Robust and Scalable Gaussian Process Regression and Its Applications

Yifan Lu[1], **Jiayi Ma**[1*], **Leyuan Fang**[2], **Xin Tian**[1], **and Junjun Jiang**[3]
[1] Wuhan University, China [2] Hunan University, China [3] Harbin Institute of Technology, China
{lyf048, xin.tian}@whu.edu.cn, {jyma2010, fangleyuan}@gmail.com, jiangjunjun@hit.edu.cn

Abstract

This paper introduces a robust and scalable Gaussian process regression (GPR) model via variational learning. This enables the application of Gaussian processes to a wide range of real data, which are often large-scale and contaminated by outliers. Towards this end, we employ a mixture likelihood model where outliers are assumed to be sampled from a uniform distribution. We next derive a variational formulation that jointly infers the mode of data, i.e., inlier or outlier, as well as hyperparameters by maximizing a lower bound of the true log marginal likelihood. Compared to previous robust GPR, our formulation approximates the exact posterior distribution. The inducing variable approximation and stochastic variational inference are further introduced to our variational framework, extending our model to large-scale data. We apply our model to two challenging real-world applications, namely feature matching and dense gene expression imputation. Extensive experiments demonstrate the superiority of our model in terms of robustness and speed. Notably, when matching 4k feature points, its inference is completed in milliseconds with almost no false matches. The code is at github.com/YifanLu2000/Robust-Scalable-GPR.

1. Introduction

Gaussian processes (GPs) [31] are probably the primary non-parametric method for inference on latent functions. They have a wide range of applications from biology [3] to computer vision [41]. A commonly used observation model for Gaussian process regression (GPR) is the Normal distribution, which brings great convenience to the inference. Unfortunately, a well-known limitation of the Gaussian observation model is its sensitivity to outliers in data. As illustrated in Fig. 1 (b), a few outliers can drastically destroy the entire posterior regression result. This hinders the real-world applications of GPR for many domains, where outliers are often inevitable. This paper intends to conquer the GPR with outlier contaminated data.

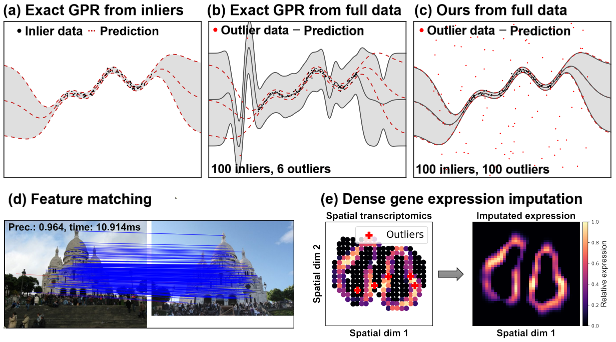

Figure 1. **Regression with our model.** (a) Perform exact GPR from 100 inliers. (b) When there are only 6 outliers in the data, the exact GPR leads to completely wrong results. (c) By comparison, our model is able to recover the exact posterior even facing 100 outliers. (d) The feature matching result using our model. (e) The dense spatial gene expression imputation result using our model.

The idea of robust regression is not new. Outlier detection has been extensively and systematically described in [6,9,10,29]. In the context of GPR, many efforts tried to replace the Gaussian likelihood with other distributions showing heavy-tail behaviors, including Student-t [16,21,28,30], Laplace [22, 30], Gaussian mixture [8, 22, 27], and data-dependent noise model [17]. The challenge with these non-Gaussian likelihoods lies in the inference, which is analytically intractable. To this end, many approximation schemes have been applied, despite having high computational complexity, *e.g.*, Markov Chain Monte Carlo (MCMC) sampling and Expectation Propagation (EP) [22].

In this paper, we propose a more effective mixture likelihood model, where uniform distribution accounts for the outliers and Gaussian for inliers. In our formulation, the outliers are independent of the GP and do not affect the computation of the posterior GP, thereby allowing to tolerate more outliers. We next introduce a variational method that jointly determines the modes of data (*i.e.*, inlier or outlier) as well as hyperparameters by maximizing a lower bound to the marginal likelihood. We highlight that the difference between our variational formulation and pervious methods is that the modes of data now become variational parameters and are obtained by minimizing the Kullback-

[*] Corresponding Author

Leibler (KL) divergence between the variational and the true posterior distribution. Thus, the proposed formulation is less likely to overfit and is able to approximate the exact posterior GP only from inliers, as in Fig. 1 (c).

Inspired by [37], the sparse inducing variable approximation is integrated into our variational framework, which retains the exact GP prior but performs posterior approximation, and reduces the time complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(nm^2)$. By treating the inducing variables as global variables [18], our variational model enjoys the acceleration by Stochastic Variational Inference (SVI) [20]. It performs stochastic optimization from natural gradient and further decreases the time complexity to $\mathcal{O}(km^2)$. This provides a guarantee for our model to scale to large-scale data.

We apply our robust GPR model to two real-world applications, say feature matching and dense spatial gene expression imputation, as illustrated in the Figs. 1 (d) and (e). Extensive experiments demonstrate the superiority of our method on both numerical data and real applications.

To summarize, our contributions include the following. (i) We present a robust Gaussian process regression model, which uses variational learning to approximate the true exact posterior. (ii) We leverage inducing variables and SVI to adapt our model to large-scale data. (iii) Two applications of our model are described. Extensive experimental validation demonstrates the superiority of our model.

2. Related Works

Robust GPR typically employs non-Gaussian likelihood with heavy tails. Representatively, [22] investigated the Student's-t noise model. They further developed variational inference and MCMC for GPR with Student's-t. In [39], Laplace approximation was used to approximate the log marginal likelihood of Student's-t. The Laplace noise model is also exploited in [22]. [30] introduced Expectation Maximization (EM) algorithm for Student's-t and Laplace. Recently, [8] used mixtures of two Gaussians assuming a lower variance for regular noise and a higher one for outliers, with EM to learn the hyperparameters. In this work, we also use a non-Gaussian likelihood, *i.e.*, mixture model. The difference is that here the outliers are considered to follow a uniform distribution, which is a more reasonable assumption since the sampling distribution of outliers should be independent of the true latent function. Besides, we derive a variational formulation in the GPR setting, which approximates the posterior distribution.

Scalable GPR improves the scalability of GPR while maintaining prediction quality for big data. The seminal work [40] adopted Nyström approximation to approximate the kernel matrix using m sparse points. Later, the idea was further promoted by [7, 15, 23, 33], which approximated the prior and performed exact inference. Unlike the prior approximations, the other line is the posterior approximations, which retain the exact prior but perform approximate inference. The most well-known work is the elegant variational free energy [37]. It directly approximates the posterior by using variational inference. With the advances in variational inference [20], SVI is introduced to GP [18], which uses natural gradients and results in remarkable computational efficiency. However, as the exact GPR, these scalable GPRs are sensitive to outliers. In contrast, our model can tolerate massive outliers while retaining highly scalability.

3. Gaussian Process Regression Revisited

Gaussian process is a collection of random variables, for which any finite subset has a joint Gaussian distribution [31]. It is completely specified by a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}')$ where the covariance function typically depends on a set of hyperparameters $\boldsymbol{\varphi}$. The GP is usually used as a prior over latent function $f(\mathbf{x}) \sim \mathcal{GP}(m, k)$. We can combine the GP prior with observed data to give a posterior over desired latent function.

