closest grid node to each data point. The bottom panels demonstrate the 'Weighted Nearest Neighbour' method, calculating the integral value at a data point as a weighted sum of grid node values within the smallest enclosing hypercube

B Appendix: optimising the length-scale

To assess the impact of length-scale selection on SLGP estimation, we follow the MAP-based strategy described in Section 3.4. We use an Inverse-Gamma prior on each length-scale parameter, perform posterior evaluations and compare optimization results under two different data usage regimes:

- Full dataset: model is trained and evaluated on the full data.
- Partial training: model is trained on 25% of the data and evaluated on the remaining 75%.

Figure A-2 displays the resulting negative log-posterior landscapes.

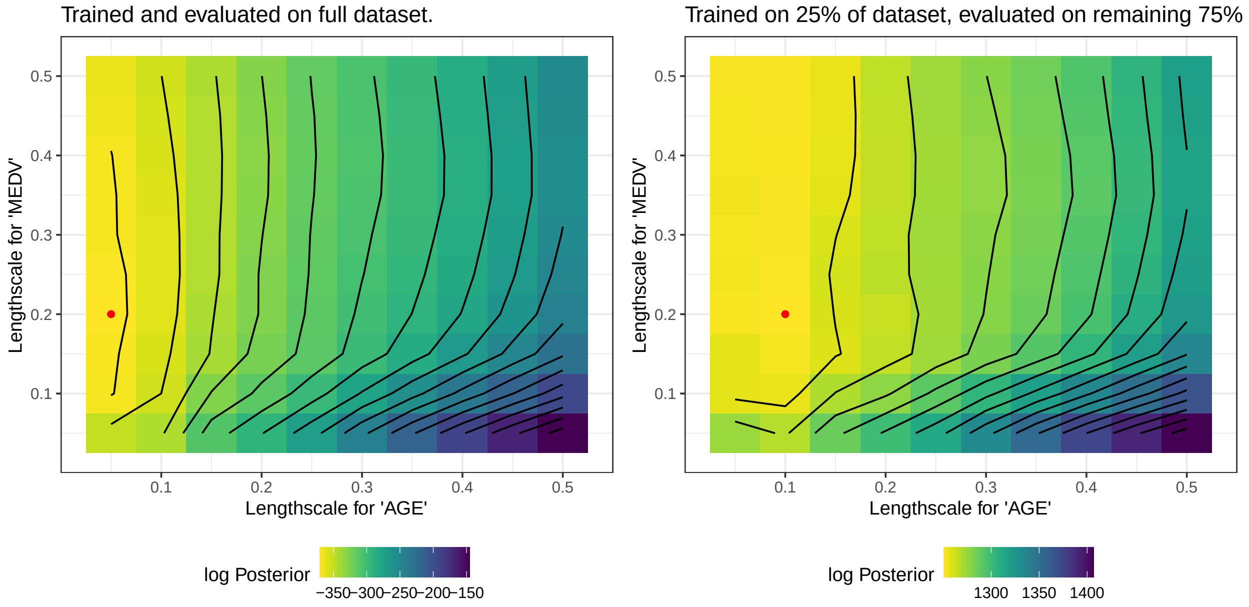

Figure A-2: Posterior log-density (up to a constant) over the length-scale grid (`age` , `medv`), under two training strategies. Left: trained and evaluated on the full dataset. Right trained on 25% of the data, evaluated on the remaining 75%. The red dot indicates the optimal length-scales in each setting.

We observe that both approaches yield similar optimal length-scales (5% and 20% of the range for the full dataset training VS 10% and 20% of the range for the partial dataset training), showing the stability of the selection. Moreover, these landscapes are fairly consistent with our proposed heuristic of setting all length-scales to 15% of the ranges.

Now, for a qualitative interpretation, we fit and compare SLGP models with four representative choices of length-scales:

- The heuristic length-scale: 15% of the `age` range and 15% of the `medv` range.
- The long length-scale: 50% of the `age` range and 50% of the `medv` range.
- The optimised length-scale: 10% of the `age` range and 20% of the `medv` range.
- The short length-scale: 1% of the `age` range and 1% of the `medv` range.

