

# Scalable Variational Gaussian Processes for Crowdsourcing: Glitch Detection in LIGO

Pablo Morales-Álvarez, Pablo Ruiz, Scott Coughlin,  
Rafael Molina, *Senior Member, IEEE*, Aggelos K. Katsaggelos, *Fellow, IEEE*

**Abstract**—In the last years, crowdsourcing is transforming the way classification training sets are obtained. Instead of relying on a single expert annotator, crowdsourcing shares the labelling effort among a large number of collaborators. For instance, this is being applied in the laureate Laser Interferometer Gravitational Waves Observatory (LIGO), in order to detect glitches which might hinder the identification of true gravitational-waves. The crowdsourcing scenario poses new challenging difficulties, as it has to deal with different opinions from a heterogeneous group of annotators with unknown degrees of expertise. Probabilistic methods, such as Gaussian Processes (GP), have proven successful in modeling this setting. However, GPs do not scale up well to large data sets, which hampers their broad adoption in real-world problems (in particular LIGO). This has led to the very recent introduction of deep learning based crowdsourcing methods, which have become the state-of-the-art for this type of problems. However, the accurate uncertainty quantification provided by GPs has been partially sacrificed. This is an important aspect for astrophysicists in LIGO, since a glitch detection system should provide very accurate probability distributions of its predictions. In this work, we first leverage a standard sparse GP approximation (SVGP) to develop a GP-based crowdsourcing method that factorizes into mini-batches. This makes it able to cope with previously-prohibitive data sets. This first approach, which we refer to as Scalable Variational Gaussian Processes for Crowdsourcing (SVGPCR), brings back GP-based methods to a state-of-the-art level, and excels at uncertainty quantification. SVGPCR is shown to outperform deep learning based methods and previous probabilistic ones when applied to the LIGO data. Its behavior and main properties are carefully analyzed in a controlled experiment based on the MNIST data set. Moreover, recent GP inference techniques are also adapted to crowdsourcing and evaluated experimentally.

**Index Terms**—Crowdsourcing, Citizen Science, Laser Interferometer Gravitational Waves Observatory, Sparse Gaussian Processes, Scalability, Uncertainty Quantification, Deep Learning.

## 1 INTRODUCTION

Crowdsourcing, also known as citizen science, is revolutionizing the way real-world data sets are obtained nowadays [1], [2]. Traditionally, the task of labelling has been accomplished by a single expert annotator in a process that is time-consuming, expensive and difficult to scale. The proliferation of web services such as Amazon Mechanical Turk ([www.mturk.com](http://www.mturk.com)) and Figure-Eight ([www.figure-eight.com](http://www.figure-eight.com), formerly Crowdfunder) allows for outsourcing this process to a distributed workforce that can collaborate virtually, sharing the effort among a huge number of annotators [3], [4]. This approach is rapidly growing in popularity, and is being applied to many different fields

- PMA and RM are with the Department of Computer Science and Artificial Intelligence at the University of Granada (Spain).
- PR and AKK are with the Department of Electrical Engineering and Computer Science at Northwestern University (USA).
- SC is with the Center for Interdisciplinary Exploration and Research in Astrophysics (CIERA) at Northwestern University, and with the Department of Physics and Astronomy at Cardiff University (UK).

This work was supported by the Spanish Ministry of Economy and Competitiveness under project DPI2016-77869-C2-2-R, the Spanish Ministry of Science and Innovation under project PID2019-105142RB-C22, the US National Science Foundation through the NSF INSPIRE 15-47880 grant (Gravity Spy project), the US Department of Energy under grant DE-NA0002520, and the University of Granada through the Visiting Scholar Program. PMA is supported by La Caixa Banking Foundation (ID 100010434, Barcelona, Spain) through La Caixa Fellowship for Doctoral Studies LCF/BQ/ES17/11600011. We would also like to thank Mike Zevin and Neda Rohani for useful discussions.

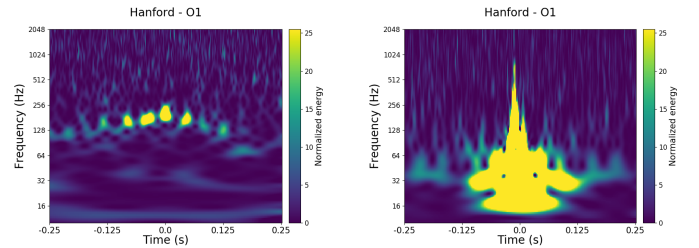


Fig. 1. Two examples of glitches observed by the LIGO detector. The fifteen types considered in this work will be carefully described in section 3, see also figure 3.

such as medical imaging [5], genetics [6], remote sensing [7], topic modelling [8], and object segmentation [9].

A very recent application of crowdsourcing in the field of astrophysics is the GravitySpy project [10], which aims at detecting glitches in the Laser Interferometer Gravitational Waves Observatory (LIGO). The LIGO collaboration is one of the most exciting and recognized scientific international initiatives [11]. It was awarded the 2017 Physics Nobel Prize for the first empirical detection of a gravitational-wave in September 2015 [12]. These waves are ripples in the fabric of spacetime, their existence was theoretically predicted by Einstein's General Relativity theory in 1916, and open a whole new way to explore the universe (beyond the electromagnetic signals available so far) [13]. However, the LIGO detector is equipped with extremely delicate technology,

which is sensitive to many different sources of noise. This produces a wide variety of glitches, see figure 1, which make the detection of true gravitational-waves difficult. The goal of GravitySpy is to leverage citizen science to label the large data set of glitches produced by LIGO, and then develop a machine learning system (based on crowdsourcing methods) to help astrophysicists classify them [10].