Consider a regression problem, where the training data[1] is $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$. Each y_i is a noisy realization of the latent function at location $\mathbf{x}_i$ such that $y_i = f_i + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(\epsilon_i|0, \sigma^2)$ is typically assumed to be an *i.i.d.* Gaussian noise and $f_i = f(\mathbf{x}_i)$. Denoting all training inputs as $\mathbf{X}$, all training outputs $\mathbf{y}$, and all latent function values at $\mathbf{X}$ as $\mathbf{f}$. Placing a GP prior on the latent function f, we obtain

$$p(\mathbf{f}|\mathbf{X}, \boldsymbol{\varphi}) \sim \mathcal{N}(\mathbf{f}|m(\mathbf{X}), \mathbf{K}_{nn}), \tag{1}$$

where $\mathbf{K}_{nn}$ is the covariance function evaluated between all the training points. For simplicity of exposition, the mean function is set to zero $m(\mathbf{x}) = 0$. It does not reduce the generalizability of the GP as long as the problem is correctly transformed, *e.g.*, by subtracting the mean of $\mathbf{y}$. One can induce a posterior distribution on $\mathbf{f}$ according to Bayes' rule

$$p(\mathbf{f}|\mathcal{D}, \sigma^2, \boldsymbol{\varphi}) \sim \mathcal{N}(\mathbf{f}|\mathbf{K}_{nn}(\mathbf{K}_{nn}^{\epsilon})^{-1}\mathbf{y}, \boldsymbol{\Sigma}), \tag{2}$$

which is also a multivariate normal distribution, where $\mathbf{K}_{nn}^{\epsilon} = \mathbf{K}_{nn} + \sigma^2\mathbf{I}$ and $\boldsymbol{\Sigma} = (\mathbf{K}_{nn}^{-1} + \sigma^{-2}\mathbf{I})^{-1}$. The posterior distribution of $\mathbf{f}$ can help to compute the posterior predictive distribution of $\mathbf{f}_*$ at any test location $\mathbf{X}_*$:

$$\begin{aligned} p(\mathbf{f}_*|\mathcal{D}, \mathbf{X}_*) &= \int p(\mathbf{f}_*|\mathbf{f}, \mathbf{X}_*)p(\mathbf{f}|\mathcal{D})d\mathbf{f} \\ &= \mathcal{N}(\mathbf{f}_*|\mathbf{K}_{*n}(\mathbf{K}_{nn}^{\epsilon})^{-1}\mathbf{y}, \mathbf{K}_{**} - \mathbf{K}_{*n}(\mathbf{K}_{nn}^{\epsilon})^{-1}\mathbf{K}_{n*}), \end{aligned} \tag{3}$$

where, σ^2 and $\boldsymbol{\varphi}$ are omitted for brevity. The above posterior depends on the values of σ^2 and $\boldsymbol{\varphi}$, which can be inferred by maximizing the log marginal likelihood:

$$\log p(\mathbf{y}|\sigma^2, \boldsymbol{\varphi}) = \log \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{nn}^{\epsilon}), \tag{4}$$

which automatically achieves the bias-variance trade-off.

[1] Note that the derivation in this paper extends to multiple independent output dimensions is straightforward.

4. Robust Gaussian Process Regression Model

In real-world scenarios, a huge discrepancy between the model and data-generating process often occurs, leading to outliers. However, a critical flaw of Gaussian observation model is its non-robustness. This poses a great challenge to GPR in practice. To this end, this section introduces a robust GPR, which identifies the mode of data and simultaneously approximates the posterior of the desired model.

4.1. Problem Formulation

We explicitly consider the generation process of outliers. Since outliers are independent of the latent function, a reasonable assumption is that they obey a uniform distribution $\frac{1}{a}$, where the output space is a bounded region that covers the output $\mathbf{y}$ and a is a constant denoting the volume of the region. The inliers follow a regular Gaussian noise $\mathcal{N}(y_i|f_i, \sigma^2)$. We then associate each data with a latent variable $z_i \in \{0, 1\}$, where $z_i = 0$ indicates that the observation is generated by outlier distribution and $z_i = 1$ inlier. The joint distribution of (y_i, z_i) is then a mixture of two models:

$$p(y_i, z_i|f_i, \boldsymbol{\theta}) = \{(1-\gamma)\frac{1}{a}\}^{1-z_i}\{\gamma\mathcal{N}(y_i|f_i, \sigma^2)\}^{z_i}, \quad (5)$$

where $\boldsymbol{\theta} = (\sigma^2, \gamma, \boldsymbol{\varphi})$ denotes the latent variables set and $\gamma = p(z_i = 1)$ is the mixing coefficient.

Prior distributions describe beliefs about the model variables before the inference. As a robust GPR problem, we place a GP prior over the latent function, *i.e.*, Eq. (1). The variable γ controls the probability of occurrence of two models. We suppose it follows a Beta distribution

$$p(\gamma) \sim \text{Beta}(\gamma|B_a, B_b), \quad (6)$$

where B_a and B_b are hyperparameters. As we will see, $p(\gamma)$ is *conjugate* to the posterior and hence easier to handle. For other variables $(\sigma^2, \boldsymbol{\varphi})$, we introduce noninformative prior.

Full joint distribution is obtained by incorporating the above prior distributions as follows:

$$p(\mathcal{D}, \mathbf{f}, \boldsymbol{\theta}) \propto p(\mathbf{f}|\mathbf{X})p(\gamma)\prod_{i=1}^{n} p(y_i, z_i|f_i, \sigma^2, \gamma), \quad (7)$$

where $\boldsymbol{\theta} = (\sigma^2, \gamma, \mathbf{Z}, \boldsymbol{\varphi})$ and $\mathbf{Z} \in \{0, 1\}^n$ is the indicator variable vector. Our goal is to obtain the posterior $p(\mathbf{f}, \boldsymbol{\theta}|\mathcal{D})$, which can be induced from $p(\mathcal{D}, \mathbf{f}, \boldsymbol{\theta})$. Nevertheless, as there are two possibilities for each observation, *i.e.*, inlier and outlier, it will generate 2^n combinations to be evaluated. Thus, the exact estimation is analytically intractable and approximation is needed. Next, we use the variational inference to approximate the exact posterior. Once the variational distribution is obtained, the posterior predictive distribution can be determined.

4.2. Variational Inference

We wish to directly approximate the posterior $p(\mathbf{f}, \boldsymbol{\theta}|\mathcal{D})$ using a variational posterior $q(\mathbf{f}, \boldsymbol{\theta})$ such that the KL divergence between $p(\mathbf{f}, \boldsymbol{\theta}|\mathcal{D})$ and $q(\mathbf{f}, \boldsymbol{\theta})$ is minimized. The minimization is equivalently expressed as maximization of lower bound for the true log marginal likelihood:

$$\mathcal{L}(\mathbf{f}, \boldsymbol{\theta}) = \int q(\mathbf{f}, \boldsymbol{\theta}) \log\left(\frac{p(\mathbf{f}, \boldsymbol{\theta}, \mathcal{D})}{q(\mathbf{f}, \boldsymbol{\theta})}\right) d\mathbf{f} d\boldsymbol{\theta}. \quad (8)$$

The maximization is, however, intractable since the optimal solution is $\hat{q}(\mathbf{f}, \boldsymbol{\theta}) = p(\mathbf{f}, \boldsymbol{\theta}|\mathcal{D})$. To this end, we consider instead a restricted distribution of $q(\mathbf{f}, \boldsymbol{\theta})$.

