Machine Learning (2023) 112:1411–1432
https://doi.org/10.1007/s10994-022-06297-3

Correlated product of experts for sparse Gaussian process regression

Manuel Schürch[1,2] · **Dario Azzimonti**[1] · **Alessio Benavoli**[1,3] · **Marco Zaffalon**[1]

Received: 26 December 2021 / Revised: 25 November 2022 / Accepted: 18 December 2022 /
Published online: 25 January 2023

Abstract
Gaussian processes (GPs) are an important tool in machine learning and statistics. However, off-the-shelf GP inference procedures are limited to datasets with several thousand data points because of their cubic computational complexity. For this reason, many sparse GPs techniques have been developed over the past years. In this paper, we focus on GP regression tasks and propose a new approach based on aggregating predictions from several local and correlated experts. Thereby, the degree of correlation between the experts can vary between independent up to fully correlated experts. The individual predictions of the experts are aggregated taking into account their correlation resulting in consistent uncertainty estimates. Our method recovers independent Product of Experts, sparse GP and full GP in the limiting cases. The presented framework can deal with a general kernel function and multiple variables, and has a time and space complexity which is linear in the number of experts and data samples, which makes our approach highly scalable. We demonstrate superior performance, in a time vs. accuracy sense, of our proposed method against state-of-the-art GP approximations for synthetic as well as several real-world datasets with deterministic and stochastic optimization.

Keywords Gaussian processes · Probabilistic regression · Expert fusion

Editors: Krzysztof Dembczynski and Emilie Devijver.

✉ Manuel Schürch
manuel.schuerch@idsia.ch

✉ Dario Azzimonti
dario.azzimonti@idsia.ch

Alessio Benavoli
alessio.benavoli@tcd.ie

Marco Zaffalon
marco.zaffalon@idsia.ch

[1] Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Lugano, Switzerland

[2] Università della Svizzera italiana (USI), Lugano, Switzerland

[3] University of Limerick (UL), Limerick, Ireland

1 Introduction

Gaussian processes (GPs) are a class of powerful probabilistic method used in many statistical models due to their modelling flexibility, robustness to overfitting and availability of well-calibrated predictive uncertainty estimates with many applications in machine learning and statistics. However, off-the-shelf GP inference procedures are limited to datasets with a few thousand data points N, because of their computational complexity $\mathcal{O}(N^3)$ and memory complexity $\mathcal{O}(N^2)$ due to the inversion of a $N \times N$ kernel matrix (Rasmussen and Williams 2006). For this reason, many GP approximation techniques have been developed over the past years. There are at least two different approaches to circumvent the computational limitation of full GP. On the one hand, there are *sparse and global* methods (Csató and Opper 2002, Quiñonero-Candela and Rasmussen 2005, Rasmussen and Williams 2006, Seeger et al. 2003) based on $M_g \ll N$ so-called (global) inducing points, which cover sparsely the input space and optimally summarizing the dependencies of the training points. This results in a low-rank approximation of the kernel matrix of size $M_g \times M_g$, which is less expensive to invert. These methods consistently approximate full GP, for instance the authors in Titsias (2009) have shown that it converges to full GP as $M_g \to N$. However, all these methods are still cubic in the number of global inducing points M_g and for many applications—in particular in higher dimensions—the amount of inducing points has to be rather large to capture the pattern of the function properly. A lot of work has been done to optimize the locations of the inducing inputs e.g., Bui et al. (2017), Snelson and Ghahramani (2006), Titsias (2009), which allows to have less inducing points but more optimization parameters. This optimization procedures were further improved by stochastic optimization e.g., Bui et al. (2017), Hensman et al. (2013), Kania et al. (2021), Schürch et al. (2020), which allows to update the parameters in mini-batches and thus speed up the inference. Optimization of these (variational) parameters helps to scale GP approximations, however, the large number of optimization parameters makes these methods hard to train and they are still limited to M_g global inducing points.

On the other hand, there are *independent and local* models based on averaging predictions from J independent local experts/models resulting in a block-diagonal approximation of the kernel matrix. The final probabilistic aggregation is then based on a product of the individual predictive densities, thus they are called *Product of Experts (PoEs)*, see Fleet (2014), Deisenroth and Ng (2015), Hinton (2002), Rullière et al. (2018), Tresp (2000), Liu et al. (2018). PoE methods provide fast and rather accurate predictions, because they have fewer hyperparameters than inducing point methods and are locally exact. However, the predictive aggregation of complete independent experts leads to unreliable uncertainty estimates and less accurate predictions in regions between experts. Further, also a rigorous connection to full GP is missing. Beside the mentioned local and global methods, there are also numerical approaches, for instance by exploiting parallelism in specialized hardware (Wang et al. 2019). For a more thorough overview of GP approximations we refer to Liu et al. (2020), Rasmussen and Williams (2006).

Our approach aims to overcome these limitations by introducing a framework based on J correlated experts, so that it approximates full GP in two orthogonal directions: sparsity and locality. Thereby, our model is a generalization of the independent PoEs and sparse global GPs by introducing local correlations between experts. These experts correspond to local and sparse GP models represented by a set of *local inducing points*, which are points on the GP summarizing locally the dependencies of the training data. The degree of correlation C between the experts can vary between independent up to fully correlated

experts in a consistent way, so that our model recovers independent PoEs, sparse global GP and full GP in the limiting cases. Our method exploits the conditional independence between the experts resulting in a sparse and low-rank prior as well as posterior precision (inverse of covariance) matrix, which can be used to efficiently obtain local and correlated predictions from each expert. These correlated predictions are aggregated by the covariance intersection method (Julier and Uhlmann 1997), which is useful for combining consistently several estimates with unknown correlations. The resulting predictive distribution is a smooth weighted average of the predictive distributions of the individual experts. Our algorithm works with a general kernel function and performs well in higher dimensional input spaces. The number of hyperparameters to optimize of our method is the same as for full GP, which are just a few parameters (depending on the kernel). These parameters can be similarly estimated via the log marginal likelihood, which is analytically and efficiently computable for our model. In our inference, also log normal priors can be incorporated leading to maximum-a-posteriori estimates for the hyperparameters.

Compared to the number of *global* inducing point M_g, which is usual much smaller than the number of data points N, our approach allows a much higher of total *local* inducing points in the order of N which helps to cover the space and therefore model more complicated functions. Compared to the independent PoEs, the performance can already significantly improve by modelling just a few of the pairwise correlations between the experts. Our method shares also some similarities with other sparse precision matrix GP approximations. The works Durrande et al. (2019), Grigorievskiy et al. (2017) exploit a band precision matrix together with univariate kernels, whereas Bui and Turner (2014) propose a precision structure according to a tree. The authors Datta et al. (2016), Katzfuss and Guinness (2021) use a more general precision matrix structure, however they need to know the prediction points in advance and are only well suited for low dimensional data (i.e. 1D and 2D), which is usually not useful in the context of machine learning, where the dimension is higher and predictions are needed after training.

In Sect. 2, we briefly review full GP for regression and *sparse and global* as well as *independent and local* approaches for GP approximation. In Sect. 3, we propose our method *Correlated Product of Experts* (CPoEs), where we introduce the graphical model (Sect. 3.1) of our method and explain the local and sparse character of the prior approximation (Sect. 3.2). Further, we discuss how to make inference (Sect. 3.3) and prediction (Sect. 3.4) in our model. In Sect. 3.5, we show that the quality of our approximation consistently improves in terms of Kullback–Leibler-(KL)-divergence (B11) w.r.t. full GP for increasing degree of correlation. Further, we present deterministic and stochastic hyperparameter optimization techniques (Sect. 3.6). In Sect. 4 we compare against state-of-the-art GP approximation methods in a time versus accuracy sense, for synthetic as well as several real-world datasets. Moreover, comparison to non-GP regression methods are provided. We demonstrate superior performance of our proposed method for different (non-trivial) kernels in multiple dimensions. Section 5 concludes the work and presents future research directions.

2 GP regression

Suppose we are given a training set $\mathcal{D} = \{y_i, X_i\}_{i=1}^N$ of N pairs of inputs $X_i \in \mathbb{R}^D$ and noisy scalar outputs y_i generated by adding independent Gaussian noise to a latent function f, that is $y_i = f(X_i) + \varepsilon_i$, where $\varepsilon_i \sim \mathcal{N}(0, \sigma_n^2)$. We denote $\boldsymbol{y} = [y_1, \ldots, y_N]^T$ the

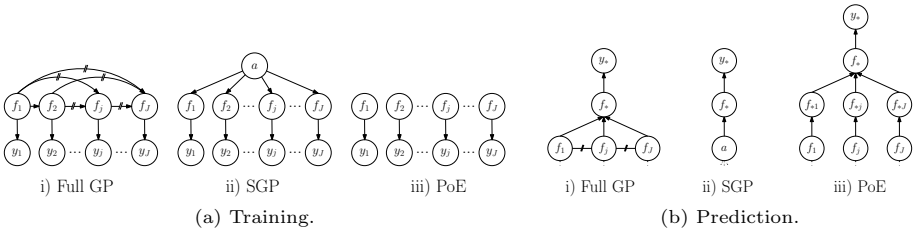

Fig. 1 Graphical models of different GP approaches

vector of observations and with $\boldsymbol{X} = [X_1^T, \dots, X_N^T]^T \in \mathbb{R}^{N \times D}$. We model f with a *Gaussian Process*, i.e. $f \sim \mathrm{GP}(m, k_{\boldsymbol{\theta}})$ with mean $m(X)$ and a covariance function (or kernel) $k_{\boldsymbol{\theta}}(X, X')$ for any $X, X' \in \mathbb{R}^D$, where $\boldsymbol{\theta}$ is a set of hyperparemeters. For the sake of simplicity, we assume $m(X) \equiv 0$ and a *squared exponential* (SE) kernel with individual lengthscales for each dimension if not otherwise stated, however, the mean function can be arbitrary and the covariance any positive definite kernel function (see, e.g., Rasmussen and Williams (2006), Chap. 4). For any input matrix $\boldsymbol{A} = [A_1; \dots; A_M] \in \mathbb{R}^{M \times D}$ consisting of rows $A_i \in \mathbb{R}^D$, we define the GP output value $\boldsymbol{a} = f(\boldsymbol{A}) = \left[f(A_1), \dots, f(A_M)\right]^T = \left[a_1, \dots, a_M\right]^T \in \mathbb{R}^M$, so that the joint distribution $p(\boldsymbol{a}) = p\left(a_1, \dots, a_M\right)$ is Gaussian $\mathcal{N}\left(\boldsymbol{a} | \boldsymbol{0}, \boldsymbol{K_{AA}}\right)$ with a kernel matrix $\boldsymbol{K_{AA}} \in \mathbb{R}^{M \times M}$, where the entries $\left[\boldsymbol{K_{AA}}\right]_{ij} = \boldsymbol{K}_{A_i A_j}$ correspond to the kernel evaluations $k_{\boldsymbol{\theta}}(A_i, A_j) \in \mathbb{R}$. In particular, the joint distribution $p(\boldsymbol{f}, f_*)$ of the training values $\boldsymbol{f} = f(\boldsymbol{X}) = \left[f(X_1), \dots, f(X_N)\right]^T$ and a test function value $f_* = f(X_*)$ at test point $X_* \in \mathbb{R}^D$ is Gaussian $\mathcal{N}\left(\boldsymbol{0}, \boldsymbol{K}_{[\boldsymbol{X}; X_*][\boldsymbol{X}; X_*]}\right)$, where $[\boldsymbol{X}; X_*]$ is the resulting matrix when stacking the matrices above each other. For GP regression, the Gaussian likelihood $p(\boldsymbol{y} | \boldsymbol{f}) = \mathcal{N}\left(\boldsymbol{y} | \boldsymbol{f}, \sigma_n^2 \mathbb{I}\right)$ can be combined with the joint prior $p(\boldsymbol{f}, f_*)$, so that the predictive posterior distribution can be analytically derived Rasmussen and Williams (2006).