The crowdsourcing scenario introduces new challenges in machine learning, such as combining the unknown expertise of annotators, dealing with disagreements on the labelled samples, or detecting the existence of spammer and adversarial annotators [14], [15]. The first approaches to deal with multiple-annotated data used to rely on some kind of label aggregation mechanism prior to training. The most straightforward one is majority voting, which assumes that every annotator is equally reliable. More elaborated methods consider the biases of the different annotators, yielding a better calibrated set of training labels, see [16] (which is usually considered the first crowdsourcing work) and [17], [18]. In all these cases, the idea is to obtain a set of clean true labels, which are then fed to the preferred standard (no-crowdsourcing) classification algorithm.

However, recent works show that jointly modelling the classifier and the annotators behavior leads to superior performance, since the features provide information to puzzle out the true labels [19], [20]. In this joint model, Bayesian methods based on Gaussian Processes (GPs) have proved extremely successful to accurately quantify uncertainty [8], [21], [22]. However, in real-world applications they have been gradually replaced by deep learning based approaches [5], [23], [24], since GPs do not scale well to large data sets [8], [25]. As a result, the sound probabilistic formulation of GPs has been sacrificed. However, large scale problems could greatly benefit from such a solid modelling. In particular, in order to develop a reliable glitch detection system, astrophysicists with the GravitySpy project are particularly interested in the Bayesian formulation given by GPs [10]. Therefore, their scalability issues must be addressed.

GP is a popular Bayesian non-parametric model for supervised learning that excels at uncertainty quantification [25]. Due to the kernel matrix inversion, its computational cost at training is  $\mathcal{O}(N^3)$ , where  $N$  is the size of the training set. To overcome this problem, different sparse GP approximations have been proposed in the last years [26]. Many of them rely on the notion of inducing points, a reduced set of  $M$  ( $M \ll N$ ) instances which condense the information contained in the whole data set [27], [28]<sup>1</sup>. One of the most widespread methods is the Scalable Variational Gaussian Processes (SVGP) method [31], which uses Variational Inference (VI) [32] and is inspired by the earlier regression method [33]. SVGP can be trained through mini-batches and yields a training computational cost of  $\mathcal{O}(N_b M^2)$ , with  $N_b$  the mini-batch size. This allows SVGP to handle data sets of practically any size [31]. In fact, very interesting rates of convergence have been obtained recently for sparse GP regression problems [34]. They provide theoretical guarantees of scalability, showing that the increase of  $M$  can be kept

slower than  $N$ , specially for large datasets. Although these proofs do not directly apply for classification or crowdsourcing problems, in practice we will also observe that a small value of  $M$  can deal with the LIGO problem successfully.

In this work, we start by extending the well-established sparse GP approximation behind SVGP to the multiple-annotated crowdsourcing setting. Importantly, the form of the Evidence Lower Bound (ELBO) is still suitable for Stochastic VI [35], which allows for training through mini-batches. To the best of our knowledge, this allows GPs to be used for crowdsourcing problems of virtually any size for the first time. This method is referred to as Scalable (or Sparse) Variational Gaussian Processes for Crowdsourcing (SVGPCR). The annotators noise model is also fully Bayesian, described by per-user confusion matrices which are assigned Dirichlet priors. The underlying true labels are modelled in a probabilistic manner as well. VI is used to approximate the posterior distribution of the model.

In order to deal with the LIGO data, SVGPCR is modelled and implemented as a multi-class method. The implementation is based on GPflow, a very popular GP library that benefits from GPU acceleration through TensorFlow [36]. Three sets of experiments are provided. First, a controlled crowdsourcing problem specified for MNIST illustrates the main properties and behavior of SVGPCR. Among these, we may highlight its accurate identification of annotators' expertise degree, reconstruction of the real underlying label, and how the number of inducing points influences its performance. Secondly, SVGPCR is compared against previous probabilistic crowdsourcing methods in a relevant binary LIGO problem<sup>2</sup>. SVGPCR stands out as the best performing approach, thanks to its innovative scalability through mini-batches. Third, SVGPCR is shown to outperform state-of-the-art DL-based methods in the full LIGO data set, specially in terms of test likelihood, due to the more robust uncertainty control.

Once SVGPCR has been successfully developed, we explore more recent GP inference techniques in the LIGO problem (beyond the standard SVGP). For instance, GPs can be combined with inference networks [37] and the amortized setting of Variational Autoencoders [38]. More expressive posteriors can be described by Normalizing Flows [39], [40], and the use of sampling is an alternative to VI [41], [42]. While the number of possibilities is large and worth exploring, here we restrict ourselves to two approaches. First, we resort to Normalizing Flows to allow for richer posterior within the SVGPCR model. The results are only slightly better than SVGPCR, which suggests that a unimodal Gaussian posterior might be enough in this application. Then, we further modify the model by extending to crowdsourcing the use of inference networks for GPs [37]. In this case, the results are clearly better for low values of  $M$  (measurement points), which is precisely one of the main benefits of [37].

The rest of the paper is organized as follows. Section 2 describes the proposed model and inference procedure. Section 3 presents the LIGO data available in the GravitySpy project. The experimental results for SVGPCR are discussed in Section 4. More recent inference approaches are explored

1. There exist other sparse GP approximations which alternatively rely on Fourier features [29], and which have been already used for crowdsourcing problems [30]. In the experiments, the proposed method will be shown to clearly outperform these alternative approaches too.

2. Most of these previous probabilistic crowdsourcing approaches were originally proposed for binary problems, and the code is available accordingly.