Mean-field factorization supposes $q(\mathbf{f}, \boldsymbol{\theta})$ can be partitioned into disjoint groups. Considering the ease of calculation, we factorize $q(\mathbf{f}, \boldsymbol{\theta}) = q_1(\mathbf{f})q_2(\gamma)q_3(\mathbf{Z})q_4(\sigma^2, \boldsymbol{\varphi})$. Besides, we assume q_4 is Dirac delta function, *i.e.*, it has a point mass at $(\sigma^2, \boldsymbol{\varphi})$. With this factorization, the maximization of Eq. (8) is obtained by optimizing each of the factors. For q_i, the minimization becomes

$$\hat{q}_i = \arg\min_{q_i} \int q_i \log\left(\frac{\exp \mathbb{E}_{j\neq i}\left[\log p(\mathbf{f}, \boldsymbol{\theta}, \mathcal{D})\right]}{q_i}\right) d\boldsymbol{\theta}_i, \quad (9)$$

where $\mathbb{E}_{j\neq i}[\cdot]$ denotes the expectation over q_j for $j \neq i$. Note that Eq. (9) is the negative KL divergence between q_i and $\mathbb{E}_{j\neq i}\left[\log p(\mathbf{f}, \boldsymbol{\theta}, \mathcal{D})\right]$. Thus, the optimal q_i is given by

$$\log \hat{q}_i = \mathbb{E}_{j\neq i}\left[\log p(\mathbf{f}, \boldsymbol{\theta}, \mathcal{D})\right] + \text{const}. \quad (10)$$

4.3. Sparse Inducing Variable Approximation

The variational posterior approximation presented above can easily integrate sparse inducing variables [37], which allows the reduction of time complexity of GPR from $\mathcal{O}(n^3)$ to $\mathcal{O}(nm^2)$. Introducing m pseudo-inputs $\mathbf{X}_m$ with inducing variables $\mathbf{f}_m$, our goal is to use $\mathbf{f}_m$ to approximate $q(\mathbf{f})$, where $\mathbf{f}_m$ akin to $\mathbf{f}$ follows the same GP prior that $p(\mathbf{f}_m) \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{mm})$. We suppose $\mathbf{f}_m$ is a sufficient statistic for $\mathbf{f}$, *i.e.*, for any value u, $p(u|\mathbf{f}, \mathbf{f}_m) = p(u|\mathbf{f}_m)$ holds [37]. In this setting, the augmented variational posterior $q(\mathbf{f}, \mathbf{f}_m) = p(\mathbf{f}|\mathbf{f}_m)\phi(\mathbf{f}_m)$ and augmented latent variables set $\boldsymbol{\theta} = (\gamma, \sigma^2, \mathbf{Z}, \boldsymbol{\varphi}, \mathbf{X}_m)$ with factorization $q(\mathbf{f}, \mathbf{f}_m, \boldsymbol{\theta}) = q_1(\mathbf{f}, \mathbf{f}_m)q_2(\gamma)q_3(\mathbf{Z})q_4(\sigma^2, \boldsymbol{\varphi}, \mathbf{X}_m)$. The bound (8) becomes

$$\mathcal{L} = \int q(\mathbf{f}, \mathbf{f}_m, \boldsymbol{\theta}) \log\left(\frac{p(\mathbf{f}_m)p(\gamma)p(\mathbf{y}, \mathbf{Z}|\mathbf{f})}{\phi(\mathbf{f}_m)q_{2-4}}\right) d\mathbf{f} d\mathbf{f}_m d\boldsymbol{\theta}, \quad (11)$$

where the term $p(\mathbf{f}|\mathbf{f}_m)$ inside the log is eliminated. In what follows, we maximize the bound analytically by solving the optimal choice of the variational distribution $q(\mathbf{f}, \mathbf{f}_m, \boldsymbol{\theta})$.

4.4. Maximize Lower Bound

To optimize the bound (11) in mean-field factorization, a common approach is to use the coordinate ascent technique, *i.e.*, optimizing one factor q_i while keeping the others $q_{j\neq i}$ fixed, and cycling through these factors in turn. Convergence is guaranteed because the bound is convex and monotonically increasing with respect to each factor [5]. Next, we give closed-form expressions for factor updates without detailed derivations, which can be found in *suppl. material*.

Updating $q_1(\mathbf{f}, \mathbf{f}_m)$ means finding the optimal posterior GP given q_2, q_3, and q_4. We denote $p_i = \mathbb{E}[z_i]$ and $\mathbf{P} = \text{diag}(p_1, p_2, \cdots, p_n)$. Integrating over $\mathbf{f}$, (11) becomes

$$\mathcal{L} = \int \phi(\mathbf{f}_m) \log\left(\frac{p(\mathbf{f}_m)Q(\mathbf{f}_m, \mathbf{y})}{\phi(\mathbf{f}_m)}\right) d\mathbf{f}_m + \text{const.}, \quad (12)$$

where

$$\log Q(\mathbf{f}_m, \mathbf{y}) = \log \mathcal{N}(\mathbf{y}|\boldsymbol{\mu}_{\mathbf{f}|\mathbf{f}_m}, \sigma^2\mathbf{P}^{-1}) - \frac{1}{2\sigma^2}\text{tr}(\mathbf{P}\mathbf{K}_{\mathbf{f}|\mathbf{f}_m}), \quad (13)$$

$\boldsymbol{\mu}_{\mathbf{f}|\mathbf{f}_m} = \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{f}_m$, $\mathbf{K}_{\mathbf{f}|\mathbf{f}_m} = \mathbf{K}_{nn} - \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mn}$ are the expectation and covariance of $p(\mathbf{f}|\mathbf{f}_m)$, respectively. With some linear algebraic manipulation, the optimal $\hat{\phi}(\mathbf{f}_m)$ is given by a multivariate Gaussian distribution

$$\begin{aligned}\hat{\phi}(\mathbf{f}_m) &\propto p(\mathbf{f}_m)Q(\mathbf{f}_m, \mathbf{y}) \\ &= \mathcal{N}(\mathbf{f}_m|\boldsymbol{\mu}_m, \mathbf{A}_m),\end{aligned} \quad (14)$$

where $\boldsymbol{\mu}_m = \sigma^{-2}\mathbf{K}_{mm}\boldsymbol{\Sigma}\mathbf{K}_{mn}\mathbf{P}\mathbf{y}$, $\mathbf{A}_m = \mathbf{K}_{mm}\boldsymbol{\Sigma}\mathbf{K}_{mm}$, and $\boldsymbol{\Sigma} = (\mathbf{K}_{mm} + \sigma^{-2}\mathbf{K}_{mn}\mathbf{P}\mathbf{K}_{nm})^{-1}$. Note that we can always recover $q(\mathbf{f})$ from $q(\mathbf{f}, \mathbf{f}_m) = p(\mathbf{f}|\mathbf{f}_m)\phi(\mathbf{f}_m)$ by marginalizing out $\mathbf{f}_m$, which gives optimal $\hat{q}(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\boldsymbol{\mu}_\mathbf{f}, \mathbf{A})$, where $\boldsymbol{\mu}_\mathbf{f} = \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\boldsymbol{\mu}_m$ and $\mathbf{A} = \mathbf{K}_{\mathbf{f}|\mathbf{f}_m} + \mathbf{K}_{nm}\boldsymbol{\Sigma}\mathbf{K}_{mn}$.