Alternatively, the posterior distribution over the latent variables given the data can be explicitly formulated as

$$p(\boldsymbol{f} | \boldsymbol{y}) \propto p(\boldsymbol{f}, \boldsymbol{y}) = \prod_{j=1}^{J} p\left(\boldsymbol{y}_j | \boldsymbol{f}_j\right) p\left(\boldsymbol{f}_j | \boldsymbol{f}_{1:j-1}\right), \tag{1}$$

where the data is split into J mini-batches of size B, i.e. $\mathcal{D} = \left\{\boldsymbol{y}_j, \boldsymbol{X}_j\right\}_{j=1}^{J}$ with inputs $\boldsymbol{X}_j \in \mathbb{R}^{B \times D}$, outputs $\boldsymbol{y}_j \in \mathbb{R}^B$ and the corresponding latent function values $\boldsymbol{f}_j = f(\boldsymbol{X}_j) \in \mathbb{R}^B$. In (1) we used the notation $\boldsymbol{f}_{k:j}$ indicating $[\boldsymbol{f}_k, \dots, \boldsymbol{f}_j]$ and the conditionals $p\left(\boldsymbol{f}_j | \boldsymbol{f}_{1:j-1}\right)$ can be derived from the joint Gaussian, where we define $p\left(\boldsymbol{f}_1 | \boldsymbol{f}_{1:0}\right) = p(\boldsymbol{f}_1)$. Given the posterior $p(\boldsymbol{f} | \boldsymbol{y})$, the predictive posterior distribution from above is equivalently obtained as $p\left(f_* | \boldsymbol{y}\right) = \int p\left(f_* | \boldsymbol{f}\right) p(\boldsymbol{f} | \boldsymbol{y}) \, \mathrm{d}\boldsymbol{f}$ via Gaussian integration (B7). The corresponding graphical model is depicted in Fig. 1(a)i) and 1(b)i), respectively.

The GP depends via the kernel matrix on the hyperparameters $\boldsymbol{\theta}$, which are typically estimated by maximizing the log marginal likelihood $\log p(\boldsymbol{y} | \boldsymbol{\theta}) = \log \mathcal{N}\left(\boldsymbol{y} | \boldsymbol{0}, \boldsymbol{K_{XX}} + \sigma_n^2 \mathbb{I}\right)$. Although GP inference is an elegant probabilistic approach for regression, the computations for inference and parameter optimization require the inversion of the matrix $\boldsymbol{K_{XX}} + \sigma_n^2 \mathbb{I} \in \mathbb{R}^{N \times N}$, which scales as $\mathcal{O}(N^3)$ in time and $\mathcal{O}(N^2)$ for memory which is infeasible for large N.

2.1 Global sparse GPs

Sparse GP regression approximations based on *global inducing points* reduce the computational complexity by introducing $M_g \ll N$ inducing points $\boldsymbol{a} \in \mathbb{R}^{M_g}$ that optimally summarize the dependency of the whole training data globally, as illustrated in the graphical model in Fig. 1(a)ii) and is denoted in the following as SGP(M_g). Thereby the inducing *inputs* $\boldsymbol{A} \in \mathbb{R}^{M_g \times D}$ are in the D-dimensional input data space and the inducing *outputs* $\boldsymbol{a} = f(\boldsymbol{A}) \in \mathbb{R}^{M_g}$ are the corresponding GP-function values.

Similarly to full GP in Eq. (1), the posterior over the inducing points $p(\boldsymbol{a}|\boldsymbol{y}) \propto \int p(\boldsymbol{a},\boldsymbol{f},\boldsymbol{y})\, \mathrm{d}\boldsymbol{f}$ can be derived from the joint distribution

$$p(\boldsymbol{a},\boldsymbol{f},\boldsymbol{y}) = \prod_{j=1}^{J} p\left(\boldsymbol{y}_j|\boldsymbol{f}_j\right) p\left(\boldsymbol{f}_j|\boldsymbol{a}\right) p(\boldsymbol{a}), \tag{2}$$

where the usual Gaussian likelihood $p\left(\boldsymbol{y}_j|\boldsymbol{f}_j\right) = \mathcal{N}\left(\boldsymbol{f}_j, \sigma_n^2 \mathbb{I}\right)$ and the Gaussian conditional $p\left(\boldsymbol{f}_j|\boldsymbol{a}\right)$ are used. Based on the joint distribution in (2), the posterior $p(\boldsymbol{a}|\boldsymbol{y})$ can be derived from which prediction can be performed using the predictive conditional $p\left(f_*|\boldsymbol{a}\right)$ as more precisely explained in Appendix E.1 and illustrated in Fig. 1(b)ii). Batch inference in these sparse global models can be done in $\mathcal{O}(M_g^2 N)$ time and $\mathcal{O}(M_g N)$ space (Quiñonero-Candela and Rasmussen (2005)).

In order to find optimal inducing inputs $\boldsymbol{A}$ and hyperparameters $\boldsymbol{\theta}$, a sparse variation of the log marginal likelihood similar to full GP can be used Bui et al. (2017), Snelson and Ghahramani (2006), Titsias (2009). For larger datasets, stochastic optimization has been applied e.g., Bui et al. (2017), Hensman et al. (2013), Kania et al. (2021), Schürch et al. (2020) to obtain faster and more data efficient optimization procedures. For recent reviews on the subject we refer to Liu et al. (2020), Quiñonero-Candela and Rasmussen (2005), Rasmussen and Williams (2006).

2.2 Local independent GPs

Local approaches constitute an alternative to global sparse inducing point methods, which exploit multiple local GPs combined with averaging techniques to perform predictions. In this work we focus on *Product of Expert (PoE)* Hinton (2002), where individual predictions from J experts based on the local data $\boldsymbol{y}_j$ are aggregated to the final predictive distribution

$$p\left(f_*|\boldsymbol{y}\right) = \prod_{j=1}^{J} g_j\left(p\left(f_{*j}|\boldsymbol{y}_j\right)\right), \tag{3}$$

where g_j is a function introduced in order to increase or decrease the importance of the experts and depends on the particular PoE method Hinton (2002), Fleet (2014), Tresp (2000), Liu et al. (2018), Liu et al. (2020). Note, in particular, the *generalized PoE (GPoE)*Fleet (2014), where the weights are set to the difference in entropy of the local prior and posterior. The individual predictions $p\left(f_{*j}|\boldsymbol{y}_j\right)$ are based on a local GP, for which the implicit joint posterior can be formulated as

$$p(\boldsymbol{f}|\boldsymbol{y}) \propto p(\boldsymbol{f},\boldsymbol{y}) = \prod_{j=1}^{J} p\left(\boldsymbol{y}_j|\boldsymbol{f}_j\right) p\left(\boldsymbol{f}_j\right), \tag{4}$$

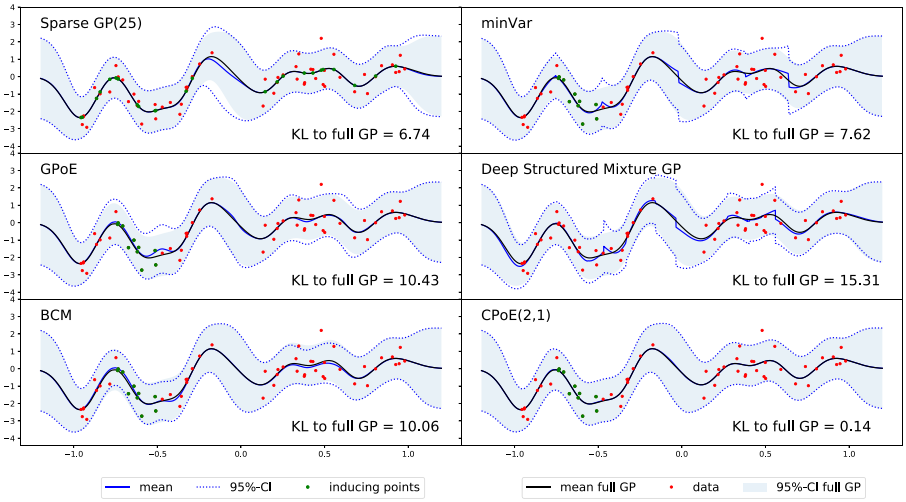

Fig. 2 Different GP approximations (with comparable time complexity) indicated with predictive mean (solid blue) and 95%-credible interval (dotted blue) compared to full GP (black and shaded blue area). The number in the right bottom corner indicates the KL-divergence (B11) to full GP. In the last plot, our method *Correlated Product of Expert (CPoE)* is presented for a degree of correlation $C = 2$ and sparsity $\gamma = 1$. We provide a second example in Figure A6 and a discussion about the relation of our method to *deep structured mixture GPs* (Trapp et al. 2020) is given in Sect. A.5

where the corresponding graphical model is depicted in Fig. 1iii) and more details are provided in Appendix E.2. Other important contributions in this field are distributed local GPs Deisenroth and Ng (2015), parallel hierarchical PoEs Buschjäger et al. (2019), and local experts with consistent aggregations Rullière et al. (2018), Nakai-Kasai and Tanaka (2021). A different category of averaging techniques are for instance *mixture of experts* (Masoudnia and Ebrahimpour 2014; Trapp et al. 2020), which basically replace the product in (3) by a sum. A particularly interesting approach is *deep structured mixtures of GPs* (Trapp et al. 2020), which exploits a sum-product network of local and independent GPs. Moreover, simple baseline methods for local methods are the *minimal variance (minVar)* and the *nearest expert (NE)* aggregation, where only the prediction from the expert with minimal variance and nearest expert is used, respectively. Although both methods show often surprisingly good performance, they suffer from the important disadvantage that there are serious discontinuities at the boundaries between the experts (see for instance Fig. 2) and thus often not useful in practice. This is also the main limitation of all local methods based only on the prediction of one single expert (e.g., deep structured mixture GPs (Trapp et al. 2020)), which was the main reason for introducing smooth PoEs with combined experts. We refer to Liu et al. (2020) for a recent overview.