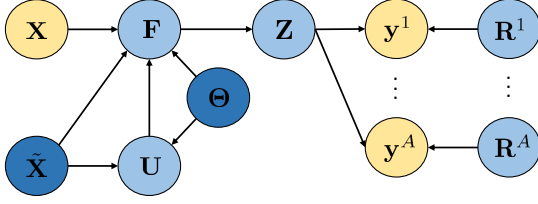


Fig. 2. Probabilistic graphical model for SVGPCR. Observed variables are depicted in yellow, and those to be estimated in blue. In the latter case, the intensity indicates whether the estimation is through a posterior distribution (light blue) or a point value (dark).

in Section 5. Finally, Section 6 contains some remarks and future outlook.

## 2 PROBABILISTIC MODEL AND INFERENCE

This section introduces the theoretical formulation of the proposed method. Figure 2 shows a graphical representation of the proposed model, which will be useful here.

### 2.1 The model

In a crowdsourcing problem with  $K$  classes, we observe training data  $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{Y}_n^a) : n = 1, \dots, N; a \in A_n\}$ , where  $\mathbf{x}_n \in \mathbb{R}^D$  are the training features, and  $\mathbf{Y}_n^a$  is the set<sup>3</sup> of annotations provided by the  $a$ -th annotator for the  $n$ -th instance. That is, each  $\mathbf{y} \in \mathbf{Y}_n^a$  is an one-hot encoded vector in  $\{\mathbf{e}_1, \dots, \mathbf{e}_K\}$  that represents the  $k$ -th class (i.e., all elements of  $\mathbf{e}_k$  are zero but the  $k$ -th one, which is one). There are  $N$  data points,  $A$  annotators, and  $A_n \subseteq \{1, \dots, A\}$  contains the annotators that labelled the  $n$ -th instance. All training instances will be grouped in  $\mathbf{X} = \{\mathbf{x}_n : n = 1, \dots, N\}$ , and analogously all annotations in  $\mathbf{Y} = \{\mathbf{Y}_n^a : n = 1, \dots, N; a \in A_n\}$ .

As with previous approaches [8], [19], [20], [21], [22], the proposed model assumes an (unknown) real label for each instance,  $\mathbf{z}_n \in \{\mathbf{e}_1, \dots, \mathbf{e}_K\}$ . The actual annotations depend on this real label and the degree of expertise of each annotator, which is modelled by the confusion matrix  $\mathbf{R}^a = (r_{ij}^a)_{1 \leq i, j \leq K}$ . Each  $r_{ij}^a \in [0, 1]$  represents the probability that the  $a$ -th annotator labels as class  $i$  an instance whose real class is  $j$ . Notice that this matrix must add up to one by columns. Mathematically, this is given by

$$p(\mathbf{Y}_n^a | \mathbf{z}_n, \mathbf{R}^a) = \prod_{\mathbf{y} \in \mathbf{Y}_n^a} \mathbf{y}^\top \mathbf{R}^a \mathbf{z}_n. \quad (1)$$

If  $\mathbf{y} = \mathbf{e}_i$  and  $\mathbf{z}_n = \mathbf{e}_j$ , the product  $\mathbf{y}^\top \mathbf{R}^a \mathbf{z}_n$  yields  $r_{ij}^a$ . Other variants of crowdsourcing likelihoods could be explored. For instance, the values in  $\mathbf{R}^a$  could depend on the input  $\mathbf{x}$  (i.e. the annotator degree of expertise might vary depending on the specific features of the input).

Assuming that all annotators label the different instances independently, we have

$$p(\mathbf{Y} | \mathbf{Z}, \mathbf{R}) = \prod_{n=1}^N \prod_{a \in A_n} p(\mathbf{Y}_n^a | \mathbf{z}_n, \mathbf{R}^a), \quad (2)$$

3. Notice that annotators are allowed to label the same instance more than once (possibly with different labels). This happens in a few cases in the LIGO data.

where  $\mathbf{Z} = \{\mathbf{z}_n : n = 1, \dots, N\}$  and  $\mathbf{R} = \{\mathbf{R}^a : a = 1, \dots, A\}$  group the corresponding individual variables, and  $p(\mathbf{Y}_n^a | \mathbf{z}_n, \mathbf{R}^a)$  is given by eq. (1).

The prior distribution for the annotators behavior is modelled through (independent) Dirichlet distributions, which are conjugate to the categorical one in eq. (1) [43]. This yields

$$p(\mathbf{R}) = \prod_{a=1}^A \prod_{j=1}^K p(\mathbf{r}_j^a) = \prod_{a=1}^A \prod_{j=1}^K \text{Dir}(\mathbf{r}_j^a | \alpha_{1j}^a, \dots, \alpha_{Kj}^a), \quad (3)$$

where  $\mathbf{r}_j^a = (r_{1j}^a, \dots, r_{Kj}^a)^\top$  denotes the  $j$ -th column of the confusion matrix  $\mathbf{R}^a$ . The hyperparameters  $\alpha = \{\alpha_{ij}^a : i, j = 1, \dots, K, a = 1, \dots, A\}$  codify any prior belief on the behavior of the annotators. The use of a prior protects from the so-called *black swan paradox* [44, Section 3.3.4.1], i.e. when trying to estimate  $\mathbf{r}_j^a$  for an annotator who provided no annotations for samples in the  $j$ -th class. As we will see in eq. (15), the prior affects the training through a KL divergence term that couples it to the posterior. If no prior knowledge is available, the default choice  $\alpha_{ij}^a = 1$  corresponds to uniform distributions. This is the most standard scenario, and the one that is considered here. Yet, notice that this would not be the optimal choice if there were very few annotations, since the prior could hide the effect of the observed data. This is not the case of LIGO, where there is an average of over 500 annotations per user (see Section 3). Additionally, the prior allows for more informative modelling. For instance, an annotator who is known to mix up two classes can be modelled by setting the corresponding values of  $\alpha_{ij}^a$ .