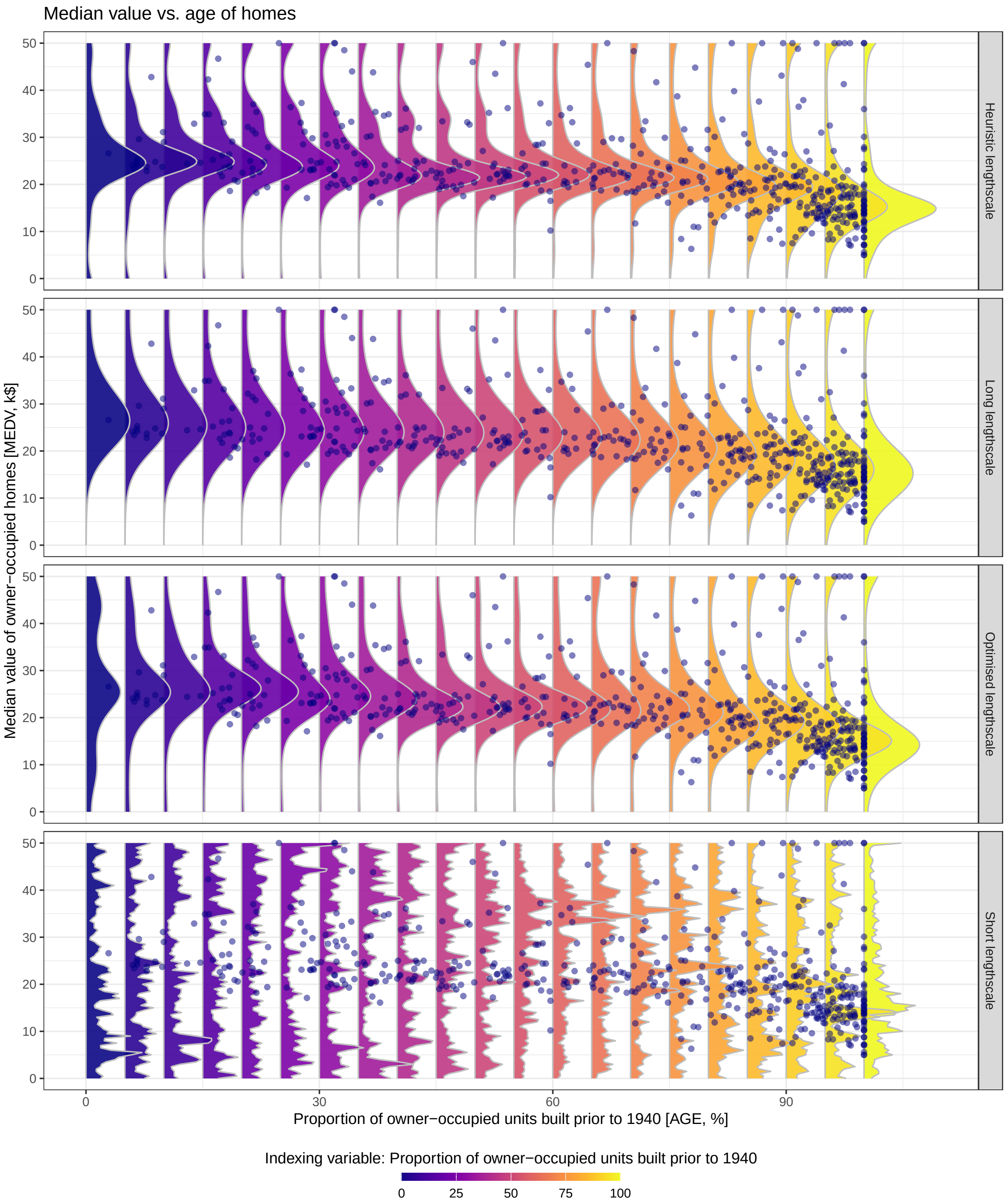

Figure A-3: Comparison of fitted densities under different length-scale settings

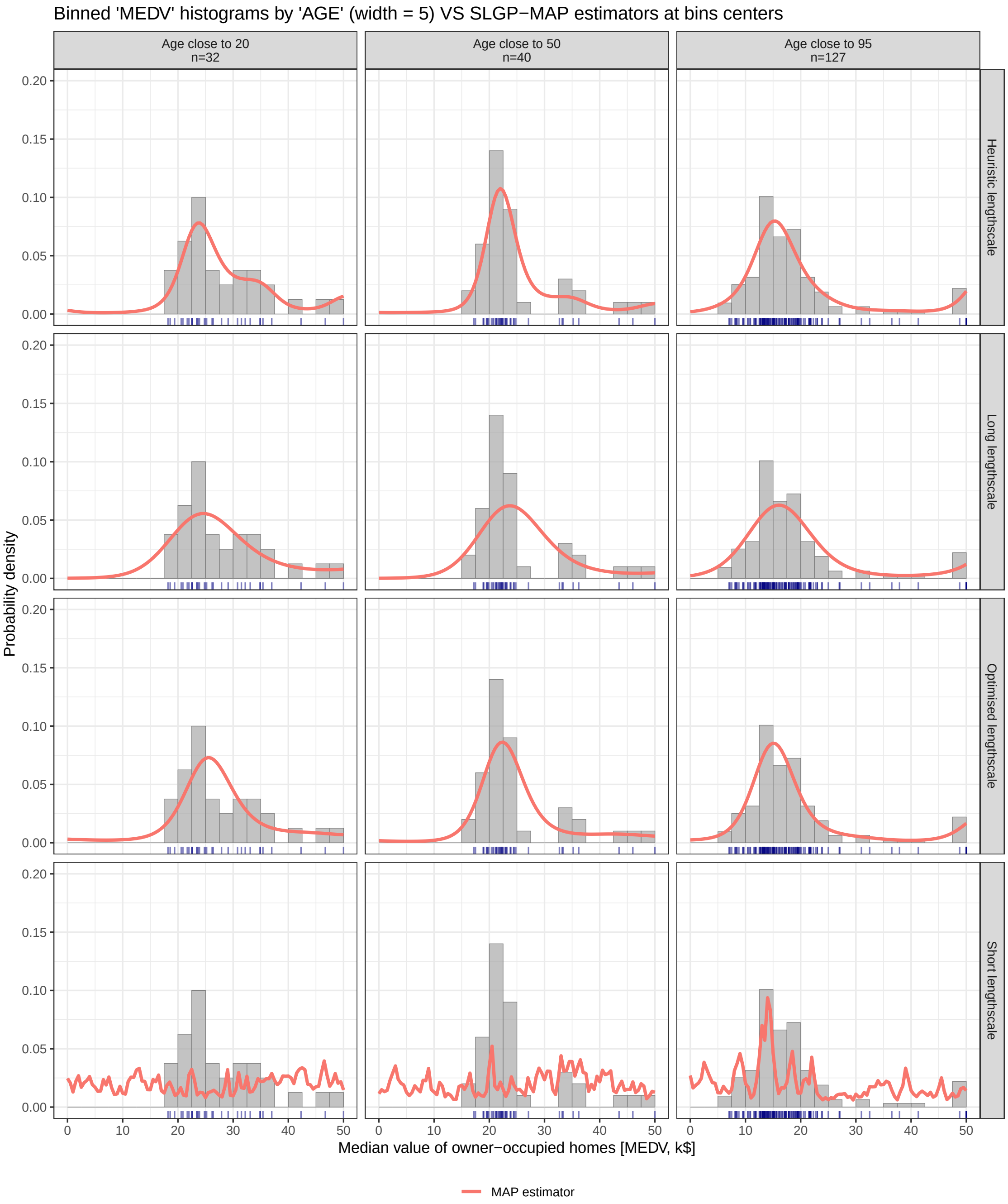

Figure A-4: Comparison of fitted densities under different length-scale settings

We observe that when the length-scale is too big, while the model captures the general support of the distribution, it fails to capture finer details, such as multi-modalities in certain distributions. This limitation likely arises from using a large length-scale, which oversmooths small-scale variations. On the opposite using too small a length scale restricts the model's spatial aggregation, limiting the reach of spatial information to a very local scope.

Using a heuristic as before, or the optimised version yield a compromise between the previously observed limitations. For simplicity, we will continue with the length-scales yielded by the heuristic, set to 15% of the ranges.