3 Correlated product of experts

In this section, we present our GP regression method *Correlated Product of Expert* CPoE(C, γ), which is a generalization of the independent PoEs and sparse global GPs. The first generalization is the introduction of correlations between the experts, which can be

adjusted by the parameter $1 \leq C \leq J$ and allows to interpolate between local and global models. Secondly, similar to the sparse global approximation, our method allows to sparsify the inducing points by sparsity parameter $0 < \gamma \leq 1$. We refer to Table 1 in the Appendix for an overview of the used notation.

3.1 Graphical model

Assuming $N = BJ$ data samples which are divided into J ordered partitions (or experts) of size B, i.e. $\mathcal{D} = \left\{\boldsymbol{y}_j, \boldsymbol{X}_j\right\}_{j=1}^{J}$ with inputs $\boldsymbol{X}_j \in \mathbb{R}^{B\times D}$ and outputs $\boldsymbol{y}_j \in \mathbb{R}^B$. We denote $\boldsymbol{f}_j = f(\boldsymbol{X}_j) \in \mathbb{R}^B$ the corresponding latent function values on the GP f. We abbreviate $\boldsymbol{y} = \boldsymbol{y}_{1:J} \in \mathbb{R}^N, \boldsymbol{X} = \boldsymbol{X}_{1:J} \in \mathbb{R}^{N\times D}$ and $\boldsymbol{f} = \boldsymbol{f}_{1:J} \in \mathbb{R}^N$.

Definition 1 (Local inducing points) We refer to local inducing points $\left\{\boldsymbol{a}_j, \boldsymbol{A}_j\right\}_{j=1}^{J}$ with inducing inputs $\boldsymbol{A}_j \in \mathbb{R}^{L\times D}$ and the corresponding inducing outputs $\boldsymbol{a}_j = f(\boldsymbol{A}_j) \in \mathbb{R}^L$ of size $L = \lfloor \gamma B \rfloor$ with $0 < \gamma \leq 1$.

These L local inducing points $\left(\boldsymbol{a}_j, \boldsymbol{A}_j\right)$ of expert j serve as local summary points for the data $\left(\boldsymbol{y}_j, \boldsymbol{X}_j\right)$, where the sparsity level can be adjusted by γ. If $\gamma = 1$, the inducing inputs $\boldsymbol{A}_j$ correspond exactly to $\boldsymbol{X}_j$ and correspondingly $\boldsymbol{a}_j = \boldsymbol{f}_j$. We abbreviate $\boldsymbol{a} = \boldsymbol{a}_{1:J} \in \mathbb{R}^M$, where $M = LJ$, for all local inducing outputs with the corresponding local inducing inputs $\boldsymbol{A} = \boldsymbol{A}_{1:J} \in \mathbb{R}^{M\times D}$. Next, we model connections between the experts by a set of neighbour experts according to the given ordering.

Definition 2 (Predecessor and Correlation Index Sets) Let $\phi_i(j) \in \{1, \ldots, j-1\}$ the index of the ith predecessor of the jth expert. For a given correlation parameter $1 \leq C \leq J$, we introduce the *predecessor set* $\boldsymbol{\pi}_C(j) = \bigcup_{i=1}^{I_j} \phi_i(j)$ satisfying

$$\boldsymbol{\pi}_C(j) \subset \{1, \ldots, j-1\} \qquad \text{and} \qquad \boldsymbol{\pi}_{C+1}(j) = \boldsymbol{\pi}_C(j) \cup \phi_{C+1}(j),$$

such that the size of the set $I_j = |\boldsymbol{\pi}_C(j)| = \min(j-1, C-1)$.

Further, we define the region of correlation with the *correlation indices* as $\boldsymbol{\psi}_C(j) = \boldsymbol{\pi}_C(j) \cup j$ if $j > C$ and $\boldsymbol{\psi}_C(j) = \boldsymbol{\psi}_C(C) = \{1, \ldots, C\}$ otherwise, so that $|\boldsymbol{\psi}_C(j)| = C$ for all j.

The purpose of these predecessor and correlation indices is to model the local correlations among the experts of degree C. If for all j the indices $\boldsymbol{\pi}_C(j)$ are the $C-1$ previous indices, we say that the predecessors are *consecutive* and *non-consecutive* otherwise. If C is clear from the context, $\boldsymbol{\pi}_C(j)$ and $\boldsymbol{\psi}_C(j)$ are abbreviated by $\boldsymbol{\pi}(j)$ and $\boldsymbol{\psi}(j)$, respectively. Details about the specific choices of the ordering, partition, inducing points and predecessor indices are given in Sect. 3.6.1.

Definition 3 (Graph) We define a directed graph $\mathcal{G}(V, E)$ with nodes $V = \boldsymbol{a} \cup \boldsymbol{f} \cup \boldsymbol{y}$ and directed edges

$$E = \{\ \{(\boldsymbol{a}_{\boldsymbol{\pi}_C^i(j)}, \boldsymbol{a}_j)\}_{i=1}^{I_j} \cup \{(\boldsymbol{a}_{\boldsymbol{\psi}_C^i(j)}, \boldsymbol{f}_j)\}_{i=1}^{C} \cup (\boldsymbol{f}_j, \boldsymbol{y}_j)\ \}_{j=1}^{J},$$

where $\boldsymbol{\pi}_C^i(j)$ and $\boldsymbol{\psi}_C^i(j)$ denote the ith element in the corresponding set.

The directed graph $\mathcal{G}$ is depicted in Fig. 4aii), where the local inducing points of the jth expert are connected with the inducing points of the I_j experts in $\boldsymbol{\pi}_C(j)$. Further, the function values $\boldsymbol{f}_j$ are connected in the region of correlation $\boldsymbol{\psi}_C(j)$ to the local inducing points. The graph $\mathcal{G} = (V, E)$ can be equipped with a probabilistic interpretation, in particular, each node $\boldsymbol{v} \in V$ and each incoming edge $(\boldsymbol{v}_i, \boldsymbol{v}) \in E$ for all predecessors $i = 1, \dots, I$ can be interpreted as a conditional probability density $p(\boldsymbol{v}|\boldsymbol{v}_1, \dots, \boldsymbol{v}_I)$.

Proposition 1 (Graphical Model; Proof 1) *We define a graphical model corresponding to the graph $\mathcal{G}(V, E)$ with the conditional probability distributions*

$$p(\boldsymbol{y}_j|\boldsymbol{f}_j) = \mathcal{N}(\boldsymbol{y}_j|\boldsymbol{f}_j, \sigma_n^2 \mathbb{I}), \tag{5}$$

$$p(\boldsymbol{f}_j|\boldsymbol{a}_{\boldsymbol{\psi}(j)}) = \mathcal{N}\left(\boldsymbol{f}_j|\boldsymbol{H}_j \boldsymbol{a}_{\boldsymbol{\psi}(j)}, \overline{\boldsymbol{V}}_j\right) \tag{6}$$

$$p(\boldsymbol{a}_j|\boldsymbol{a}_{\boldsymbol{\pi}(j)}) = \mathcal{N}(\boldsymbol{a}_j|\boldsymbol{F}_j \boldsymbol{a}_{\boldsymbol{\pi}(j)}, \boldsymbol{Q}_j), \tag{7}$$

where (5) is the usual Gaussian likelihood for GP regression with noise variance σ_n^2, (6) the projection conditional and (7) the prior transition. Thereby, the matrices are defined as $\boldsymbol{H}_j = \boldsymbol{K}_{\boldsymbol{X}_j \boldsymbol{A}_{\boldsymbol{\psi}(j)}} \boldsymbol{K}^{-1}_{\boldsymbol{A}_{\boldsymbol{\psi}(j)} \boldsymbol{A}_{\boldsymbol{\psi}(j)}} \in \mathbb{R}^{B \times LC}$, $\overline{\boldsymbol{V}}_j = Diag[\boldsymbol{K}_{\boldsymbol{X}_j \boldsymbol{X}_j} - \boldsymbol{K}_{\boldsymbol{X}_j \boldsymbol{A}_{\boldsymbol{\psi}(j)}} \boldsymbol{K}^{-1}_{\boldsymbol{A}_{\boldsymbol{\psi}(j)} \boldsymbol{A}_{\boldsymbol{\psi}(j)}} \boldsymbol{K}_{\boldsymbol{A}_{\boldsymbol{\psi}(j)} \boldsymbol{X}_j}] \in \mathbb{R}^{B \times B}$, $\boldsymbol{F}_j = \boldsymbol{K}_{\boldsymbol{A}_j \boldsymbol{A}_{\boldsymbol{\pi}(j)}} \boldsymbol{K}^{-1}_{\boldsymbol{A}_{\boldsymbol{\pi}(j)} \boldsymbol{A}_{\boldsymbol{\pi}(j)}} \in \mathbb{R}^{L \times LI_j}$, *and* $\boldsymbol{Q}_j = \boldsymbol{K}_{\boldsymbol{A}_j \boldsymbol{A}_j} - \boldsymbol{K}_{\boldsymbol{A}_j \boldsymbol{A}_{\boldsymbol{\pi}(j)}} \boldsymbol{K}^{-1}_{\boldsymbol{A}_{\boldsymbol{\pi}(j)} \boldsymbol{A}_{\boldsymbol{\pi}(j)}} \boldsymbol{K}_{\boldsymbol{A}_{\boldsymbol{\pi}(j)} \boldsymbol{A}_j} \in \mathbb{R}^{L \times L}$ *with* $\boldsymbol{F}_1 = \boldsymbol{0}$ *and* $\boldsymbol{Q}_1 = \boldsymbol{K}_{\boldsymbol{A}_1 \boldsymbol{A}_1}$.

The two conditional distributions (6) and (7) can be derived from the true joint prior distribution $p(\boldsymbol{a}, \boldsymbol{f}, \boldsymbol{y})$ as shown in Proof 1. Alternatively, a generalization of this model can be obtained when using a modified projection distribution $p(\boldsymbol{f}_j|\boldsymbol{a}_{\boldsymbol{\psi}(j)})$, so that for $C \to J$ and $\gamma < 1$ our model recovers a range of well known global sparse GP methods as described in Sect. A.1 and Prop. 5. In any case, these local conditional distributions lead to the following joint distribution.