For each instance, the true underlying label  $\mathbf{z}_n$  is modelled through  $K$  latent variables  $f_{n1}, \dots, f_{nK}$ . Both parts are related by means of the likelihood model

$$p(\mathbf{z}_n | f_{n1}, \dots, f_{nK}) = \mathbf{z}_n^\top \boldsymbol{\nu}(f_{n1}, \dots, f_{nK}), \quad (4)$$

where  $\boldsymbol{\nu}(f_{n1}, \dots, f_{nK})$  is any vector with  $K$  positive components that add up to 1. In this work we will use the popular robust-max likelihood [45]. It is given by  $\boldsymbol{\nu}(a_1, \dots, a_K) = (\nu_1, \dots, \nu_K)$ , with  $\nu_i = 1 - \varepsilon$  for  $i = \arg \max(a_1, \dots, a_K)$  and  $\nu_j = \varepsilon / (K - 1)$  for  $j \neq i$ . The value of  $\varepsilon$  is fixed to the default value  $10^{-3}$ . This likelihood is implemented in the GPflow library [36], and in practice it can be substituted by any other one available in GPflow. For instance the soft-max likelihood, which generalizes the sigmoid likelihood to multi-class, i.e.  $\nu_i = e^{a_i} / \sum_j e^{a_j}$ .

Assuming that the underlying real labels for the different instances are independent given the latent variables, it is

$$p(\mathbf{Z} | \mathbf{F}) = \prod_{n=1}^N p(\mathbf{z}_n | \mathbf{f}_{n,:}), \quad (5)$$

where  $p(\mathbf{z}_n | \mathbf{f}_{n,:})$  is given by eq. (4), and  $\mathbf{F}$  gathers the  $K$  latent variables for the  $N$  instances. Specifically,  $\mathbf{F}$  is a  $N \times K$  matrix, whose  $(n, k)$  term is the value of the  $k$ -th latent variable for the  $n$ -th instance. As usual, the  $n$ -th row of  $\mathbf{F}$  is denoted by  $\mathbf{f}_{n,:}$ , and the  $k$ -th column by  $\mathbf{f}_{:,k}$ .



Finally, independent GP priors are utilized for the latent variables  $f_1, \dots, f_K$ . This yields the joint prior

$$p(\mathbf{F}|\Theta, \mathbf{X}) = \prod_{k=1}^K p(\mathbf{f}_k|\theta_k, \mathbf{X}) = \prod_{k=1}^K \mathcal{N}(\mathbf{f}_k|\mathbf{0}, \mathbf{K}_{\theta_k}(\mathbf{X}, \mathbf{X})), \quad (6)$$

where  $\theta_k$  are the kernel hyperparameters for the  $k$ -th GP. In this work we will use the well-known squared exponential kernel,  $k(\mathbf{x}, \mathbf{y}) = \gamma \cdot \exp(-\|\mathbf{x} - \mathbf{y}\|^2/(2\sigma^2))$ , which has the hyperparameters of variance  $\gamma$  and length-scale  $\sigma$ . However, as before, GPflow allows us to use any other kernel [36].

In summary, the full probabilistic model is given by

$$p(\mathbf{Y}, \mathbf{Z}, \mathbf{F}, \mathbf{U}, \mathbf{R}|\Theta) = p(\mathbf{Y}|\mathbf{Z}, \mathbf{R})p(\mathbf{R})p(\mathbf{Z}|\mathbf{F})p(\mathbf{F}|\Theta), \quad (7)$$

with the four factors on the right hand side defined through eqs. (2), (3), (5) and (6), respectively. Here, the dependency on the observed features  $\mathbf{X}$  has been omitted for simplicity.

In order to introduce the sparse GP approximation, let us expand this model by introducing  $M$  inducing points for each GP. Namely, each GP prior  $p(\mathbf{f}_k)$  can be naively rewritten as the marginal of  $p(\mathbf{f}_k, \mathbf{u}_k)$ , where  $\mathbf{u}_k = (u_{1k}, \dots, u_{Mk})$  are  $M$  inducing points. These represent the value of the  $k$ -th GP on  $M$  new locations called inducing inputs,  $\tilde{\mathbf{X}} = \{\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_M\}$ , just like  $\mathbf{f}_k$  does for  $\mathbf{X}$ , i.e.  $p(\mathbf{u}_k) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{\theta_k}(\tilde{\mathbf{X}}, \tilde{\mathbf{X}}))$ .<sup>4</sup> Analogously to  $\mathbf{F}$ , we write  $\mathbf{U}$  for the  $M \times K$  matrix gathering all the inducing points, whose rows and columns are denoted by  $\mathbf{u}_{m,:}$  and  $\mathbf{u}_k$  respectively. The idea is that, in sparse GPs, these inducing points are used to summarize the information from the training data. By taking  $M \ll N$ , computational tractability can be achieved.