Remark 1 *With the optimal* $\hat{\phi}(\mathbf{f}_m)$*, we obtain* $\langle\mathcal{N}_i\rangle = \exp(\mathbb{E}[\log\mathcal{N}_i])$ *as follows*

$$\langle\mathcal{N}_i\rangle = \mathcal{N}(y_i|\boldsymbol{\mu}_{\mathbf{f}i}, \sigma^2)\exp(-\frac{1}{2\sigma^2}\mathbf{A}_{ii}), \quad (15)$$

where $\mathcal{N}_i$ *denotes* $\mathcal{N}(y_i|f_i, \sigma^2)$*,* $\boldsymbol{\mu}_{\mathbf{f}i}$ *is the* i*-th element of* $\boldsymbol{\mu}_\mathbf{f}$*, and* $\mathbf{A}_{ii}$ *is the* i*-th diagonal element of* $\mathbf{A}$*.*

Updating $q_2(\gamma)$ indicates updating the inlier ratio or mixture. Suppose q_1, q_3, and q_4 are given and note $\hat{n} = \text{tr}(\mathbf{P})$. According to Eq. (10), we conclude that $\hat{q}_2(\gamma)$ again follows a Beta distribution

$$\hat{q}_2(\gamma) = \text{Beta}(\gamma|B_a + \hat{n}, B_b + n - \hat{n}). \quad (16)$$

Remark 2 *With the optimal* $\hat{q}_2(\gamma)$*, according to the properties of Beta distribution, we obtain* $\langle\gamma\rangle = \exp(\mathbb{E}[\log\gamma])$ *and* $\langle 1-\gamma\rangle = \exp(\mathbb{E}[\log(1-\gamma)])$ *as follows*

$$\begin{aligned}\langle\gamma\rangle &= \exp(\psi(B_a + \hat{n}) - \psi(B_a + B_b + n)), \\ \langle 1-\gamma\rangle &= \exp(\psi(B_b + n - \hat{n}) - \psi(B_a + B_b + n)),\end{aligned} \quad (17)$$

where $\psi(\cdot)$ *is the digamma function.*

Updating $q_3(\mathbf{Z})$ represents determining the probability of each data being an inlier given q_1, q_2, and q_4. It encodes the mode of each data point. From Eq. (10), it leads to a further factorization $\hat{q}_3(\mathbf{Z}) = \prod_{i=1}^n \hat{q}_3^{[i]}(z_i)$ and $\hat{q}_3^{[i]}(z_i)$ is shown to follow a Bernoulli distribution:

$$\hat{q}_3^{[i]}(z_i) = (1-p_i)^{1-z_i}p_i^{z_i}, \quad (18)$$

where

$$p_i = \frac{\langle\gamma\rangle\langle\mathcal{N}_i\rangle}{\langle 1-\gamma\rangle/a + \langle\gamma\rangle\langle\mathcal{N}_i\rangle}. \quad (19)$$

Remark 3 p_i *shows the posterior probability of the* i*-th data being an inlier. It depends on both the deviation between* y_i *and the posterior mean* $\boldsymbol{\mu}_{\mathbf{f}i}$*, and the term containing the posterior variance* $\mathbf{A}_{ii}$*. The importance of this variance term is that it encodes the uncertainty of the latent function at* $\mathbf{x}_i$*, which means if we are uncertain about that data, then we cannot identify it as an inlier, even though* y_i *and* $\boldsymbol{\mu}_{\mathbf{f}i}$ *are close. This is helpful to avoid overfitting.*

Updating $q_4(\sigma^2, \boldsymbol{\varphi}, \mathbf{X}_m)$ involves the optimization of hyperparameters. Specifically, σ^2 controls the noise level, $\boldsymbol{\varphi}$ represents the shape of the kernel function, and $\mathbf{X}_m$ is the location of the pseudo-inputs. As assumed previously, $q_4(\sigma^2, \boldsymbol{\varphi}, \mathbf{X}_m)$ obeys the Dirac delta distribution, which means we directly maximize the lower bound (11) with respect to $(\sigma^2, \boldsymbol{\varphi}, \mathbf{X}_m)$, rather than using Eq. (10).

Given q_1, q_2, and q_3, taking derivative of (11) over σ^2 and setting to zero, we obtain a closed-form expression

$$\hat{\sigma}^2 = \frac{1}{\hat{n}}(\mathbf{y} - \boldsymbol{\mu}_\mathbf{f})^\top\mathbf{P}(\mathbf{y} - \boldsymbol{\mu}_\mathbf{f}) + \frac{1}{\hat{n}}\text{tr}(\mathbf{P}\mathbf{A}), \quad (20)$$

where $\text{tr}(\cdot)$ is the trace. The lower bound (11) is, however, difficult to optimize for the remaining hyperparameters $(\boldsymbol{\varphi}, \mathbf{X}_m)$ since the integral is intractable. To this end, we use the reverse Jensen's inequality and obtain

$$\mathcal{L}_2 := \log\mathcal{N}(\mathbf{y}|0, \mathbf{K}_\mathbf{y}) - \frac{1}{2\sigma^2}\text{tr}(\mathbf{P}\mathbf{K}_{\mathbf{f}|\mathbf{f}_m}) + \text{const.} \leq \mathcal{L}, \quad (21)$$

where $\mathbf{K}_\mathbf{y} = \sigma^2\mathbf{P}^{-1} + \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mn}$. To set the optimized hyperparameters $(\boldsymbol{\varphi}, \mathbf{X}_m)$, we seek partial derivatives of $\mathcal{L}_2$ with respect to them. Thereafter, the standard gradient descent algorithm provides the value that maximizes $\mathcal{L}_2$.

4.5. Stochastic Variational Inference

Modern applications often involve handling massive data. However, the GPR equipped with variational inference does not easily scale to big data, even with inducing variables. This is because we need to go through the entire training data at each coordinate update. As the training data size grows, the computational overhead becomes larger.

An alternative to coordinate ascent is to use SVI [20], which is similar to stochastic gradient descent (SGD), that performs stochastic optimization from the noisy but cheap-to-compute estimates of the gradient. It iterates between subsampling a subset of data and adjusting the hidden structure solely based on the subset. Thus, SVI is much more efficient than conventional variational inference. To apply SVI to the probabilistic model, a set of global variables is needed [18]. In our robust GPR setting, the inducing variables $\mathbf{f}_m$ and mixing coefficient variable γ fill this role.

Natural Gradient. In gradient-based optimization, the natural gradient accounts for the geometric structure of probability parameters and is used in SVI to replace the Euclidean gradient. In exponential families, it is given by premultiplying the usual gradient by the inverse Fisher information matrix $G(\boldsymbol{\eta})^{-1}$, where $\boldsymbol{\eta}$ is the canonical parameter of the exponential families. Since both $\phi(\mathbf{f}_m)$ and $q(\gamma)$ are exponential family distributions, they enjoy simple natural gradients [19] of the bound (11) such that $g(\boldsymbol{\eta}) = \mathbb{E}[\boldsymbol{\eta}] - \boldsymbol{\eta}$, where $g(\cdot)$ denotes the natural gradient. For Gaussian distribution $\phi(\mathbf{f}_m)$, its canonical parameter is $[\mathbf{A}_m^{-1}\boldsymbol{\mu}_m, -\frac{1}{2}\mathbf{A}_m^{-1}]$. For Beta distribution $q(\gamma)$, its canonical parameter is $[B_a + \hat{n}, B_b + n - \hat{n}]$. Thereafter, we obtain the update rule using natural gradient with a step size ϵ_t

$$\boldsymbol{\eta}^t = \boldsymbol{\eta}^{t-1} + \varepsilon_t g(\boldsymbol{\eta}^{t-1}) = (1 - \varepsilon_t)\boldsymbol{\eta}^{t-1} + \varepsilon_t \mathbb{E}[\boldsymbol{\eta}]. \quad (22)$$

Note that when the step size $\varepsilon_t = 1$, we recover the original variational update in Eq. (14) and Eq. (16). Now, the noisy natural gradient can be easily computed by sampling either individual or mini-batch of the data.