Definition 4 (Joint distribution) For the graphical model corresponding to graph $\mathcal{G}$, the joint distribution over all variables $\boldsymbol{f}, \boldsymbol{a}, \boldsymbol{y}$ can be written as

$$q_{c,\gamma}(\boldsymbol{f}, \boldsymbol{a}, \boldsymbol{y}) = \prod_{j=1}^{J} p(\boldsymbol{y}_j|\boldsymbol{f}_j) p(\boldsymbol{f}_j|\boldsymbol{a}_{\boldsymbol{\psi}(j)}) p(\boldsymbol{a}_j|\boldsymbol{a}_{\boldsymbol{\pi}(j)}).$$

In the case $\gamma = 1$ and thus $\boldsymbol{a} = \boldsymbol{f}$, the joint distribution simplifies (Proof 2) to

$$q_{c,1}(\boldsymbol{f}, \boldsymbol{y}) = \prod_{j=1}^{J} p(\boldsymbol{y}_j|\boldsymbol{f}_j) p(\boldsymbol{f}_j|\boldsymbol{f}_{\boldsymbol{\pi}(j)}).$$

We use $q = q_{c,\gamma}$ instead of p in order to indicate that it is an approximate distribution. The joint distributions in Def. 4 and the corresponding graphical model in Fig. 4a allow interesting comparisons to other GP models in Fig. 1 and the corresponding formulas (1), (2), (4). Whereas the conditioning set for full GP are all the previous latent values $\boldsymbol{f}_{1:j-1}$, for sparse GPs some global inducing points $\boldsymbol{a}$ and for local independent experts the empty set, we propose to condition on the $C - 1$ predecessors $\boldsymbol{f}_{\boldsymbol{\pi}(j)}$ (or a sparsified version in the general case). From this point of view, we can notice that our probabilistic

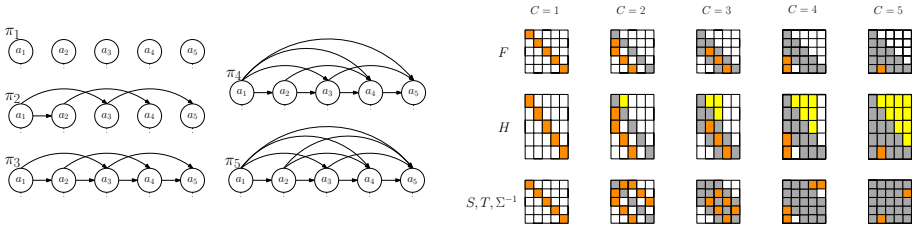

Fig. 3 Correlation structure $\boldsymbol{\pi}_C$ between the $J = 5$ experts for different degrees of correlation $1 \leq C \leq J$. Left: Graphical model among the local inducing points $\boldsymbol{a}_j$. Right: Structure of sparse transition matrix $\boldsymbol{F}$, projection matrix $\boldsymbol{H}$, prior precision $\boldsymbol{S}$, likelihood precision $\boldsymbol{T}$ and posterior precision $\boldsymbol{\Sigma}^{-1}$. Note that $\boldsymbol{\pi}_C$ does not have to be consecutive, e.g $2 \notin \boldsymbol{\pi}_2(3)$

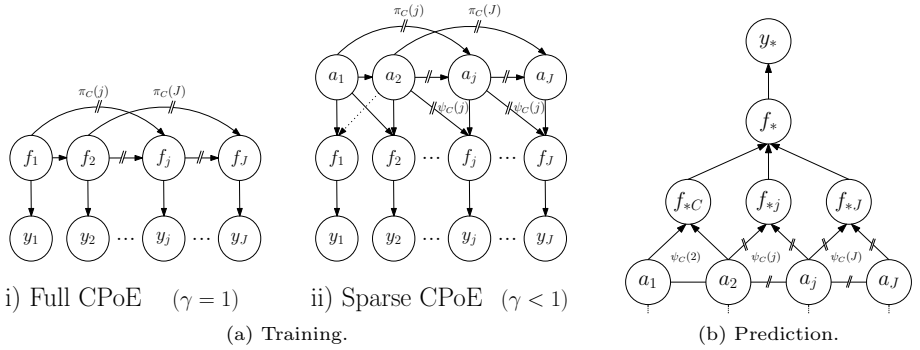

Fig. 4 Graphical model for training and prediction of CPoE(C, γ)

model is equal to full GP, sparse GP and PoEs under certain circumstances, which are more precisely formulated in Prop. 5.

3.2 Sparse and local prior approximation

The conditional independence assumptions between the experts induced by the predecessor structure $\boldsymbol{\pi}_C$ lead to an approximate prior $q_{c,\gamma}(\boldsymbol{a})$ and approximate projection $q_{c,\gamma}(\boldsymbol{f}|\boldsymbol{a})$ yielding a sparse and local joint prior $q_{c,\gamma}(\boldsymbol{a},\boldsymbol{f},\boldsymbol{y})$.

Proposition 2 (Joint prior approximation, Proof 4) *The prior over all local inducing points* $\boldsymbol{a}$ *in our CPoE model is*

$$q_{c,\gamma}(\boldsymbol{a}) = \prod_{j=1}^{J} p\big(\boldsymbol{a}_j|\boldsymbol{a}_{\boldsymbol{\pi}(j)}\big) = \mathcal{N}\big(\boldsymbol{a}|\boldsymbol{0}, \boldsymbol{S}_C^{-1}\big),$$

with prior precision $\boldsymbol{S}_C = \boldsymbol{S} = \boldsymbol{F}^T\boldsymbol{Q}^{-1}\boldsymbol{F} \in \mathbb{R}^{M\times M}$, *where* $\boldsymbol{Q} = \mathrm{Diag}\big[\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_J\big] \in \mathbb{R}^{M\times M}$ *and* $\boldsymbol{F} \in \mathbb{R}^{M\times M}$ *is given as the sparse lower triangular matrix in Fig. 5. Moreover, the projection is*

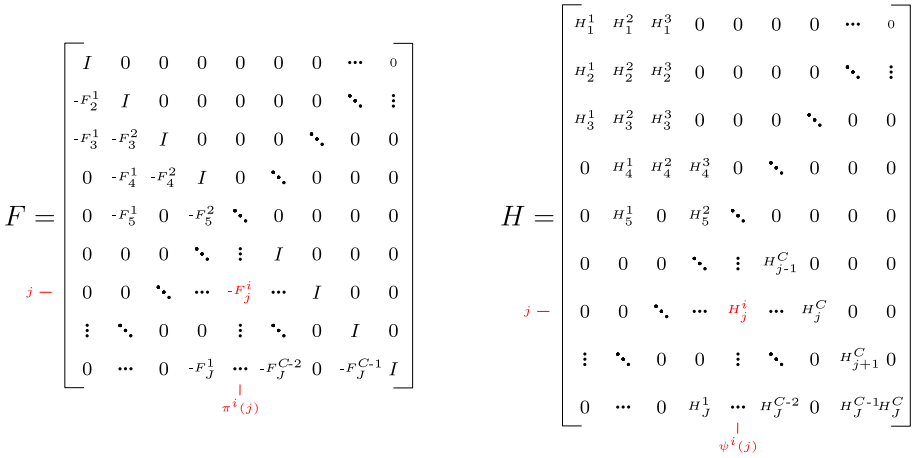

Fig. 5 Sparse transition $\boldsymbol{F} \in \mathbb{R}^{M\times M}$ and projection $\boldsymbol{H} \in \mathbb{R}^{N\times M}$ matrices, where $\boldsymbol{F}_j^i \in \mathbb{R}^{L\times L}$ and $\boldsymbol{H}_j^i \in \mathbb{R}^{B\times L}$ are the ith part of $\boldsymbol{F}_j \in \mathbb{R}^{L\times L(C-1)}$ and $\boldsymbol{H}_j \in \mathbb{R}^{B\times LC}$, respectively, corresponding to the ith entries in $\boldsymbol{\pi}^i(j)$ and $\boldsymbol{\psi}^i(j)$

$$q_{c,\gamma}(\boldsymbol{f}|\boldsymbol{a}) = \prod_{j=1}^{J} p\left(\boldsymbol{f}_j|\boldsymbol{a}_{\boldsymbol{\psi}(j)}\right) = \mathcal{N}\left(\boldsymbol{f}|\boldsymbol{H}\boldsymbol{a}, \overline{\boldsymbol{V}}\right),$$

where $\boldsymbol{H} \in \mathbb{R}^{N\times M}$ *defined in Fig. 5 and* $\overline{\boldsymbol{V}} = \mathrm{Diag}\left[\overline{\boldsymbol{V}}_1, \ldots, \overline{\boldsymbol{V}}_J\right] \in \mathbb{R}^{N\times N}$. *Together with the exact likelihood* $p(\boldsymbol{y}|\boldsymbol{f}) = \prod_{j=1}^{J} p(\boldsymbol{y}_j|\boldsymbol{f}_j) = \mathcal{N}\left(\boldsymbol{y}|\boldsymbol{f}, \sigma_n^2\mathbb{I}\right)$ *determines the joint approximate prior*

$$q_{c,\gamma}(\boldsymbol{a},\boldsymbol{f},\boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{f})\, q_{c,\gamma}(\boldsymbol{f}|\boldsymbol{a})\, q_{c,\gamma}(\boldsymbol{a}).$$

Note that the joint prior $q_{c,\gamma}(\boldsymbol{a},\boldsymbol{f},\boldsymbol{y})$ is Gaussian $\mathcal{N}(\boldsymbol{0},\boldsymbol{W})$ with dense covariance $\boldsymbol{W}$ and sparse precision $\boldsymbol{Z} = \boldsymbol{W}^{-1}$ as shown in Fig. C7 in the Appendix. If the predecessor set is consecutive, the matrix $\boldsymbol{F}$ is a lower band (block)matrix with bandwidth C and in the non-consecutive case each row has exactly C non-zero blocks. The sparsity pattern of $\boldsymbol{F}$ is inherited to the prior precision $\boldsymbol{S} = \boldsymbol{F}^T\boldsymbol{Q}^{-1}\boldsymbol{F}$, which is also a sparse matrix (see Fig. 3). For the consecutive case, $\boldsymbol{S}$ is a block-band matrix with bandwidth $2C-1$. Note that, the inverse $\boldsymbol{S}^{-1}$ is dense. The likelihood matrix $\boldsymbol{H}$ is exact in the corner up to indices C which ensures that our model recovers sparse global GP in the limiting case $C = J$. The quality of the approximation of our CPoE(C,γ) model is discussed in Sect. 3.5, where we show that $q_{c,\gamma}(\boldsymbol{a},\boldsymbol{f},\boldsymbol{y})$ converges to the true prior $p(\boldsymbol{a},\boldsymbol{f},\boldsymbol{y})$ for $C \to J$.

3.3 Inference

For our model it is possible to infer analytically the posterior $q_{c,\gamma}(\boldsymbol{a}|\boldsymbol{y})$ and the marginal likelihood $q_{c,\gamma}(\boldsymbol{y})$ used later for prediction and for hyperparameter estimation, respectively.