Then, if the joint GP  $p(\mathbf{f}_k, \mathbf{u}_k)$  is factorized as  $p(\mathbf{f}_k|\mathbf{u}_k)p(\mathbf{u}_k)$ , the model in eq. (7) can be analogously rewritten as

$$p(\mathbf{Y}, \mathbf{Z}, \mathbf{F}, \mathbf{U}, \mathbf{R}|\Theta) = p(\mathbf{Y}|\mathbf{Z}, \mathbf{R}) \cdot p(\mathbf{R}) \cdot p(\mathbf{Z}|\mathbf{F}) \times p(\mathbf{F}|\mathbf{U}, \Theta) \cdot p(\mathbf{U}|\Theta), \quad (8)$$

where the Gaussian conditional  $p(\mathbf{F}|\mathbf{U}, \Theta)$  is given by  $\prod_k \mathcal{N}(\mathbf{f}_k|\mathbf{B}_k\mathbf{u}_k, \mathbf{K}_{\theta_k}(\mathbf{X}, \mathbf{X}) - \mathbf{B}_k\mathbf{K}_{\theta_k}(\tilde{\mathbf{X}}, \mathbf{X}))$  and  $\mathbf{B}_k = \mathbf{K}_{\theta_k}(\mathbf{X}, \tilde{\mathbf{X}})[\mathbf{K}_{\theta_k}(\tilde{\mathbf{X}}, \tilde{\mathbf{X}})]^{-1}$ .

It is worth stressing that, by marginalizing out  $\mathbf{U}$ , this model is equivalent to the one in eq. (7). This is important because sparse GP approximations are grouped into two big categories: those which approximate the model and perform exact inference (like FITC [27]), and those which keep the model unaltered and introduce the approximation at the inference step. Our approach, like SVGP, belongs to the second group, and the approximation is carried out next.

## 2.2 Variational inference

Given the model in eq. (8), an exact solution would involve calculating the marginal likelihood  $p(\mathbf{Y}|\Theta)$ , in order to estimate the optimal kernel hyperparameters  $\Theta$  and then obtain the posterior  $p(\mathbf{Z}, \mathbf{F}, \mathbf{U}, \mathbf{R}|\mathbf{Y}, \Theta)$ . However, integrating out  $\mathbf{Z}, \mathbf{F}, \mathbf{U}$  and  $\mathbf{R}$  is analytically intractable, and we resort to variational inference to approximate the computations [32].

4. Notice that the inducing locations  $\tilde{\mathbf{X}}$  do not depend on  $k$ . Although different inducing locations could be used for each GP, in practice they are usually considered the same. However, the inducing points  $\mathbf{u}_k$  do depend on  $k$ , as each GP models a different function.

The core of variational inference is the following decomposition of the log marginal likelihood, which is straightforward and holds for any distribution  $q(\mathbf{z}, \mathbf{F}, \mathbf{U}, \mathbf{R})$ <sup>5</sup>:

$$\begin{aligned} \log p(\mathbf{Y}|\Theta) &= \text{KL}(q(\mathbf{Z}, \mathbf{F}, \mathbf{U}, \mathbf{R})||p(\mathbf{Z}, \mathbf{F}, \mathbf{U}, \mathbf{R}|\mathbf{Y}, \Theta)) \\ &+ \underbrace{\int q(\mathbf{Z}, \mathbf{F}, \mathbf{U}, \mathbf{R}) \log \frac{p(\mathbf{Y}, \mathbf{Z}, \mathbf{F}, \mathbf{U}, \mathbf{R}|\Theta)}{q(\mathbf{Z}, \mathbf{F}, \mathbf{U}, \mathbf{R})} d\mathbf{Z}d\mathbf{F}d\mathbf{U}d\mathbf{R}}_{\text{ELBO}}. \end{aligned} \quad (9)$$

This distribution  $q$  must be understood as an approximation to the true posterior  $p(\mathbf{Z}, \mathbf{F}, \mathbf{U}, \mathbf{R}|\mathbf{Y}, \Theta)$ . The second term in the right hand side of eq. (9) is called the Evidence Lower Bound (ELBO), since it is a lower bound for the model evidence or log marginal likelihood  $\log p(\mathbf{Y}|\Theta)$  (recall that the first term, the KL divergence, is always non-negative, and is zero if and only if both distributions coincide).

The idea of variational inference is to propose a parametric form for  $q$ . Then, the ELBO in eq. (9) is maximized with respect to these new variational parameters, the kernel hyperparameters  $\Theta$ , and the inducing locations  $\tilde{\mathbf{X}}$  (which are not usually considered fixed). Notice that, by maximizing the ELBO, we are at the same time considering the log marginal likelihood  $\log p(\mathbf{Y}|\Theta)$  and the KL divergence between  $q$  and the real posterior (just solve for the ELBO in eq. (9)). Thus, variational inference converts the problem of posterior distribution approximation into an optimization one [32], which in practice is addressed through optimization algorithms such as Adam Optimizer [46].

Here, the following parametric form is proposed for  $q$ :

$$q(\mathbf{Z}, \mathbf{F}, \mathbf{U}, \mathbf{R}) = q(\mathbf{Z})q(\mathbf{F}|\mathbf{U}, \Theta)q(\mathbf{U})q(\mathbf{R}), \quad \text{with} \quad (10)$$

$$q(\mathbf{Z}) = \prod_{n=1}^N q(\mathbf{z}_n) = \prod_{n=1}^N \mathbf{z}_n^T \mathbf{q}_n, \quad (11)$$

$$q(\mathbf{F}|\mathbf{U}, \Theta) = p(\mathbf{F}|\mathbf{U}, \Theta), \quad (12)$$

$$q(\mathbf{U}) = \prod_{k=1}^K q(\mathbf{u}_k) = \prod_{k=1}^K \mathcal{N}(\mathbf{u}_k|\mathbf{m}_k, \mathbf{S}_k), \quad (13)$$

$$q(\mathbf{R}) = \prod_{a=1}^A \prod_{k=1}^K q(\mathbf{r}_k^a) = \prod_{a=1}^A \prod_{j=1}^K \text{Dir}(\mathbf{r}_j^a|\tilde{\alpha}_{1j}^a, \dots, \tilde{\alpha}_{Kj}^a). \quad (14)$$