4.6. Making Predictions

So far we have described how to infer the approximate posterior of the latent variables. To make predictions, we use the predictive distribution similar to Eq. (3):

$$\begin{aligned} p(\mathbf{f}_*|\mathcal{D}, \mathbf{X}_*) &= \int p(\mathbf{f}_*|\mathbf{f}_m)p(\mathbf{f}|\mathbf{f}_m)\phi(\mathbf{f}_m)d\mathbf{f}_m d\mathbf{f} \\ &= \mathcal{N}(\mathbf{f}_*|\mathbf{K}_{*m}\mathbf{K}_{mm}^{-1}\boldsymbol{\mu}_m, \mathbf{K}_{\mathbf{f}_*|\mathbf{f}_m} + \mathbf{K}_{*m}\boldsymbol{\Sigma}\mathbf{K}_{m*}), \end{aligned} \quad (23)$$

where $\mathbf{K}_{\mathbf{f}_*|\mathbf{f}_m} = \mathbf{K}_{**} - \mathbf{K}_{*m}\mathbf{K}_{mm}^{-1}\mathbf{K}_{m*}$.

5. Applications

In this section, we describe how to use the proposed GPR model to deal with real-world tasks, including feature matching and dense gene expression imputation.

5.1. Feature Matching

Feature matching aims at establishing reliable feature point correspondences. To solve the matching problem in a regression manner, we first construct a putative correspondence set $\mathcal{P} = \{(\mathbf{x}_i, \mathbf{x}'_i)\}_{i=1}^n$ by nearest neighbor (NN) matching, where $\mathbf{x}_i$ and $\mathbf{x}'_i$ represent the coordinates of feature points. The putative set is then converted to motion vector set, *i.e.*, $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, where $\mathbf{y}_i = \mathbf{x}'_i - \mathbf{x}_i$. The motion of an object projected on the image plane is known to be slow-and-smooth [38], which is well described by GP. Thus, the motion vectors of inliers can be seen as sampling on the underlying model $\mathbf{f}$, while the outliers are spuriously distributed. Our goal is to determine the correctness of each data. The task, however, is non-trivial as there are often massive outliers (even up to 90%). Moreover, the feature points in each image can reach several thousands, which requires high computational efficiency. As we will see, our model is robust and fast enough to handle such cases.

Inference Setting. To apply our model to feature matching, some special considerations need to be set. The feature coordinates are first normalized. For the covariance kernel, we choose the squared exponential (SE) kernel $k(\mathbf{x}, \mathbf{x}') = \lambda^{-1}\exp(-\beta\|\mathbf{x} - \mathbf{x}'\|^2)$. We find that fixed hyperparameters work well for most situations, which are empirically set to $\lambda = 4$ and $\beta = 0.1$. The positions of the inducing variables are randomly selected from the training data and kept fixed during inference. The ratio information is also considered as a prior. The mini-batch size is set to $\max\{\frac{n}{8}, 200\}$. After inference convergences, the i-th data is decided to be inlier if $p_i > \tau$, where τ is a threshold and is set to $\tau = 0.75$.

Predictions. One of the attractive properties of our model for this task is that when the training is completed, the resulting posterior GP is able to predict the motion vector of each feature point. Combined with the descriptors' information, we can yield much more correct correspondences rather than being limited to putative set. More correct correspondences are crucial for downstream tasks, as they can significantly improve the accuracy.

5.2. Dense Gene Expression Imputation

Recently, technological advances have made it possible to measure spatially resolved transcriptome-wide mRNA expressions with spatial information in tissue samples. GPR is thus utilized to infer gene expression in discrete samples, generating dense gene expression in 2D, 3D, and even spatiotemporal [3]. Unfortunately, due to equipment malfunctions and process disturbances, the readout data may contain bad spots. In this case, a vanilla GPR may lead to completely wrong results and thus mislead the subsequent analysis. By comparison, the robust GPR developed in this paper copes well with outliers and is able to give meaningful imputation results. For this application, the implementation is simple and requires little extra consideration.

6. Experiments

In this section, we first conduct numerical experiments to evaluate the performance in a fully controlled environment. After that, we apply our model to tackle feature matching problem and compare it with many state-of-the-art methods tailored for feature matching. In the end, the evaluation of gene expression imputation is performed. All experiments are conducted on a desktop with Intel i7-10700 CPU, 16 GB memory with MATLAB except deep learning methods.

6.1. Numerical Experiments

We study two datasets, *i.e.*, Neal dataset [28] and Friedman dataset [14]. The Neal dataset is one-dimensional data,

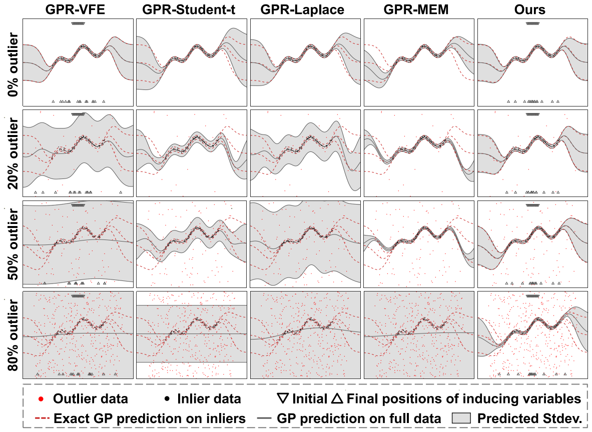

Figure 2. **Visual examples on Neal data.** Our model approximates the exact posterior GP using only inliers, even with massive outliers. The legend is shown in the bottom.

where the input x is drawn from a uniform distribution within $[-2.5, 2.5]$. The target value is calculated as:

$$f(x) = 0.3 + 0.4x + 0.5\sin(2.7x) + 1.1/(1 + x^2).$$

The Gaussian noise is added to the target value with zero mean and variance of 0.1. Next, outliers are injected into the training data, where the value is drawn from uniform distribution within $[-5, 5]$. The Friedman is 10-dimensional data and the function value depends on the first five dimensions

$$f(\mathbf{x}) = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5,$$

while the remaining dimensions complicate the task by adding a feature selection process. The input $\mathbf{x}$ is sampled uniformly within $[0, 1]^{10}$. Gaussian noise $\mathcal{N}(0, 0.01)$ is added and random outliers are distributed among $[5, 25]$.

We fix the inlier number to 100 and change the outlier ratio. GPR with Student's-*t* (St) [32], Laplace (Lap) [32], and Gaussian mixture distribution (MEM) [8], as well as variational free energy (VEF) [37], are chosen as competitors. Some intuitive examples on Neal data are shown in Fig. 2, where outlier ratio increases from 0 to 80%. We see that the prediction by our model almost exactly reproduces the full GP prediction using only inliers. In contrast, GPR-St and -Lap give poor approximations. GPR-MEM is able to give a better answer, yet still not as good as ours, especially at extremely high outlier ratios.