Proposition 3 (Posterior approximation; Proof 12) *From the joint distribution, the latent function values* $\boldsymbol{f}$ *can be integrated out yielding*

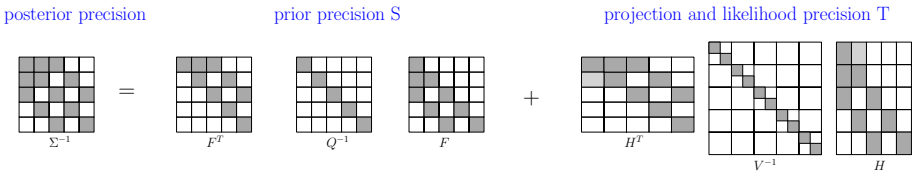

Fig. 6 Sparse posterior precision approximation

$$q_{c,\gamma}(\boldsymbol{a},\boldsymbol{y}) = \int q_{c,\gamma}(\boldsymbol{f},\boldsymbol{a},\boldsymbol{y})\,\mathrm{d}\boldsymbol{f} = q_{c,\gamma}(\boldsymbol{y}|\boldsymbol{a})q_{c,\gamma}(\boldsymbol{a}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{H}\boldsymbol{a},\boldsymbol{V})\mathcal{N}\big(\boldsymbol{a}|\boldsymbol{0},\boldsymbol{S}^{-1}\big)$$

with $\boldsymbol{V} = \overline{\boldsymbol{V}} + \sigma_n^2\mathbb{I} \in \mathbb{R}^{N\times N}$. *The posterior can be analytically computed by*

$$q_{c,\gamma}(\boldsymbol{a}|\boldsymbol{y}) = \frac{q_{c,\gamma}(\boldsymbol{a},\boldsymbol{y})}{q_{c,\gamma}(\boldsymbol{y})} \propto q_{c,\gamma}(\boldsymbol{a},\boldsymbol{y}) = \mathcal{N}(\boldsymbol{a}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \mathcal{N}^{-1}(\boldsymbol{a}|\boldsymbol{\eta},\boldsymbol{\Lambda}),$$

with $\boldsymbol{\Sigma}^{-1} = \boldsymbol{\Lambda} = \boldsymbol{T} + \boldsymbol{S} \in \mathbb{R}^{M\times M}$, $\boldsymbol{\mu} = \boldsymbol{\Sigma}\boldsymbol{\eta} \in \mathbb{R}^M$, $\boldsymbol{\eta} = \boldsymbol{H}^T\boldsymbol{V}^{-1}\boldsymbol{y} \in \mathbb{R}^M$ *and* $\boldsymbol{T} = \boldsymbol{H}^T\boldsymbol{V}^{-1}\boldsymbol{H} \in \mathbb{R}^{M\times M}$.

The posterior precision matrix $\boldsymbol{\Sigma}^{-1} = \boldsymbol{T} + \boldsymbol{S}$ inherits the sparsity pattern of the prior, since the addition of the projection precision $\boldsymbol{T} = \boldsymbol{H}^T\boldsymbol{V}^{-1}\boldsymbol{H}$ has the same sparsity structure, as depicted in Figs. 3 and 6. On the other hand, the posterior covariance $\boldsymbol{\Sigma}$ is dense, therefore it will be never explicitly fully computed. Instead, the sparse linear system of equations $\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} = \boldsymbol{\eta}$ can be efficiently solved for $\boldsymbol{\mu} = \boldsymbol{\Sigma}\boldsymbol{\eta}$. Further, in our CPoE model, the marginal likelihood $q_{c,\gamma}(\boldsymbol{y}|\boldsymbol{\theta})$ can be analytically computed by $\int q_{c,\gamma}(\boldsymbol{y},\boldsymbol{a})\,\mathrm{d}\boldsymbol{a} = \mathcal{N}(\boldsymbol{0},\boldsymbol{P})$ (see Proof 9) with the (dense) matrix $\boldsymbol{P} = \boldsymbol{H}\boldsymbol{S}^{-1}\boldsymbol{H}^T + \boldsymbol{V} \in \mathbb{R}^{N\times N}$, which is used in Sect. 3.6.2 for hyperparameter optimization. The posterior approximation $q_{c,\gamma}(\boldsymbol{a}|\boldsymbol{y})$ as well as the approximate marginal likelihood $q_{c,\gamma}(\boldsymbol{y})$ converge to the true distributions $p(\boldsymbol{a}|\boldsymbol{y})$ and $p(\boldsymbol{y})$, respectively, for $C \to J$. In particular, they correspond exactly to the posterior and marginal likelihood of full GP and sparse global GP with $\lfloor\gamma N\rfloor$ inducing points for $C = J, \gamma = 1$ and $C = J, \gamma < 1$, respectively.

3.4 Prediction

The final predictive posterior distribution is obtained by an adaptation of the PoE aggregation in (3). The main idea is to consistently aggregate weighted local predictions form the experts, such that the correlations between them are taken into account resulting in a smooth and continuous predictive distribution.

Proposition 4 (Prediction aggregation; Proof 17) *Similarly to the PoE aggregation* (3), *we define the final predictive posterior distribution for a query point* $\boldsymbol{x}_* \in \mathbb{R}^D$ *as*

$$q_{c,\gamma}(f_*|\boldsymbol{y}) = \prod_{j=C}^{J} q_{c,\gamma}(f_{*j}|\boldsymbol{y})^{\beta_{*j}}, \tag{8}$$

involving the local predictions $q_{c,\gamma}(f_{*j}|\boldsymbol{y}) = \mathcal{N}\big(m_{*j}, v_{*j}\big)$ *and weights* $\beta_{*j} \in \mathbb{R}$ *defined in Prop.* 8 *and Def.* 5, *respectively. Moreover, the distribution* $q_{c,\gamma}(f_*|\boldsymbol{y}) = \mathcal{N}\big(m_*, v_*\big)$ *with*

$m_* = v_* \sum_{j=C}^{J} \beta_{*j} \frac{m_{*j}}{v_{*j}}$ *and* $\frac{1}{v_*} = \sum_{j=C}^{J} \frac{\beta_{*j}}{v_{*j}}$ *is analytically available. The final noisy prediction is* $p(y_*|\boldsymbol{y}) = \mathcal{N}(m_*, v_* + \sigma_n^2)$.

The graphical model corresponding to this prediction procedure is depicted in Fig. 4b and A3 in the Appendix. Further, the local predictions $q_{c,\gamma}(f_{*j}|\boldsymbol{y})$ in Equation (8) are based on the region $\boldsymbol{\psi}(j)$, where the correlations are modelled and can be computed as $q_{c,\gamma}(f_{*j}|\boldsymbol{y}) = \int p(f_{*j}|\boldsymbol{a}_{\boldsymbol{\psi}(j)}) q_{c,\gamma}(\boldsymbol{a}_{\boldsymbol{\psi}(j)}|\boldsymbol{y}) \, \mathrm{d}\boldsymbol{a}_{\boldsymbol{\psi}(j)}$ involving the local posteriors $q_{c,\gamma}(\boldsymbol{a}_{\boldsymbol{\psi}(j)}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\psi}(j)}, \boldsymbol{\Sigma}_{\boldsymbol{\psi}(j)})$ and the predictive conditional $p(f_{*j}|\boldsymbol{a}_{\boldsymbol{\psi}(j)})$, as thoroughly shown in Proposition 8 in the Appendix. Thereby, the local posteriors with mean $\boldsymbol{\mu}_{\boldsymbol{\psi}(j)}$ and covariance entries $\boldsymbol{\Sigma}_{\boldsymbol{\psi}(j)}$ could be obtained from the corresponding entries $\boldsymbol{\psi}(j)$ of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. However, computing explicitly some entries in the dense covariance $\boldsymbol{\Sigma}$ based on the sparse precision $\boldsymbol{\Sigma}^{-1}$ is not straightforward since in the inverse the blocks are no longer independent. However, we can exploit the particular sparsity and block-structure of our precision matrix and obtain an efficient implementation of this part, which is key to achieve a competitive performance of our algorithm. More details are given in the Appendix in Sect. A.2.

Definition 5 (Aggregation weights) The input depending weights $\beta_{*j} = \beta_j(X_*)$ at query point X_* model the impact of expert j. In particular, the unnormalized weights

$$\bar{\beta}_{*j} = H[p(f_*)] - H[p(f_{*j}|\boldsymbol{y})] = \frac{1}{2} \log\left(\frac{v_{*0}}{v_{*j}}\right),$$

are set to the difference in entropy H (B10) before and after seeing the data similarly proposed by Fleet (2014). Thereby, the predictive prior is $p(f_*) = \mathcal{N}(0, v_{*0})$ with $v_{*0} = \boldsymbol{k}_{X_*X_*}$ and the predictive posterior defined in Prop. 8. The normalized weights are then obtained by $\beta_{*j} = b^{-1}\bar{\beta}_{*j}^Z$ where $b = \sum_{j=C}^{J} \bar{\beta}_{*j}^Z$ and $Z = \log(N)C$.

These weights bring the flexibility of increasing or reducing the importance of the experts based on the predictive uncertainty. However, independent of the particular weights, our aggregation of the predictions is consistent since it is based on the *covariance intersection* method (Julier and Uhlmann 1997), which is useful for combining several estimates of random variables with known mean and variance but unknown correlation between them.

3.5 Properties

Proposition 5 (Equality; Proof 3) *Our model correlated Product of Experts CPoE*(C, γ) *is equal to full GP for* $C = J$ *and* $\gamma = 1$. *For* $\gamma < 1$, *our model correspond to sparse global GP with* $M_g = \lfloor \gamma N \rfloor$ *inducing points. Further, with* $C = 1$ *and* $\gamma = 1$, *our model is equivalent to independent PoEs. That is, we have*

$$\mathrm{CPoE}(J, 1) = \mathrm{GP}; \quad \mathrm{CPoE}(J, \gamma) = \mathrm{SGP}(\lfloor \gamma N \rfloor); \quad \mathrm{CPoE}(1, 1) = \mathrm{GPoE}^*,$$

where SGP refers to the FITC model Snelson and Ghahramani (2006) and GPoE correspond to GPoE Fleet (2014) with slightly different weights ($Z = 1$) *in the prediction.*

In Sect. A.1 in the Appendix we present a generalization of our model, so that CPoE(J, γ) correspond to a range of other well known versions of sparse global GP by changing the projection distribution and adding a correction term in the log marginal likelihood similarly discussed in Schürch et al. (2020) for the global case. For instance, we can extend our model analogously to the variational version of Titsias (2009).

For correlations between the limiting cases $C = 1$ and $C = J$, we investigate the difference in KL of the true GP model with CPoE(C, γ) and CPoE(C_2, γ) for $1 \leq C \leq C_2 \leq J$. For that reason, we define the difference in KL between the true distribution of $\boldsymbol{x}$ and two different approximate distributions, i.e.

$$\mathbb{D}_{(C,C_2)}[\boldsymbol{x}] = KL[p(\boldsymbol{x}) \mid\mid q_{c,\gamma}(\boldsymbol{x})] - KL[p(\boldsymbol{x}) \mid\mid q_{c_2,\gamma}(\boldsymbol{x})].$$

Similarly, the difference in KL for a conditional distribution is defined in Eq. (B15). Using these definitions, we show that the approximation quality of the prior $q_{c,\gamma}(\boldsymbol{a})$ and projection approximation $q_{c,\gamma}(\boldsymbol{f}|\boldsymbol{a})$ monotonically improves for $C \to J$, so that the KL between the true joint distribution $p(\boldsymbol{a},\boldsymbol{f},\boldsymbol{y})$ and our approximate joint distribution $q_{c,\gamma}(\boldsymbol{a},\boldsymbol{f},\boldsymbol{y})$ is decreasing for $C \to J$.