The proposed posterior on  $\mathbf{Z}$  factorizes across data points, and each  $\mathbf{q}_n = (q_{n1}, \dots, q_{nK}) \in [0, 1]^K$  describes the probability that  $K$  is the real class for  $\mathbf{x}_n$  (i.e.,  $\sum_k q_{nk} = 1$ ). The prior conditional  $\mathbf{F}|\mathbf{U}$  does not introduce any new variational parameter. The posterior on  $\mathbf{U}$  factorizes across dimensions, and each one is given by a Gaussian with mean  $\mathbf{m}_k \in \mathbb{R}^M$  and (positive-definite) covariance matrix  $\mathbf{S}_k \in \mathbb{R}^{M \times M}$ . Finally,  $q(\mathbf{R})$  factorizes across annotators and dimensions, and they are assigned Dirichlet distributions with parameters  $\tilde{\alpha}_{ij}^a > 0$ . In the sequel, all these variational parameters  $\{\mathbf{q}_n : n = 1, \dots, N\}$ ,  $\{\mathbf{m}_k, \mathbf{S}_k : k = 1, \dots, K\}$ ,  $\{\tilde{\alpha}_{ij}^a : i, j = 1, \dots, K; a = 1, \dots, A\}$  will be denoted by  $\mathbf{V}$ .

In the proposed form described by eqs. (10)–(14), the prior conditional  $p(\mathbf{F}|\mathbf{U}, \Theta)$  arises in a natural way if the GP values are assumed conditionally independent on any other value given the inducing points  $\mathbf{U}$ . This is the original assumption of Titsias in [28], and intuitively implies that all

5. Observe that, in order to “lighten” the notation, we use the integral symbol also for the discrete variable  $\mathbf{Z}$ .

TABLE 1

Specifying the dependence of the ELBO on the variational parameters, the kernel hyperparameters, and the inducing locations through its five terms in eq. (15).

ELBO term	Parameters it depends on
$\sum q_{nk} \mathbb{E}_{q(\mathbf{r}_k^a)} [\log p(\mathbf{y} \mathbf{e}_k, \mathbf{r}_k^a)]$	$q_{nk}, \tilde{\alpha}_{ij}^a$
$\sum q_{nk} \mathbb{E}_{q(\mathbf{f}_{n,:})} [\log p(\mathbf{e}_k \mathbf{f}_{n,:})]$	$q_{nk}, \mathbf{m}_k, \mathbf{S}_k, \boldsymbol{\Theta}, \tilde{\mathbf{X}}$
$\sum q_{nk} \log q_{nk}$	$q_{nk}$
$\sum \text{KL}(q(\mathbf{u}_k)  p(\mathbf{u}_k))$	$\mathbf{m}_k, \mathbf{S}_k, \boldsymbol{\Theta}, \tilde{\mathbf{X}}$
$\sum \text{KL}(q(\mathbf{r}_k^a)  p(\mathbf{r}_k^a))$	$\tilde{\alpha}_{ij}^a$

the information is condensed by and propagated through the inducing points  $\mathbf{U}$ . This form of  $\mathbf{F}|\mathbf{U}$ , plus that of  $q(\mathbf{U})$ , are at the core of the sparse GP approximation that we are inspired by, SVGP [31]. The distributions  $q(\mathbf{Z})$  and  $q(\mathbf{R})$  are given the functional form that would arise if a mean-field approach was applied [43, Eq. (10.9)]. For that, the conjugacy between the Dirichlet distribution in  $p(\mathbf{R})$  and the categorical in  $p(\mathbf{Y}|\mathbf{Z}, \mathbf{R})$  is essential.

Now, we can compute the explicit expression for the ELBO in our case, which must be maximized w.r.t.  $\mathbf{V}$ ,  $\boldsymbol{\Theta}$ , and  $\tilde{\mathbf{X}}$ :

$$\begin{aligned}
\text{ELBO}(\mathbf{V}, \boldsymbol{\Theta}, \tilde{\mathbf{X}}) &= \mathbb{E}_{q(\mathbf{Z})p(\mathbf{F}|\mathbf{U})q(\mathbf{U})q(\mathbf{R})} \log \frac{p(\mathbf{Y}|\mathbf{Z}, \mathbf{R})p(\mathbf{Z}|\mathbf{F})p(\mathbf{F}|\mathbf{U})p(\mathbf{U})p(\mathbf{R})}{q(\mathbf{Z})p(\mathbf{F}|\mathbf{U})q(\mathbf{U})q(\mathbf{R})} \\
&= \sum_{n=1}^N \sum_{a \in A_n} \sum_{\mathbf{y} \in \mathbf{Y}_n^a} \sum_{k=1}^K q_{nk} \mathbb{E}_{q(\mathbf{r}_k^a)} [\log p(\mathbf{y}|\mathbf{e}_k, \mathbf{r}_k^a)] \\
&+ \sum_{n=1}^N \sum_{k=1}^K q_{nk} \mathbb{E}_{q(\mathbf{f}_{n,:})} [\log p(\mathbf{e}_k|\mathbf{f}_{n,:})] - \sum_{n=1}^N \sum_{k=1}^K q_{nk} \log q_{nk} \\
&- \sum_{k=1}^K \text{KL}(q(\mathbf{u}_k)||p(\mathbf{u}_k)) - \sum_{a=1}^A \sum_{k=1}^K \text{KL}(q(\mathbf{r}_k^a)||p(\mathbf{r}_k^a)), \quad (15)
\end{aligned}$$