C Appendix: Additional figures

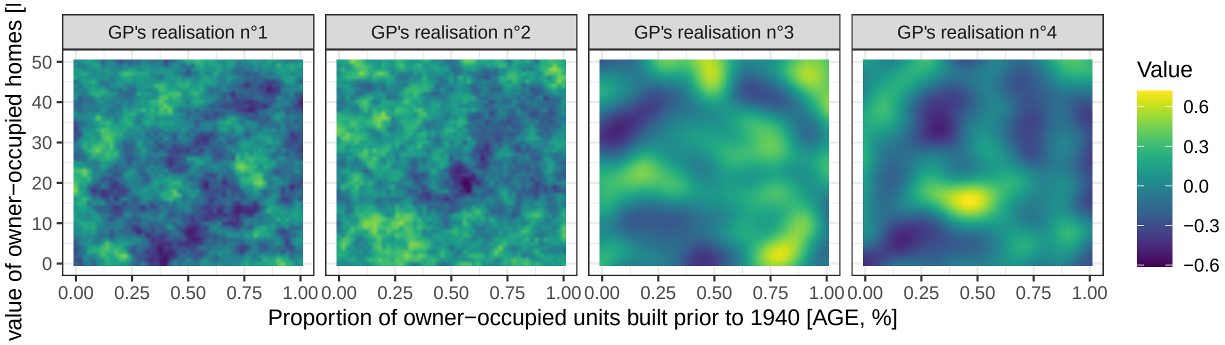

Figure A-5: The four realisations of GPs underlying the SLGPs in Figure 6

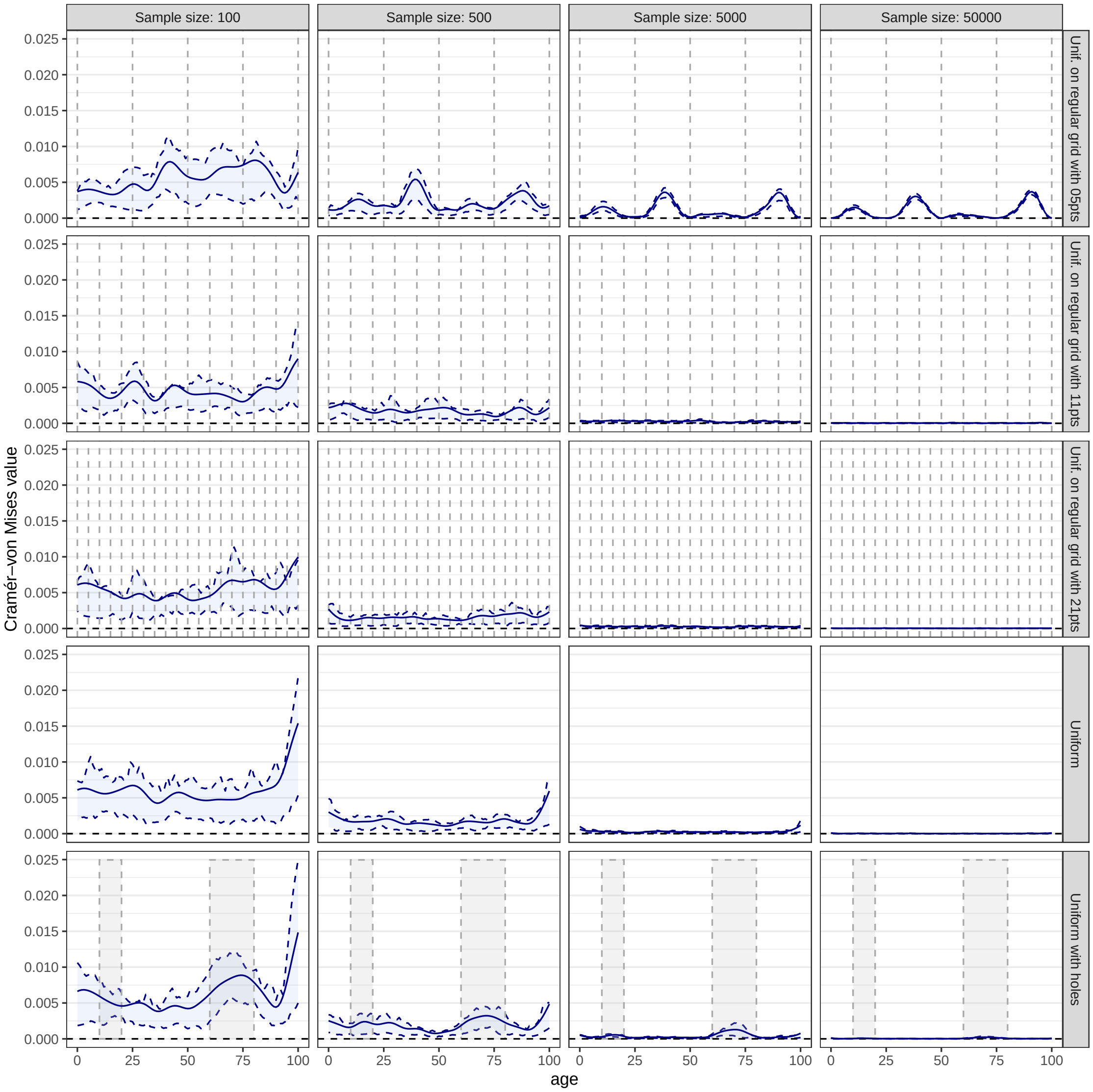

Figure A-6: Impact of the `age` selection scheme on SLGP density estimation using MAP. Local Cramér–von Mises criterion between the estimated densities and the reference field are shown across varying training sample sizes for different schemes: mean (solid lines) and 25th–75th percentile range (shaded ribbon).