Proposition 6 (Decreasing KL; Proof 6) *For any predecessor structure $\boldsymbol{\pi}_C$ and any $0 < \gamma \leq 1$ and $1 \leq C \leq C_2 \leq J$, the difference in KL of the marginal prior, projection and data likelihood are non negative, i.e.*

$$\mathbb{D}_{(C,C_2)}[\boldsymbol{a}] \geq 0, \qquad \mathbb{D}_{(C,C_2)}[\boldsymbol{f}|\boldsymbol{a}] \geq 0, \qquad \mathbb{D}_{(C,C_2)}[\boldsymbol{y}|\boldsymbol{f}] = 0,$$

so that the joint difference in KL is also non-negative

$$\mathbb{D}_{(C,C_2)}[\boldsymbol{a},\boldsymbol{f},\boldsymbol{y}] = \mathbb{D}_{(C,C_2)}[\boldsymbol{a}] + \mathbb{D}_{(C,C_2)}[\boldsymbol{f}|\boldsymbol{a}] + \mathbb{D}_{(C,C_2)}[\boldsymbol{y}|\boldsymbol{f}] \geq 0.$$

Moreover, we can quantify the approximation quality, in particular $\mathbb{D}_{(C,C_2)}[\boldsymbol{a}] = \frac{1}{2}\log\frac{|\boldsymbol{Q}_C|}{|\boldsymbol{Q}_{C_2}|}$ *and* $\mathbb{D}_{(C,C_2)}[\boldsymbol{f}|\boldsymbol{a}] = \frac{1}{2}\log\frac{|\tilde{\boldsymbol{V}}_C|}{|\tilde{\boldsymbol{V}}_{C_2}|}$.

The last statement demonstrates that our CPoE model is a sound GP prior precision approximation, which converges monotonically to the true prior for $C \to J$. The decreasing KL of the joint prior is depicted in Fig. 7 together with the decreasing KL of the posterior, marginal likelihood and predictive posterior. More details and proofs are given in Appendix C.

3.6 Computational details

3.6.1 Graph

The graphical model in Sect. 3.1 is generically defined and several choices are left for completely specifying the graph $\mathcal{G}(V, E)$ for a particular dataset: the partition method, the ordering of the partition, the selection of the predecessors and the local inducing points. We tried to make these choices as simple and straightforward as possible with focus on computational efficiency, however, there might be more sophisticated heuristics. Concretely, we use KD-trees Maneewongvatana and Mount (2001) for partitioning the data $\mathcal{D}$ into J regions and the ordering starts with a random partition which is then greedily extended by the closest partition in euclidean distance (represented by the mean of the

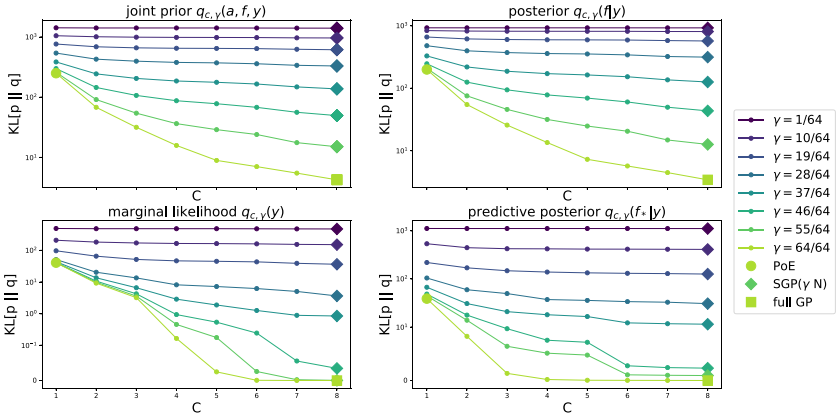

Fig. 7 Decreasing KL$[p||q]$ between true distribution p of full GP and approximate distribution $q = q_{c,\gamma}$ of CPoE for increasing values of C and γ for the joint prior, posterior, marginal likelihood and predictive posterior for synthetic GP data ($N = 1024, D = 2$, SE kernel)

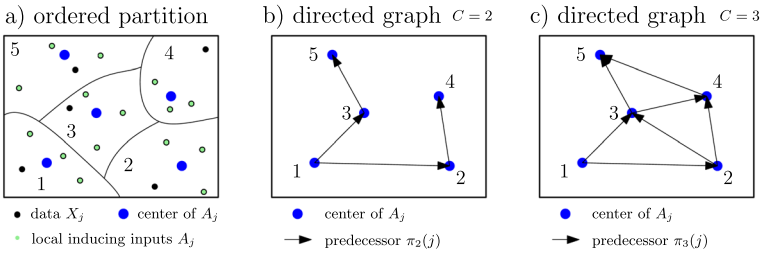

Fig. 8 Toy example for partition, local inducing points, predecessors and directed graph illustrated for $D = 2$ with $J = 5$ experts/partitions each with $B = 4$ samples, $\gamma = 0.75$ and thus $L = 3$ local inducing points. In a) the ordered partition with the data (black), local inducing points (green) and their mean (blue) are depicted. In b) and c) the directed graph for $C = 2$ and $C = 3$ are shown with corresponding predecessors $\boldsymbol{\pi}_2(1) = \{\}$, $\boldsymbol{\pi}_2(2) = \{1\}$, $\boldsymbol{\pi}_2(3) = \{1\}$, $\boldsymbol{\pi}_2(4) = \{2\}$, $\boldsymbol{\pi}_2(5) = \{3\}$ and $\boldsymbol{\pi}_3(1) = \{\}$, $\boldsymbol{\pi}_3(2) = \{1\}$, $\boldsymbol{\pi}_3(3) = \{1, 2\}$, $\boldsymbol{\pi}_3(4) = \{2, 3\}$, $\boldsymbol{\pi}_3(5) = \{3, 4\}$, respectively. In the previous example, $\boldsymbol{\pi}_3$ is consecutive and $\boldsymbol{\pi}_2$ is non-consecutive

inducing points). The $L \leq B$ inducing inputs $\boldsymbol{A}_j \in \mathbb{R}^{L \times D}$ of the jth partition (or expert) can be in principle arbitrary, however, in this work they are chosen as a random subset of the data inputs $\boldsymbol{X}_j \in \mathbb{R}^{B \times D}$ of the jth expert for the sake of simplicity. For the predecessors (block-)indices $\boldsymbol{\pi}_C$, the $C - 1$ closest partitions among the previous (according to the ordering) predecessors in euclidean distance are greedily selected. These concepts are illustrated for a toy example in Fig. 8.

3.6.2 Hyperparameter estimation

In Sect. 3, we introduced CPoE for fixed hyperparameters $\boldsymbol{\theta}$ where implicitly all distributions are conditioned on $\boldsymbol{\theta}$, however, we omitted the dependencies on $\boldsymbol{\theta}$ in the most cases for the sake of brevity. Similar to full GP, sparse GP or PoEs, the *log marginal likelihood (LML)* can be used as an objective function for optimizing the few hyperparameters $\boldsymbol{\theta}$. The log of the marginal likelihood of our model formulated in Sect. 3.3 is

Table 1 Complexity for training, pointwise predictions for N_t points and number of optimization parameters for different GP algorithms

	Full GP	Sparse GP	PoE	CPoE
Time	$\mathcal{O}(N^3)$	$\mathcal{O}(NM_g^2)$	$\mathcal{O}(NB^2)$	$\mathcal{O}(NB^2\alpha^3)$
Space	$\mathcal{O}(N^2)$	$\mathcal{O}(NM_g)$	$\mathcal{O}(NB)$	$\mathcal{O}(NB\alpha^2)$
Time_t	$\mathcal{O}(N^2N_t)$	$\mathcal{O}(M_g^2N_t)$	$\mathcal{O}(NBN_t)$	$\mathcal{O}(NBN_t\alpha^2)$
Space_t	$\mathcal{O}(NN_t)$	$\mathcal{O}(M_gN_t)$	$\mathcal{O}(NN_t)$	$\mathcal{O}(NN_t\alpha)$
#pars	$\lvert\boldsymbol{\theta}\rvert$	$MD + \lvert\boldsymbol{\theta}\rvert$	$\lvert\boldsymbol{\theta}\rvert$	$\lvert\boldsymbol{\theta}\rvert$

$\mathcal{L}(\boldsymbol{\theta}) = \log q(\boldsymbol{y}|\boldsymbol{\theta}) = \log \mathcal{N}(\mathbf{0}, \boldsymbol{P})$ with $\boldsymbol{P} = \boldsymbol{H}\boldsymbol{S}^{-1}\boldsymbol{H}^T + \boldsymbol{V}$ which can be efficiently computed as detailed in Sect. A.3 and can be used for *deterministic optimization* with full batch $\boldsymbol{y}$ for moderate sample size N. However, in order to scale this parameter optimization part to larger number of samples N in a competitive time, *stochastic optimization* techniques exploiting subsets of data have to be developed similarly done for the global sparse GP model (SVI Hensman et al. (2013); REC Schürch et al. (2020); IF Kania et al. (2021)). We adapt the hybrid approach IF of Kania et al. (2021) where we can also exploit an independent factorization of the log marginal likelihood which decomposes into a sum of J terms, so that it can be used for stochastic optimization. This constitutes a very fast and accurate alternative for our method as shown in the Appendix A.3 and will also be exploited in Sect. 4 for large data sets. Alternatively to the log marginal likelihood (LML) maximization as presented above, the *maximum a posteriori* (MAP) estimator for $\boldsymbol{\theta}$ can be used. This means, that some suitable prior on the hyperparameters are introduced, as explained in Sect. A.3.3 and an example is presented in Sect. 4.4.

3.6.3 Complexity

The time complexity for computing the posterior and the marginal likelihood in our algorithm is dominated by J operations which are cubic in LC (inversion, matrix-matrix multiplication, determinants). This leads to $\mathcal{O}(NB^2\alpha^3)$ and $\mathcal{O}(NB\alpha^2)$ for time and space complexity, respectively, where we define the approximation quality parameter $\alpha = C\gamma$. Similarly, for N_t testing points the time and space complexities are $\mathcal{O}(NB\alpha^2N_t)$ and $\mathcal{O}(N\alpha N_t)$ (an approach to remove the dependency of N is outlined in A.4). In Table 1, the asymptotic complexities of our model together with other GP algorithms are indicated. It is interesting that for $\alpha = 1$, our algorithm has the same asymptotic complexity for training as sparse global GP with $M_g = B$ global inducing points but we can have $M = LJ = \gamma BJ = \gamma N$ total local inducing points! Thus, our approach allows much more total local inducing points M in the order of N (e.g., $M = 0.5N$ with $C = 2$) whereas for sparse global GP usually $M_g \ll N$. This has the consequence that the local inducing points can cover the input space much better and therefore represent much more complicated functions. As a consequence, there is also no need to optimize the local inducing points resulting in much fewer parameters to optimize. Consider the following example with $N = 10'000$ in $D = 10$ dimensions. Suppose a sparse global GP model with $M_g = 500$ global inducing points. A CPoE model with the same asymptotic complexity has a batch size $B = M_g = 500$ and $\alpha = 1$. Therefore, we have $J = \frac{N}{B} = 20$ experts and we choose $C = 2$ and $\gamma = \frac{1}{2}$ such that we obtain $L = \gamma B = 250$ local inducing points per experts and $M = \gamma N = 5'000$ total local inducing points! Further, the number of hyperparameters to optimize with a SE kernel is for global sparse GP $M_gD + |\boldsymbol{\theta}| = 5012$, whereas for CPoE there are only $|\boldsymbol{\theta}| = 12$. For an extended version of this section consider A.4 in the Appendix.