A detailed derivation of this expression is provided in the supplemental material. Notice that the inclusion of the prior conditional  $p(\mathbf{F}|\mathbf{U})$  in the approximate posterior makes the highlighted cancellation possible, which is essential for the scalability of the method. All these five terms in eq. (15) but the second one can be expressed in closed-form as a function of  $\mathbf{V}$ ,  $\boldsymbol{\Theta}$ , and  $\tilde{\mathbf{X}}$ . Similarly, the second one can be approximated explicitly through Gaussian-Hermite quadrature [47], which is already implemented in GPflow for many different likelihoods (like the robust-max used here) [36]. Further details and the specific expressions can be found in the supplemental material. As a summary, Table 1 shows which parameters each term in eq. (15) depends on.

Importantly, observe that the expression for the ELBO factorizes across data points, which allows for stochastic optimization through mini-batches [35]. To the best of our knowledge, this allows GP-based crowdsourcing methods to scale up to previously prohibitive data sets for the first time. More specifically, the computational complexity to evaluate the ELBO in eq. (15) in terms of the training set size is  $\mathcal{O}(N_b(M^2 + A_b K))$ , where  $N_b$  is the mini-batch size,  $M$  the number of inducing points,  $K$  the number of classes, and  $A_b$  the number of annotations per instance in

the mini-batch. Although this is theoretically linear in  $N$ , the amount of inducing points  $M$  might grow with  $N$ . An interesting alternative, which is based on inference networks and addresses this issue, will be presented in Section 5.2. It is also interesting to compare eq. (15) with the expression for the ELBO in SVGP [31, Eq. (19)]. The second and fourth terms, which come from the prior and the classification likelihood, are analogous to the two terms in [31]. The other three terms arise naturally from the crowdsourcing modelling.

Once the ELBO is maximized w.r.t.  $\mathbf{V}$ ,  $\boldsymbol{\Theta}$  and  $\tilde{\mathbf{X}}$ , we can make predictions for previously unseen data points. Given a new  $\mathbf{x}^*$ , we have

$$\begin{aligned}
p(f_k^*|\mathbf{x}^*, \mathcal{D}) &= \int p(f_k^*|\mathbf{u}_k)p(\mathbf{u}_k|\mathcal{D})d\mathbf{u} \approx \mathbb{E}_{q(\mathbf{u}_k)} p(f_k^*|\mathbf{u}_k) \\
&= \mathcal{N}(f_k^*|\mathbf{B}_{\mathbf{x}^* \tilde{\mathbf{X}}} \mathbf{m}_k, k_{\mathbf{x}^* \mathbf{x}^*} + \mathbf{B}_{\mathbf{x}^* \tilde{\mathbf{X}}}(\mathbf{S}_k - \mathbf{K}_{\tilde{\mathbf{X}} \tilde{\mathbf{X}}})\mathbf{B}_{\tilde{\mathbf{X}} \mathbf{x}^*}), \quad (16)
\end{aligned}$$

where  $\mathbf{B}_{\mathbf{x}^* \tilde{\mathbf{X}}}$  stands for  $\mathbf{K}_{\mathbf{x}^* \tilde{\mathbf{X}}} \mathbf{K}_{\tilde{\mathbf{X}} \tilde{\mathbf{X}}}^{-1}$ , and we are using the values of  $\mathbf{m}_k$ ,  $\mathbf{S}_k$ ,  $\boldsymbol{\Theta}$ , and  $\tilde{\mathbf{X}}$  estimated after training. The predictive distribution on the real label  $\mathbf{z}^*$  is obtained as  $p(\mathbf{z}^*) = \int p(\mathbf{z}^*|\mathbf{f}^*)p(\mathbf{f}^*)d\mathbf{f}^*$ . For classification likelihoods like ours, this is computed by GPflow through Gaussian-Hermite quadrature. Moreover, as we will illustrate in the experiments, the posterior distributions  $q(\mathbf{Z})$  and  $q(\mathbf{R})$  provide an estimation for the underlying real label of the training points and for the annotators degree of expertise, respectively. Finally, in order to exploit GPU acceleration through TensorFlow, the novel SVGPCR is implemented within the popular GP framework GPflow [36]. The code will be made publicly available in GitHub upon acceptance of the paper, and will be listed in the “projects using GPflow” section of the GPflow site <https://github.com/GPflow/GPflow>.

### 3 LIGO DATA DESCRIPTION

The Laser Interferometer Gravitational-Waves Observatory (LIGO) is a large-scale physics experiment and observatory to detect gravitational waves (GWs) [11]. These are ripples in the space-time produced by non-symmetric movements of masses, being their energy much higher for events such as binary black holes or neutron stars mergers. Their existence is a direct consequence of the General Relativity theory postulated in 1916. However, Albert Einstein himself believed they would be extremely difficult to detect by any technology foreseen at that time [48].

The first direct observation of GWs was made one hundred years later by LIGO, on September 14th, 2015. The discovery had a tremendous impact in the scientific community. Not only as an empirical validation of one of the most recognized Physics theories, but also as a whole new way to explore the universe. So far, astrophysicists could perceive the outer space only through one “sense” (electromagnetic radiation), but were “deaf” to GWs. This detection has inaugurated a new era of the so-called GWs astronomy, and has been awarded the 2017 Physics Nobel Prize [12].