To statistically verify performance, we select mean absolute error (MAE, m), root mean square error (RMSE, r), and negative log of predictive probability (NLP, n) as metrics. 1000 testing points are generated in equal intervals. The statistical results on Neal and Friedman in Table 1 suggest that our method outperforms other alternatives in terms of robustness and accuracy. More results including KL and statistical variance are provided in *suppl. material*.

6.2. Feature Matching

We collect three large and diverse feature matching datasets with three different vision tasks, say YFCC100M dataset [36] (4000 pairs) for relative pose estimation, CPC dataset [42] (1000 pairs) for fundamental matrix estimation, and HPatches dataset [1] (580 pairs) for homography estimation. Six state-of-the-arts from handcrafted to deep learning are chosen for comparison, namely, MAGSAC++ [2], LPM [26], MCDM [25], CRC [12], PointCN [43], and OANet [44], with baseline ratio test. Briefly, MAGSAC++ is resampling-based method, LPM is a heuristic method that preserves local structure of feature points, CRC interpolates a smooth vector field using Fourier bases, PointCN uses deep neural networks with context normalization to encode global contextual information, OANet further captures local contextual information by clustering unordered correspondences. Note that we assess currently other robust GPR methods are not capable to handle the feature matching problem, thus none of them are chosen for comparison.

Table 1. **Quantitative results on Neal and Friedman data** with different outlier ratios. **Bold** indicates the best.

Neal [28]									
Method	10%			50%			80%		
	m	**r**	**n**	**m**	**r**	**n**	**m**	**r**	**n**
GPR-VFE [37]	0.18	0.21	0.73	0.58	0.77	1.75	0.76	1.08	1.96
GPR-St [32]	0.03	0.03	-0.41	0.09	0.19	0.39	0.77	1.15	1.66
GPR-Lap [32]	0.03	0.07	-1.17	0.42	0.56	1.62	0.35	0.72	2.03
GPR-MEM [8]	0.04	0.08	-1.36	0.06	0.12	-0.66	0.77	1.52	2.04
Ours	**0.01**	**0.02**	**-1.38**	**0.03**	**0.04**	**-1.25**	**0.03**	**0.03**	**-0.89**

Friedman [14]									
Method	20%			30%			50%		
	m	**r**	**n**	**m**	**r**	**n**	**m**	**r**	**n**
GPR-VFE [37]	1.12	1.24	2.28	1.47	1.62	3.08	2.88	3.16	6.93
GPR-St [32]	**0.04**	**0.04**	-0.08	0.91	1.10	3.59	1.47	2.18	7.16
GPR-Lap [32]	0.23	0.29	0.37	0.46	0.61	0.94	2.03	2.23	3.29
GPR-MEM [8]	0.11	0.21	-0.99	0.13	0.24	-0.66	2.13	2.32	2.98
Ours	**0.04**	0.06	**-1.46**	**0.06**	**0.08**	**-0.35**	**0.10**	**0.14**	**0.55**

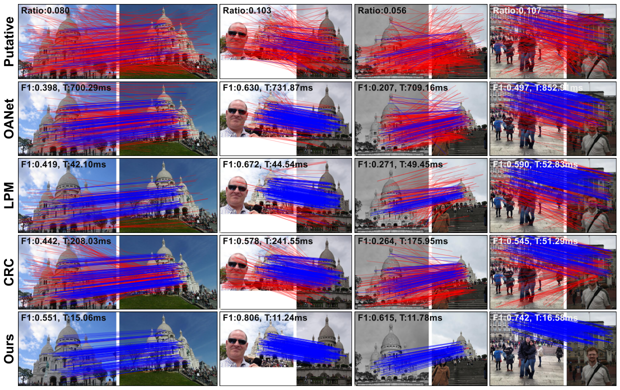

Figure 3. **Qualitative illustration of relative pose estimation.** We show the true positive in blue and false positive in red. Best viewed in color with 200% zoom in.

6.2.1 Relative Pose Estimation

Following [44], Yahoo's YFCC100M dataset [36] generates 4000 testing image pairs. SIFT [24] with NN matching is adopted to establish putative correspondences. The maximum number of keypoints for each image is limited to 4000. RANSAC [13] is used to estimate the relative camera pose.

Table 2. **Quantitative comparison for relative pose estimation**. **Bold** indicates the best and <u>underline</u> ranks the second.

Method	YFCC100M [36]					
	AUC			P	R	F
	@5°	@10°	@20°			
Ratio test	28.8	39.2	52.1	24.3	55.3	32.8
MSC++ [2]	24.9	33.7	45.7	24.3	55.3	32.8
PointCN [43]	45.1	54.6	65.0	47.2	<u>73.4</u>	54.9
OANet [44]	<u>53.6</u>	**64.1**	**75.8**	51.2	**87.0**	<u>61.7</u>
LPM [26]	38.6	48.8	59.7	51.4	43.6	45.9
MCDM [25]	47.8	59.4	70.7	<u>63.0</u>	46.5	52.7
CRC [12]	38.1	48.2	58.6	41.2	57.8	44.5
Ours	**53.8**	<u>63.7</u>	<u>73.8</u>	**85.3**	54.0	**63.7**

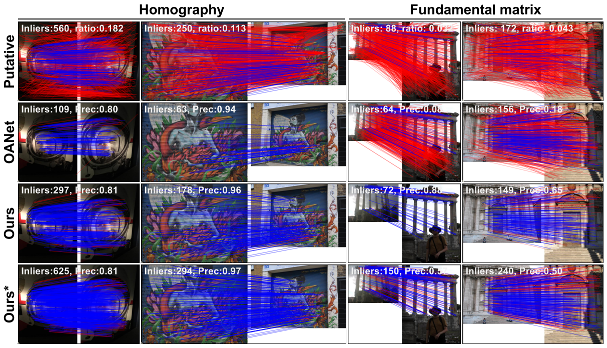

Figure 4. **Visual examples of homography and fundamental matrix estimation.** We show the true positive in blue and false positive in red. Best viewed in color with 200% zoom in.

We measure the maximum of rotation and translation errors in degree and report the approximate Area Under the Curve (AUC) with thresholds 5, 10, and 20 degrees. Precision, recall, and F1-score before RANSAC with respect to the ground-truth inliers are also presented. See *suppl. material* for a detailed description of the dataset and the metrics.

Some visualized examples are demonstrated in Fig. 3. Statistics results are comprehensively reported in Table 2. Since deep learning methods such as PointCN and OANet are trained in such dataset with geometric loss, they can capture geometric information and therefore have higher AUC compared to other handcrafted methods. By comparison, our method achieves comparable results to these deep learning methods with much higher precision and F1-score.

6.2.2 Homography & Fundamental Matrix Estimation

Homography and fundamental matrix estimation are critical parts in computer vision. The maximum keypoints for HPatches and CPC are limited to 4000. For evaluation metrics, we choose the precision, recall, and F1-score as well as the accuracy of the estimation. For homography estimation, we adopt *homography error* defined in [11], and for fundamental matrix, we follow [4] and use *normalized symmetric geometry distance* (NSGD) as the metrics. An estimate is classified as accurate or not by a threshold, which is set to 4 pixels for homography error and 0.05 for the NSGD [4].

Table 3. **Statistical results for homography and fundamental matrix estimation**. **Bold** ranks the first, <u>underline</u> the second.