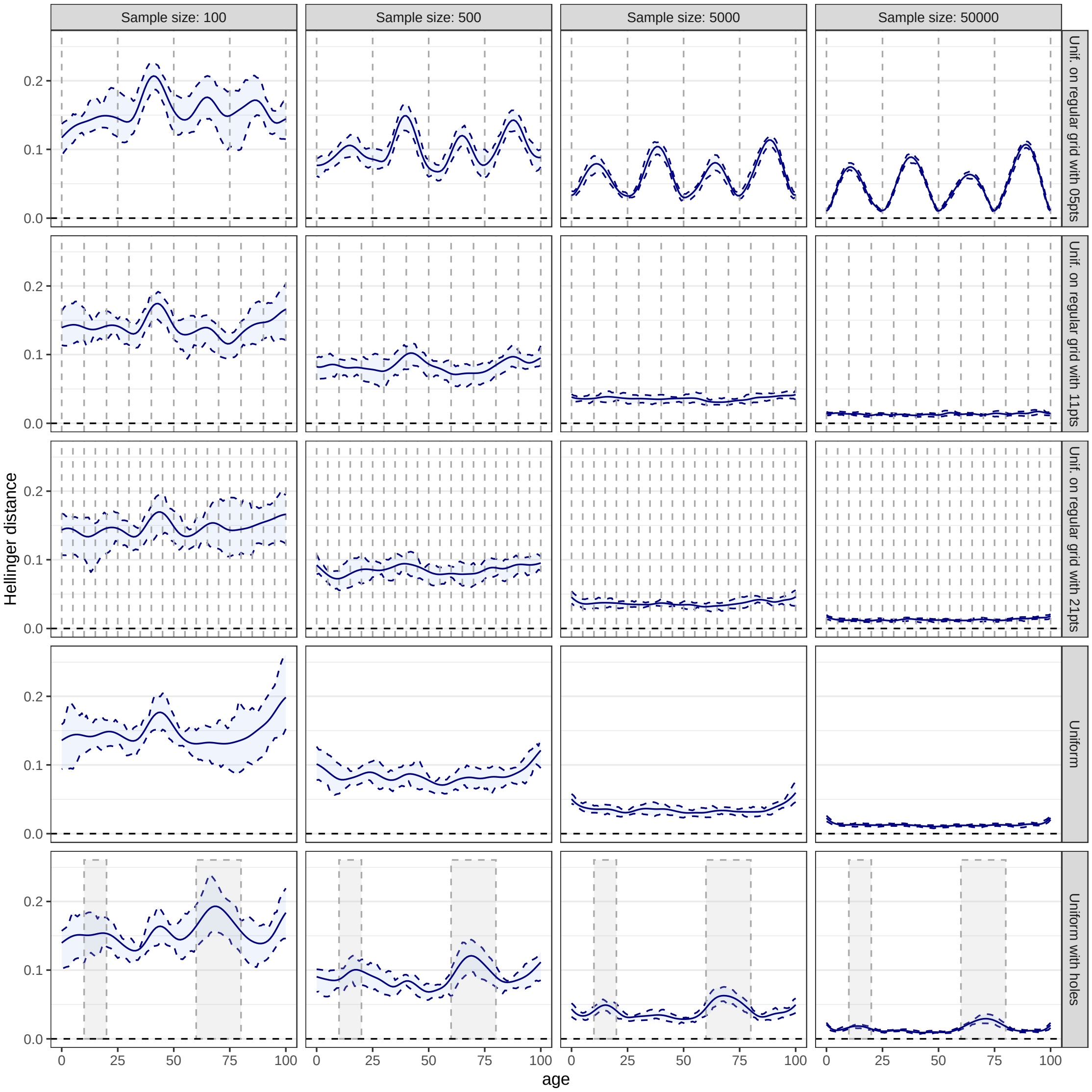

Figure A-7: Impact of the age selection scheme on SLGP density estimation using MAP. Local Hellinger distances between the estimated densities and the reference field are shown across varying training sample sizes for different schemes: mean (solid lines) and 25th–75th percentile range (shaded ribbon).