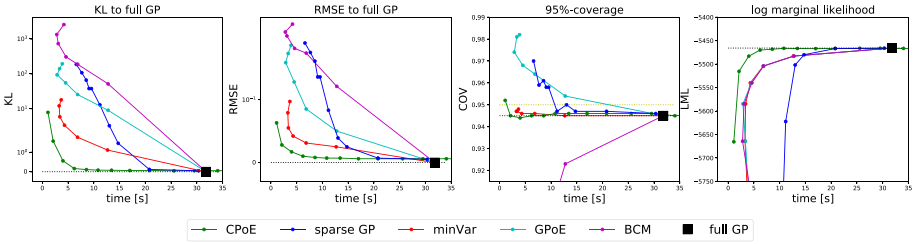

Fig. 9 Average accuracy versus time performance of different GP algorithms

4 Comparison

In this section, we compare the performance with competitor methods for GP approximations using *synthetic* and several *real world* datasets as summarized in Table 3. Moreover, we provide a comparison to non-GP regression methods as well as an application about probabilistic time series prediction both exploiting non-trivial kernels. More details about the experiments and implementations are provided in Sects. A.6, A.7 and F in the Appendix.

4.1 Synthetic data

First, we examine the accuracy vs. time performance of different GP algorithms for fixed hyperparameters in a simulation study with *synthetic GP data*. We generated $N = 8192$ data samples in $D = 2$ with 5 repetitions from the sum of two SE kernels with a shorter and longer lengthscale such that both global and local patterns are present in the data (compare Fig. A5). In Fig. 9 the mean results are shown for the KL and RMSE to full GP, the 95%-coverage and the log marginal likelihood against time in seconds. The results for sparse GP with increasing number of global inducing points M are shown in blue, the results for minVar, GPoE and BCM for increasing number of experts J are depicted in red, cyan and magenta, respectively. For CPoE, the results for increasing correlations C are shown in green. We observe superior performance of our method compared to competitors in terms of accuracy compared to full GP versus time. Moreover, one can observe that the confidence information of our model are reliable already for small approximation orders since it is based on the consistent covariance intersection method. A precise description of the experiment is provided in Sect. A.7.1 in the Appendix.

4.2 Real world data

Second, we benchmark our method with 10 real world datasets as summarized in Table 3 and more details are given in Sect. A.7.2 in the Appendix (e.g., how to access and preprocess the data). For the 5 smaller datasets in the first block we use deterministic parameter optimization for which the average results over 10 training/testing splits are depicted in Table 2. In particular, the KL to full GP (left) and time (right) for different GP methods are shown. Similarly, the average accuracy and times for the 4 larger datasets in the second

Table 2 Average KL to full GP (left) and time (right) for different GP methods and 5 datasets with 10 repetitions. More results are provided in Appendix F

	KL					Time				
	Concrete	mg	Space	Abalone	Kin	Concrete	mg	Space	Abalone	Kin
fullGP	0.0	0.0	0.0	0.0	0.0	7.3	25.5	114.8	237.9	161.5
SGP(100)	352.9	9.9	108.1	15.6	603.7	36.4	14.4	46.6	58.9	42.2
minVar	122.2	19.4	63.6	25.1	211.0	1.5	2.0	7.2	6.4	9.3
GPoE	174.4	54.2	98.0	50.3	342.3	1.4	1.9	7.2	6.3	9.4
GRBCM	224.6	69.1	105.6	36.4	129.8	1.7	2.3	6.5	7.6	11.9
CPoE(1)	111.1	12.2	63.0	16.8	152.4	1.5	2.1	7.8	6.4	9.2
CPoE(2)	89.6	8.4	36.5	8.1	79.9	2.1	2.8	10.6	7.5	12.9
CPoE(3)	82.2	7.8	36.3	6.2	46.9	2.5	3.1	12.9	9.3	19.8
CPoE(4)	**79.5**	**7.6**	**36.0**	**4.7**	**32.8**	2.8	3.3	14.9	10.4	27.8

Table 3 Summary of used datasets and results for the *elecdemand* time series

	N	D	N_{test}	J
concrete	927	8	103	4
mg	1247	6	138	8
space	2797	6	310	8
abalone	3760	8	417	16
kin	5192	8	3000	16
kin2	7373	8	819	16
cadata	19640	8	1000	64
sarcos	43484	21	1000	128
casp	44730	9	1000	128
elecdemand	2184	3	15288	13

(a) Description of datasets.

	KL	KL IN	KL OUT	time
full GP	0.0	0.0	0.0	404.3
SGP(100)	120.9	110.5	146.7	56.3
SGP(200)	114.9	65.6	238.3	75.2
minVar	503.0	406.5	744.5	20.7
GPoE	328.0	336.0	307.9	20.4
GRBCM	393.4	382.1	421.8	28.2
CPoE(1)	289.5	255.1	375.5	20.5
CPoE(2)	113.1	108.5	124.3	36.8
CPoE(3)	86.4	61.9	147.6	39.7
CPoE(4)	**58.3**	**59.4**	**55.5**	52.9

(b) KL to full GP and time of different methods.

block where stochastic parameter optimization is exploited can be found in Table A4 in the Appendix.

In general, the local methods perform better than the global sparse method. Further, the performance of our correlated PoEs is superior to the one of independent PoEs for all datasets. In particular, the KL to full GP can be continuously improved for increasing degree of correlation, i.e. larger C values. The time for CPoE(1) is comparable with the independent PoEs and for increasing C, our approximation has a moderate increase in time with a significant decrease in KL. For more details about the experiments consider Sect. A.7.2 in the Appendix and more results including standard deviations are provided in Appendix F.

4.3 Comparison to non-GP methods

Third, we compare our probabilistic regression method CPoE to other popular non-GP regression methods, in particular, dense neural networks (MLPs), eXtreme Gradient Boosting (XGboost) and linear regression[1]. We use three different architectures for the neural networks, that is, MLP(100, 100), MLP(500, 500), *MLP*(100, 100, 100), where the numbers in the parentheses correspond to the number of hidden nodes

[1] We use the algorithms in https://scikit-learn.org.

Table 4 Average *RMSE* for our CPoE methods compared to non-GP regression methods

	Concrete	mg	Space	Abalone	Kin	Cadata	Sarcos	Casp
fullGP-SE	0.311	0.511	0.471	0.635	0.267			
fullGP-FLEX	0.254	0.509	0.455	0.638	0.28			
CPoE(1)-SE	0.333	**0.508**	0.506	0.637	0.31	0.476	0.099	0.597
CPoE(2)-SE	0.326	0.512	0.49	0.634	0.292	0.47	0.1	0.59
CPoE(3)-SE	0.323	0.513	0.489	**0.634**	**0.28**	0.47	0.099	0.59
CPoE(1)-FLEX	0.266	0.511	0.631	0.687	0.334	0.456	0.094	0.525
CPoE(2)-FLEX	0.259	0.515	0.446	0.669	0.315	0.423	0.094	0.522
CPoE(3)-FLEX	**0.255**	0.516	**0.444**	0.659	0.303	**0.42**	**0.092**	**0.522**
MLP(100-100)	0.289	0.525	0.482	0.652	0.287	0.456	0.117	0.591
MLP(500-500)	0.292	0.522	0.475	0.761	0.284	0.485	0.097	0.577
MLP(100-100-100)	0.285	0.531	0.476	0.762	0.299	0.485	0.106	0.585
XGboost	0.323	0.545	0.543	0.65	0.667	0.474	0.251	0.767
LinReg	0.626	0.633	0.645	0.66	0.765	0.605	0.27	0.854

The methods ending with SE were run with a squared-exponential and a flexible kernel (9), respectively. Best method (beside GP full) is indicated in bold

per hidden layer. Moreover, we used ADAM optimizer with learning rate 0.01. For XGboost(*max_depth*, *n_estimators*, *learning_rate*), we use XGboost(3, 100, 0.1). All these hyperparameters are chosen in primary experiments so that those methods obtain advantageous test performance. For our CPoE method, we use a SE kernel as in the previous sections, and in addition, we run the algorithm with a more flexible kernel, namely

$$k_\theta(\boldsymbol{x}_1, \boldsymbol{x}_2) = k_{SM_1}(\boldsymbol{x}_1, \boldsymbol{x}_2) + k_{SM_2}(\boldsymbol{x}_1, \boldsymbol{x}_2) + k_{MLP}(\boldsymbol{x}_1, \boldsymbol{x}_2) + k_{LIN}(\boldsymbol{x}_1, \boldsymbol{x}_2), \tag{9}$$

where k_{SM_i} is a spectral-mixture kernel (Wilson and Adams 2013), k_{MLP} an (infinite) wide 1-hidden-layer neural network kernel (Neal 1995) and k_{LIN} a linear kernel. We run full GP for smaller datasets as comparison. The average *RMSE*, *ABSE* and *time* results are provided in Tables 4, F22 and F23, respectively. For instance in Table 4, we can observed that the GP approximation methods using either a SE kernel or a more flexible kernel achieve competitive performance.

Finally, we would like to emphasize that our probabilistic CPoE model provides a predictive *distribution*, that is, it models the predictive uncertainty and can thus provide reliable credible-intervals. Computing also the predictive variances is a harder task than only computing the predictive means, as the most other regression algorithms do. Therefore, the slightly higher computational times (Table F23) for similar accuracy (Tables 4 and F22) are very reasonable in our opinion. More detailed results are given in Tables F15–F21 and on github.[2]

4.4 Time series application

In this section, our method is applied on time series data with covariates using a non-stationary kernel together with priors on the hyperparameters as discussed in Sect. A.3.3 by

[2] https://github.com/manschuer/CPoE/blob/main/experiments/comparison_non_GP.ipynb.