Method	HPatches [1]				CPC [42]			
	Acc.	P	R	F	Acc.	P	R	F
Ratio test	71.7	71.6	51.6	63.0	17.0	12.7	63.9	20.5
MSC++ [2]	74.4	<u>87.3</u>	92.4	<u>86.0</u>	6.9	19.7	29.6	19.4
PointCN [43]	73.1	76.9	85.7	78.6	29.3	19.8	70.2	28.8
OANet [44]	70.0	73.5	77.2	72.1	47.0	34.6	<u>81.0</u>	<u>45.4</u>
LPM [26]	60.6	72.5	50.5	52.3	23.8	21.9	48.9	26.1
MCDM [25]	77.7	62.9	91.0	73.0	34.8	23.0	61.0	32.5
CRC [12]	71.7	75.5	92.5	80.4	28.7	24.8	47.2	30.4
Ours	<u>78.1</u>	**89.9**	<u>94.3</u>	**91.5**	<u>49.6</u>	**61.5**	59.3	**57.2**
Ours*	**82.4**	73.9	**109.6**	-	**51.9**	<u>39.2</u>	**97.3**	-

Table 4. **Ablation studies on feature matching. Bold** is the best.

Method		Prec.	Recall	F1	Time
CRC [12]		50.7	67.8	54.9	68.1
OANet (on GPU) [44]		47.9	**75.2**	56.0	41.2
RS-GPR	No SVI	76.3	66.1	70.0	47.1
	No inducing variables	72.6	69.0	69.5	1.6e5
	Opt. Hyperparameters	85.0	61.2	67.7	1556
	EM optimization	67.6	69.8	65.5	43.7
	Default	**88.0**	68.2	**75.6**	**9.8**

Several visual examples are provided in Fig. 4. Note that our predictive model is also presented and indicated by the superscript “*”. It establishes significantly more correct correspondences compared to other methods. Table 3 reports the statistical results. As it indicates, our method outperforms all other methods in the two tasks. When using the predictive model, it can even produce more correct correspondences than the putative set (*e.g.*, the recall in HPatches is higher than 100), thus further improving accuracy.

6.2.3 Ablation Studies of Feature Matching

To measure performance, especially runtime, in different settings, we conduct ablation studies on 100 image pairs from YFCC100M. It mainly includes four choices, *i.e.*, not using SVI, not using inducing variables, optimizing the kernel hyperparameters, and using EM optimization instead of variational inference. We also include the CRC and OANet to compare the runtime. The results are reported in Table 4, where we see that our method performs the best with the default setting described in Sect. 5.1. In particular, the running time is much faster compared to other settings and methods.

6.3. Dense Gene Expression Imputation

We collect two spatial transcriptomics datasets from mouse olfactory bulb and human breast cancer biopsies [34], respectively. Briefly, the spatial transcriptomics gene expression data is measured from thin tissue sections placed on an array with ploy probes and spatially resolved DNA barcodes. These create a grid of circular spots where the mRNA abundance of each spot is measured using probes with the barcodes encoding the spatial locations. The mouse olfactory bulb and human breast cancer biopsies datasets

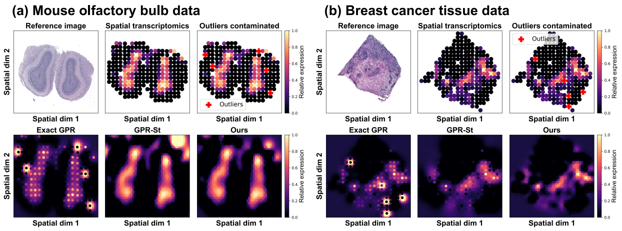

Figure 5. **Visualization of dense spatial gene expression imputation.** The spatial transcriptomics data is plotted on the top row. The bottom row shows the imputation results of different methods. We stack the spatial transcriptomics (shown as dots) on the imputated results to show the quality of the imputation.

Table 5. **Statistical results of dense gene expression imputation. Bold** indicates the best.

Method	**Pattern 1**			**Pattern 2**		
	m	**r**	**n**	**m**	**r**	**n**
Mouse olfactory bulb						
GPR-Exact	0.46	0.59	1.25	0.33	0.40	1.10
GPR-St [32]	0.07	0.14	-0.57	**0.02**	**0.04**	-1.33
GPR-Lap [32]	0.20	0.27	0.33	0.48	0.54	0.94
Ours	**0.03**	**0.05**	**-1.99**	**0.02**	**0.04**	**-2.21**
Breast cancer tissue						
GPR-Exact	0.24	0.35	1.44	0.23	0.33	1.15
GPR-St [32]	0.19	0.32	0.37	0.18	0.30	0.22
GPR-Lap [32]	0.25	0.36	0.56	0.25	0.37	0.53
Ours	**0.12**	**0.16**	**-0.54**	**0.11**	**0.17**	**-0.39**

contain 14859 and 12856 measured genes, respectively. To identify spatially variable genes, we adopt SpatialDE [35] and select two patterns for each dataset. We next add 5 random outliers to each identified pattern to simulate the outlier scenario. In the end, different GPR methods are applied to impute dense spatial gene expression.

Two visual examples are presented in Fig. 5, where in each example, the second row shows the imputed results. As can be seen, our method accurately infers the correct gene expression regardless of the outliers. To give quantitative results, we randomly split each dataset into 80% training data and 20% test data. As mentioned above, we use the MAE, RMSE, and NLP to measure the performance, and report the statistical results in Table 5. We see that our method always has the lowest error.

7. Analysis

Optimization Process. We visualize the intermediate results of the optimization process on Neal data with 50% outliers and present it in the top row of Fig. 6. We see that the field of p in Eq. (19) gradually converges to the inliers. After convergence, the posterior mean and variance functions are very close to the ground truth, and the inducing variables are restricted to the interval of the inliers.

The Role of Term $\mathbf{A}_{ii}$. We remove the term containing $\mathbf{A}_{ii}$ in Eq. (15) and show the results in the bottom of Fig. 6.

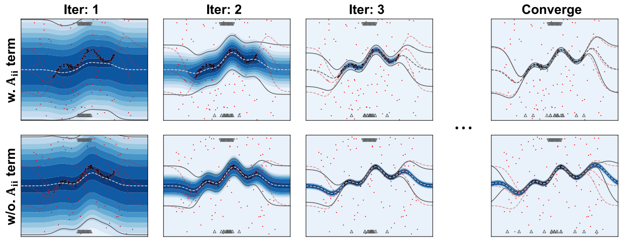

Figure 6. **Intermediate results of the optimization process.** The top row shows the full model while the bottom removes term $\mathbf{A}_{ii}$. We visualize the field of p, which indicates the inlier probability at any location. The deeper the blue color, the higher the probability.

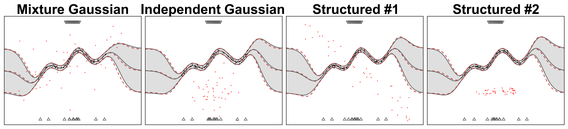

Figure 7. **Our model deal with different non-uniform outliers.**

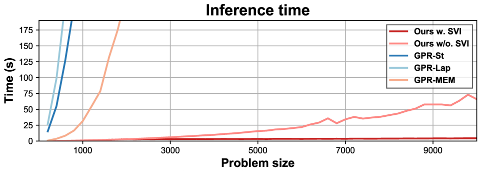

Figure 8. **The inference time of each GPR method.** Our model with SVI can easily handle big data.

Without variance term, the regression tends to overfit.