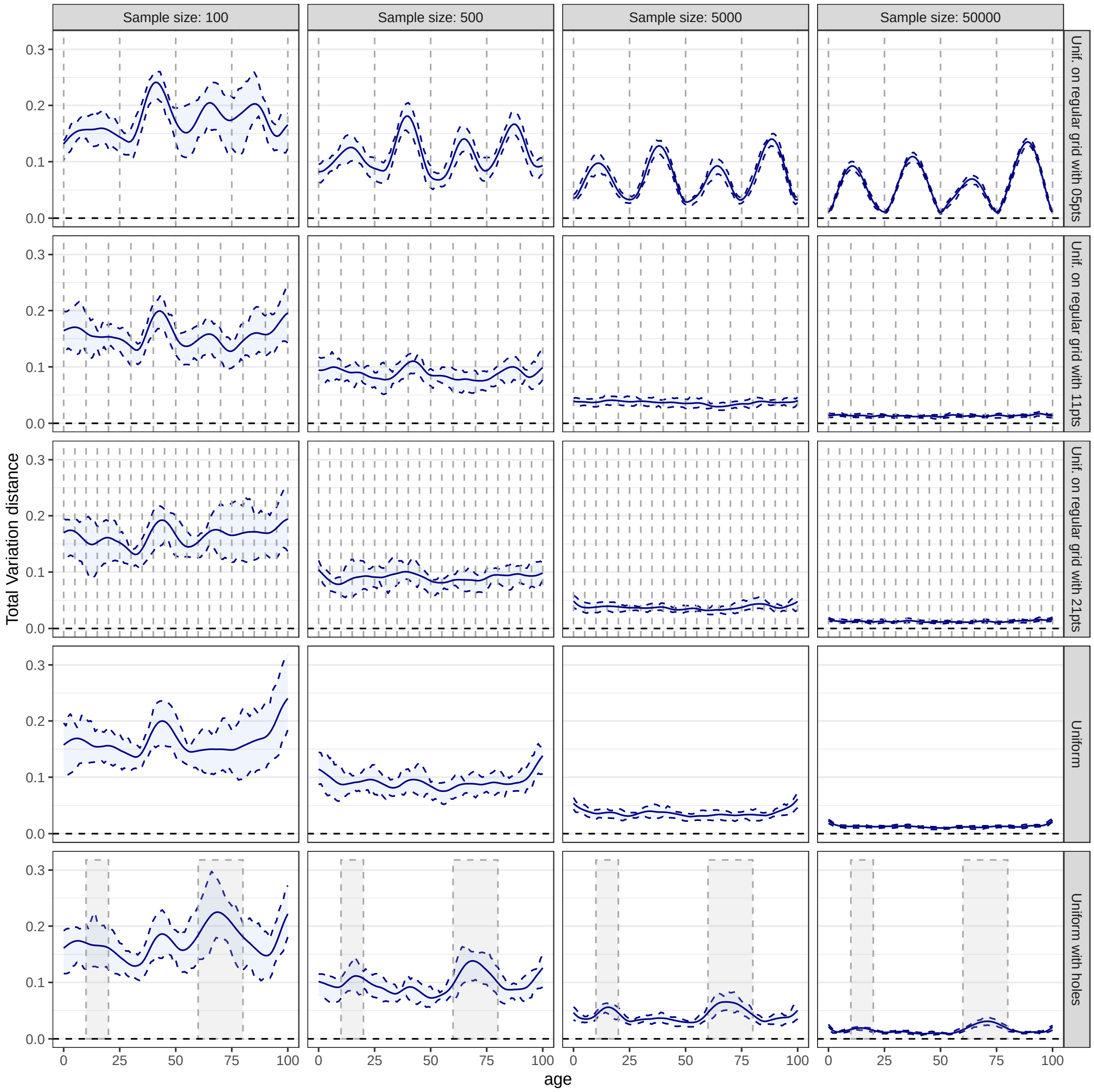

Figure A-8: Impact of the `age` selection scheme on SLGP density estimation using MAP. Local total variation distances between the estimated densities and the reference field are shown across varying training sample sizes for different schemes: mean (solid lines) and 25th–75th percentile range (shaded ribbon).