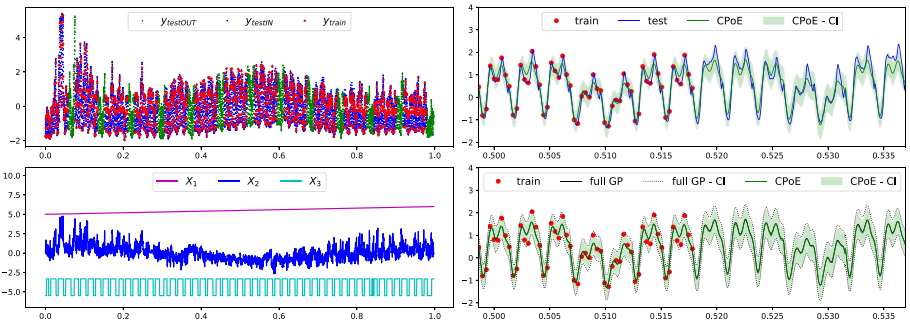

Fig. 10 Time series data with covariates and prior on hyperparameters

using MAP estimation. A recent work Corani et al. (2021) demonstrates that GPs constitute a competitive method for modelling time series using a sum of kernels including priors on the hyperparameters, which are previously learnt from a large set of different time series. We adapt their idea by using the same priors and a slightly modified kernel. In particular, for two data points $\boldsymbol{x}_1 = [t_1, x_{1,2}, \ldots, x_{1,D}]$ and $\boldsymbol{x}_2 = [t_2, x_{2,2}, \ldots, x_{2,D}]$, we model the kernel as the sum of 4 components

$$k_{\boldsymbol{\theta}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = k_{P_1}(t_1, t_2) + k_{P_2}(t_1, t_2) + k_{SM}(t_1, t_2) + k_{SE}(\boldsymbol{x}_1, \boldsymbol{x}_2),$$

where k_{P_1} and k_{P_1} are standard periodic kernels with period p_1 and p_2, respectively, k_{SM} a spectral-mixture kernel and k_{SE} a squared-exponential kernel. Note that, the former 3 kernels only depend on the first variable corresponding to time, whereas the SE-kernel depends on all variables, thus models the influence of the additional variables. With our CPoE model it is straightforward to handle time series with covariates, as opposed to other time series methods (Benavoli and Corani (2021), Corani et al. (2021), Sarkka et al. (2013), Hyndman and Athanasopoulos (2018)). We demonstrate the MAP estimation for $\boldsymbol{\theta}$ on the *elecdemand* time series (Hyndman (2020), Table 3), which contains the electricity demand as response y together with the time as the first variable x_1, the the corresponding temperature as x_2 and the variable whether it is a working day as x_3 which is depicted in the plots in Fig. 10 on the left, where we shifted the first and third variable in the second plot for the sake of clarity. Similarly as in the previous section, we run full GP, SGP, PoEs and CPoE and optimized the hyperparameter deterministically using the MAP as objective function taking into account the priors. The results are provided in Table 3 and in Fig. 10 on the right, which again show very competitive performance also for a general kernel with priors on the hyperparameters. More details about the experiment is given in Sect. A.7.3 in the Appendix.

5 Conclusion

In this paper, we introduce a novel GP approximation algorithm CPoE, where the degree of approximation can be adjusted by a locality and a sparsity parameter, so that the proposed method recovers independent PoEs, sparse global GP and full GP. Thereby, our method consistently approximates full GP, in particular, we proved that increasing the correlations between the experts decreases monotonically the KL of the joint prior of full GP to them

of our model. The presented algorithm has only a few hyperparameters, which allows an efficient deterministic and stochastic optimization. Further, our presented algorithm works with a general kernel, with several variables and also priors on the hyperparameters can be included. Moreover, the time and space complexity is linear in the number of experts and number of data samples, which makes it highly scalable. This is demonstrated with efficient implementations, so that a dataset with several ten thousands of samples can be processed in around a minute on a standard laptop. In several experiments with synthetic and real world data, superior performance in a accuracy vs. time sense compared to state-of-the-art methods, is demonstrated, which makes our algorithm a competitive GP regression approximation method.

Our approach could be enhanced in several directions. The first improvement would be more practical. While the current implementation of our algorithm works very competitively for moderate large datasets (on a standard laptop), further work has been done to scale it up to very large datasets. The current limitations are particularly factorizing the sparse block Cholesky matrices. We are convinced, that the theoretical properties of our algorithm—in particular the linearity in the number of experts and data samples—enables large scale implementations when exploiting more low level linear algebra tools. Another interesting direction would be to investigate the connection of our sparse precision matrix to state space systems, such that sequential learning algorithm could be exploited, which is briefly outlined in D. Further, it would be interesting to apply variational methods to our model, so that a connection to full GP in a posterior sense might be established, where some ideas are outlined in A.1.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10994-022-06297-3.

Author contributions MS conceived the presented idea, developed the theory, carried out the experiments and had the lead in writing the manuscript. DA and AB assisted by developing the idea and theory, designed the experiments, wrote part of the manuscript and reviewed drafts of the paper. MZ supervised the project, gave inputs for writing the manuscript and reviewed drafts of the paper.

Funding Open access funding provided by Università della Svizzera italiana. This work is supported by the Swiss National Research Programme 75 "Big Data" (NRP 75) with grant number 167199.

Data availability The data used in this paper is available on public data repositories as indicated in the supplementary material.

Code availability We provide the code of the proposed algorithms in the paper on https://github.com/manschuer/CPoE including descriptions how to use it and the performed experiments.

Declarations

Conflict of interest The authors do not have any conflict of interest.

Consent to participate Not applicable.

Consent for publication Not applicable.

Ethical approval This paper does not require special ethics approval.

References

Benavoli, A., & Corani, G. (2021). State space approximation of gaussian processes for time series forecasting. In *International workshop on advanced analytics and learning on temporal data*. Springer, Cham.

Bui, T. D., & Turner, R. E. (2014). Tree-structured gaussian process approximations. *Advances in Neural Information Processing Systems, 27,* 2213–2221.

Bui, T. D., Yan, J., & Turner, R. E. (2017). A unifying framework for sparse gaussian process approximation using power expectation propagation. *Journal of Machine Learning Research, 18,* 1–72.

Bui, T.D., Nguyen, C., Turner, R.E. (2017). Streaming sparse gaussian process approximations. In *Advances in Neural Information Processing Systems* (pp. 3301–3309).

Buschjäger, S., Liebig, T., Morik, K. (2019). Gaussian model trees for traffic imputation. In ICPRAM (pp. 243–254).

Chen, Y., Davis, T. A., Hager, W. W., & Rajamanickam, S. (2008). Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. *ACM Transactions on Mathematical Software (TOMS), 35*(3), 1–14.

Corani, G., Benavoli, A., Zaffalon, M. (2021). Time series forecasting with gaussian processes needs priors. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 103–117).

Csató, L., & Opper, M. (2002). Sparse online Gaussian processes. *Neural computation, 14*(3), 641–668.

Datta, A., Banerjee, S., Finley, A. O., & Gelfand, A. E. (2016). Hierarchical nearest-neighbor gaussian process models for large geostatistical datasets. *Journal of the American Statistical Association, 111*(514), 800–812.

Deisenroth, M.P., Ng, J.W. (2015). Distributed gaussian processes. arXiv preprint arXiv:1502.02843.

Durrande, N., Adam, V., Bordeaux, L., Eleftheriadis, S., Hensman, J. (2019). Banded matrix operators for gaussian markov models in the automatic differentiation era. In *The 22nd International Conference on Artificial Intelligence and Statistics* (pp. 2780–2789). PMLR

Fleet, Y.C.D.J. (2014). Generalized product of experts for automatic and principled fusion of gaussian process predictions. arXiv preprint arXiv:1410.7827.

GPy (2012). GPy: A Gaussian process framework in python. http://github.com/SheffieldML/GPy (since 2012)

Grigorievskiy, A., Lawrence, N., Särkkä, S. (2017). Parallelizable sparse inverse formulation gaussian processes (spingp). In *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)* (pp. 1–6). IEEE.

Hensman, J., Fusi, N., Lawrence, N.D. (2013). Gaussian processes for big data. In *Conference for Uncertainty in Artificial Intelligence*.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural computation, 14*(8), 1771–1800.

Hyndman, R. (2020). Fpp2: Data for "Forecasting: Principles and Practice" (2nd Edn.). R package version 2.4. https://CRAN.R-project.org/package=fpp2

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and practice*. OTexts.

Julier, S.J., Uhlmann, J.K. (1997). A non-divergent estimation algorithm in the presence of unknown correlations. In *Proceedings of the 1997 American Control Conference* (Cat. No. 97CH36041) (vol. 4, pp. 2369–2373). IEEE.

Kania, L., Schürch, M., Azzimonti, D., Benavoli, A. (2021). Sparse information filter for fast gaussian process regression. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*.

Katzfuss, M., & Guinness, J. (2021). A general framework for Vecchia approximations of Gaussian processes. *Statistical Science, 36*(1), 124–141.

Katzfuss, M., Guinness, J., Gong, W., & Zilber, D. (2020). Vecchia approximations of gaussian-process predictions. *Journal of Agricultural, Biological and Environmental Statistics, 25*(3), 383–414.

Kingma, D.P., Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Liu, H., Cai, J., Wang, Y., Ong, Y.S. (2018). Generalized robust Bayesian committee machine for large-scale gaussian process regression. In *International Conference on Machine Learning* (pp. 3131–3140). PMLR.

Liu, H., Ong, Y.-S., Shen, X., & Cai, J. (2020). When gaussian process meets big data: A review of scalable GPS. *IEEE Transactions on Neural Networks and Learning Systems, 31*(11), 4405–4423.

Maneewongvatana, S., Mount, D.M. (2001). On the efficiency of nearest neighbor searching with data clustered in lower dimensions. In *International Conference on Computational Science* (pp. 842–851).

Masoudnia, S., & Ebrahimpour, R. (2014). Mixture of experts: A literature survey. *Artificial Intelligence Review, 42*(2), 275–293.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT Press.

Nakai-Kasai, A., Tanaka, T. (2021). Nested aggregation of experts using inducing points for approximated gaussian process regression. *Machine Learning*, 1–24

Neal, R.M. (1995). Bayesian learning for neural networks. PhD Thesis, CAN. AAINN02676

Quiñonero-Candela, J., & Rasmussen, C. E. (2005). A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research, 6,* 1939–1959.

Rasmussen, C. E., & Williams, C. K. (2006). *Gaussian processes for machine learning* (Vol. 1). MIT press.

Rullière, D., Durrande, N., Bachoc, F., & Chevalier, C. (2018). Nested kriging predictions for datasets with a large number of observations. *Statistics and Computing, 28*(4), 849–867.

Sarkka, S., Solin, A., & Hartikainen, J. (2013). Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing: A look at Gaussian process regression through Kalman filtering. *IEEE Signal Processing Magazine, 30*(4), 51–61.

Schürch, M., Azzimonti, D., Benavoli, A., & Zaffalon, M. (2020). Recursive estimation for sparse gaussian process regression. *Automatica, 120,* 109127.

Seeger, M., Williams, C., Lawrence, N. (2003). Fast forward selection to speed up sparse gaussian process regression. In *Artificial intelligence and statistics* (Vol. 9).

Snelson, E., Ghahramani, Z. (2006). Sparse gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems* (pp. 1257–1264).

Takahashi, K. (1973). Formation of sparse bus impedance matrix and its application to short circuit study. In *Proceedings PICA Conference*, June, 1973.

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

Trapp, M., Peharz, R., Pernkopf, F., Rasmussen, C.E. (2020). Deep structured mixtures of gaussian processes. In *International Conference on Artificial Intelligence and Statistics* (pp. 2251–2261). PMLR.

Tresp, V. (2000). A Bayesian committee machine. *Neural computation, 12*(11), 2719–2741.

Wang, K., Pleiss, G., Gardner, J., Tyree, S., Weinberger, K. Q., & Wilson, A. G. (2019). Exact gaussian processes on a million data points. *Advances in Neural Information Processing Systems, 32,* 14648–14659.

Wilson, A., Adams, R. (2013) Gaussian process kernels for pattern discovery and extrapolation. In *International Conference on Machine Learning* (pp. 1067–1075). PMLR.