Non-uniform Outliers. Different types of non-uniform outliers are tested in Fig. 7. More detailed results are given in *suppl. material*. Our model exhibits strong generalization and robustness. The reason is that uniform distribution is a very weak assumption with high flexibility. It accommodates many outlier distributions.

Scalability. We test the runtime towards different problem sizes on Neal data with 50% outliers. Other methods as well as ours w. and w/o. SVI are included. The number of inducing variables is set to 15 for our method in both two settings. As plotted in Fig. 8, our method can easily scale to large-scale data, especially when equipped with SVI.

8. Conclusion

This paper presents a robust and scalable GPR using variational learning. It uses an outlier robust mixture likelihood model, where the uniform distribution accounts for the outliers. A variational formulation is introduced to learn the mode of data and hyperparameters by minimizing the KL divergence between the true posterior distribution and the approximate one. Its most attractive property is that it can rigorously approximate the exact posterior. Inducing variable approximation and SVI further extend our model to big data. We apply our model to feature matching and dense gene expression imputation. Extensive results show the significant improvement over existing robust GPRs.

References

[1] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5173–5182, 2017. 6, 7

[2] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. Magsac++, a fast, reliable and accurate robust estimator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1304–1312, 2020. 6, 7

[3] Sophie Bergmann, Christopher A Penfold, Erin Slatery, Dylan Siriwardena, Charis Drummer, Stephen Clark, Stanley E Strawbridge, Keiko Kishimoto, Alice Vickers, Mukul Tewary, et al. Spatial profiling of early primate gastrulation in utero. *Nature*, 609(7925):136–143, 2022. 1, 5

[4] Jia-Wang Bian, Yu-Huan Wu, Ji Zhao, Yun Liu, Le Zhang, Ming-Ming Cheng, and Ian Reid. An evaluation of feature matchers for fundamental matrix estimation. In *Proceedings of the British Machine Vision Conference*, 2019. 7

[5] Christopher M Bishop and Nasser M Nasrabadi. *Pattern Recognition and Machine Learning*. Springer, 2006. 4

[6] Michael J Black and Anand Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision*, 19(1):57–91, 1996. 1

[7] Lehel Csató and Manfred Opper. Sparse on-line gaussian processes. *Neural Computation*, 14(3):641–668, 2002. 2

[8] Atefeh Daemi, Hariprasad Kodamana, and Biao Huang. Gaussian process modelling with gaussian mixture likelihood. *Journal of Process Control*, 81:209–220, 2019. 1, 2, 6

[9] A Philip Dawid. Posterior expectations for large observations. *Biometrika*, 60(3):664–667, 1973. 1

[10] Bruno De Finetti. The bayesian approach to the rejection of outliers. In *Proceedings of the fourth Berkeley Symposium on Probability and Statistics*, pages 199–210, 1961. 1

[11] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236, 2018. 7

[12] Aoxiang Fan, Xingyu Jiang, Yong Ma, Xiaoguang Mei, and Jiayi Ma. Smoothness-driven consensus based on compact representation for robust feature matching. *IEEE Transactions on Neural Networks and Learning Systems*, 2021. 6, 7

[13] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 6

[14] Jerome H Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67, 1991. 5, 6

[15] Yarin Gal and Richard Turner. Improving the gaussian process sparse spectrum approximation by representing uncertainty in frequency inputs. In *Proceedings of the International Conference on Machine Learning*, pages 655–664, 2015. 2

[16] John Geweke. Bayesian treatment of the independent student-t linear model. *Journal of Applied Econometrics*, 8(S1):S19–S40, 1993. 1

[17] Paul Goldberg, Christopher Williams, and Christopher Bishop. Regression with input-dependent noise: A gaussian process treatment. *Advances in Neural Information Processing Systems*, 10, 1997. 1

[18] James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013. 2, 5

[19] James Hensman, Magnus Rattray, and Neil Lawrence. Fast variational inference in the conjugate exponential family. *Advances in Neural Information Processing Systems*, 25, 2012. 5

[20] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013. 2, 4

[21] Pasi Jylänki, Jarno Vanhatalo, and Aki Vehtari. Robust gaussian process regression with a student-t likelihood. *Journal of Machine Learning Research*, 12(11):3227–3257, 2011. 1

[22] Malte Kuss. *Gaussian process models for robust regression, classification, and reinforcement learning*. PhD thesis, echnische Universität Darmstadt Darmstadt, Germany, 2006. 1, 2

[23] Miguel Lázaro-Gredilla, Joaquin Quinonero-Candela, Carl Edward Rasmussen, and Aníbal R Figueiras-Vidal. Sparse spectrum gaussian process regression. *The Journal of Machine Learning Research*, 11:1865–1881, 2010. 2

[24] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 6

[25] Jiayi Ma, Aoxiang Fan, Xingyu Jiang, and Guobao Xiao. Feature matching via motion-consistency driven probabilistic graphical model. *International Journal of Computer Vision*, 130(9):2249–2264, 2022. 6, 7

[26] Jiayi Ma, Ji Zhao, Junjun Jiang, Huabing Zhou, and Xiaojie Guo. Locality preserving matching. *International Journal of Computer Vision*, 127(5):512–531, 2019. 6, 7

[27] Andrew Naish-Guzman and Sean Holden. Robust regression with twinned gaussian processes. *Advances in Neural Information Processing Systems*, 20, 2007. 1

[28] Radford M Neal. Monte carlo implementation of gaussian process models for bayesian regression and classification. *arXiv preprint physics/9701026*, 1997. 1, 5, 6

[29] Anthony O'Hagan. On outlier rejection phenomena in bayes inference. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(3):358–367, 1979. 1

[30] Rishik Ranjan, Biao Huang, and Alireza Fatehi. Robust gaussian process modeling using em algorithm. *Journal of Process Control*, 42:125–136, 2016. 1, 2

[31] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003. 1, 2

[32] Carl Edward Rasmussen and Hannes Nickisch. Gaussian processes for machine learning (gpml) toolbox. *The Jour-*

nal of Machine Learning Research, 11:3011–3015, 2010. 6, 8

[33] Carl Edward Rasmussen and Joaquin Quinonero-Candela. Healing the relevance vector machine through augmentation. In *Proceedings of the International Conference on Machine learning*, pages 689–696, 2005. 2

[34] Patrik L Ståhl, Fredrik Salmén, Sanja Vickovic, Anna Lundmark, José Fernández Navarro, Jens Magnusson, Stefania Giacomello, Michaela Asp, Jakub O Westholm, Mikael Huss, et al. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science*, 353(6294):78–82, 2016. 7

[35] Valentine Svensson, Sarah A Teichmann, and Oliver Stegle. Spatialde: identification of spatially variable genes. *Nature Methods*, 15(5):343–346, 2018. 8

[36] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 6, 7

[37] Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Proceedings of the Artificial Intelligence and Statistics*, pages 567–574, 2009. 2, 3, 6

[38] Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 203(1153):405–426, 1979. 5

[39] Jarno Vanhatalo, Pasi Jylänki, and Aki Vehtari. Gaussian process regression with student-t likelihood. *Advances in Neural Information Processing Systems*, 22, 2009. 2

[40] Christopher Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. *Advances in Neural Information Processing Systems*, 13, 2000. 2

[41] Oliver Williams and Andrew Fitzgibbon. Gaussian process implicit surfaces. In *Gaussian Processes in Practice*, 2006. 1

[42] Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In *Proceedings of the European Conference on Computer Vision*, pages 61–75, 2014. 6, 7

[43] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2666–2674, 2018. 6, 7

[44] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5845–5854, 2019. 6, 7