To identify GWs, LIGO is able to detect changes of the length of a 4 kilometers arm by a thousandth of the width of a proton [11]. This is proportionally equivalent

to changing the distance to the nearest star outside the Solar System by one hair’s width. Such precision requires cutting-edge technology that is also extremely sensitive to different instrumental and environmental sources of noise. In the spectrograms that astrophysicists analyze to search for GWs, this contamination manifests itself in the form of glitches, which are noisy patterns that adopt many different morphologies [10]. The presence of these glitches hinders the detection of true GWs. Figure 3 shows the 15 types of glitches considered in this work, which will be later described.

The goal of the GravitySpy project is to develop a system to accurately classify the different types of glitches [10]. This would help astrophysicists to gain insights on their taxonomy and potential causes, enhancing detection of true GWs. Since LIGO produces a constant stream of data, GravitySpy leverages crowdsourcing techniques through the Zooniverse platform in order to label a training set <https://www.zooniverse.org/projects/zooniverse/gravity-spy>. Then, machine learning crowdsourcing algorithms that can learn from this multiple-annotated data must be applied (like the SVGPCR presented here).

Our training set contains 173565 instances (glitches) and 1828981 annotations (i.e., a mean value of more than 10 labels per instance), which have been provided by 3443 collaborators through the Zooniverse platform. These instances are time-frequency plots (spectrograms) like those in figures 1 and 3, taken with four time windows. For each one, we will use 256 relevant features extracted in [49]. These glitches have been classified into 15 different classes proposed by astrophysicists (recall figure 3). Next, we provide a brief description of them (see [49] for a more detailed explanation).

**1080 Line:** It appears as a string of short yellow dots, always around 1080Hz. It was reduced after an update on 2017, although it is still present.

**1400 Ripple:** Glitches of 0.05s or longer around 1400Hz. So far, their origin is unknown. They are commonly confused with 1080Line and Violin Mode Harmonic.

**Blip:** Short glitches with a symmetric “teardrop” shape in time-frequency. Blips are extremely important since they hamper the detection of binary black hole mergers [50].

**Extremely Loud:** These are caused by major disturbances, such as an actuator reaching the end of its range and “railing”, or a photodiode “saturating”. They look very bright, due to their very high energy.

**Koifish:** Similar to Blips, but resemble a fish with the head at the low frequency end, pectoral fins around 30 Hz, and a thin tail around 500Hz. LIGO scientists do not understand the physical origin of this glitch.

**Low Frequency Burst:** Resembles a hump with a nearly triangular shape growing from low frequency to a peak, and then dying back down in one or two seconds. It is caused by scattered light driven by motion of the output mirrors.

**Low Frequency Lines:** These appear as horizontal lines at low frequencies. Can be confused with Scattered Light (the latter shows some curvature) and Low Frequency Bursts (the former continues to look like a line in the 4s window).

**No Glitch:** No glitch refers to images that do not have any glitch visible at all. The spectrograms would appear

dark blue with only small fluctuations.

**Other:** This category is a catch-all for glitches that do not fit into the other categories. Therefore, it presents a great variability in its morphology.

**Power-line 60Hz:** In US, the mains power is alternating current at 60Hz. When equipment running on this power switches on or off, glitches can occur at 60Hz or harmonics (120, 180...). These glitches usually look narrow in frequency, centered around 60Hz or harmonics.

**Repeating blips:** Analogous to blips, but repeat at regular intervals, usually every 0.125, 0.25 or 0.5 seconds.

**Scattered Light:** After hitting optical components, some light from LIGO beam is scattered. It may then reflect off of other objects and re-enter the beam with a different phase. It usually looks like upward humps, with frequency below 30 Hz. It hinders searches of binary neutron stars, neutron star black hole binaries, and binary black holes.

**Scratchy:** Wide band of mid-frequency signals that looks like a ripply path through time-frequency space. This glitch hampers searches for binary black hole mergers.

**Violin Mode Harmonic:** Test masses in LIGO are suspended from fibers with resonances. These are called violin modes, as they resemble violin strings resonances. Thermal excitations of the fibers produce movements at the violin mode frequencies, centered around 500Hz. Thus, these glitches are short and located around 500 Hz and harmonics.

**Whistle:** Usually appear with a characteristic W or V shape. Caused by radio frequency signals beating with the LIGO Voltage Controlled Oscillators. Whistles mainly contaminate searches for binary black hole mergers [51].

For testing purposes, the astrophysicists at GravitySpy have labelled a set of 9997 instances, including glitches from all the 15 types explained above.

## 4 EXPERIMENTAL RESULTS

In this section, the proposed SVGPCR is empirically validated and compared against current crowdsourcing methods, with a special focus on the LIGO data introduced in the previous section. Three blocks of experiments are presented in sections 4.1, 4.2 and 4.3. Firstly, the behavior of SVGPCR is thoroughly analyzed in a controlled crowdsourcing experiment based on the popular MNIST set. Secondly, SVGPCR is compared with previous probabilistic (mainly GP-based) approaches on the LIGO data. Since most of these methods were proposed for binary problems, we consider a binary task relevant to the GravitySpy project. Thirdly, SVGPCR is compared against state-of-the-art DL-based crowdsourcing methods in the full LIGO data set.

### 4.1 Understanding the proposed method

Before comparing against other crowdsourcing methodologies, let us analyze the behavior and main properties of SVGPCR. To do so, we simulate five different crowdsourcing annotators for the well-known MNIST data set. The availability of simulated annotators and real training labels on this graphic data set constitutes a controlled setting that allows for a comprehensive analysis.

We use the standard train/test split of MNIST with 60K/10K hand-written digits from 0 to 9 (multi-class problem with 10 classes) [52]. Notice that 60K training instances



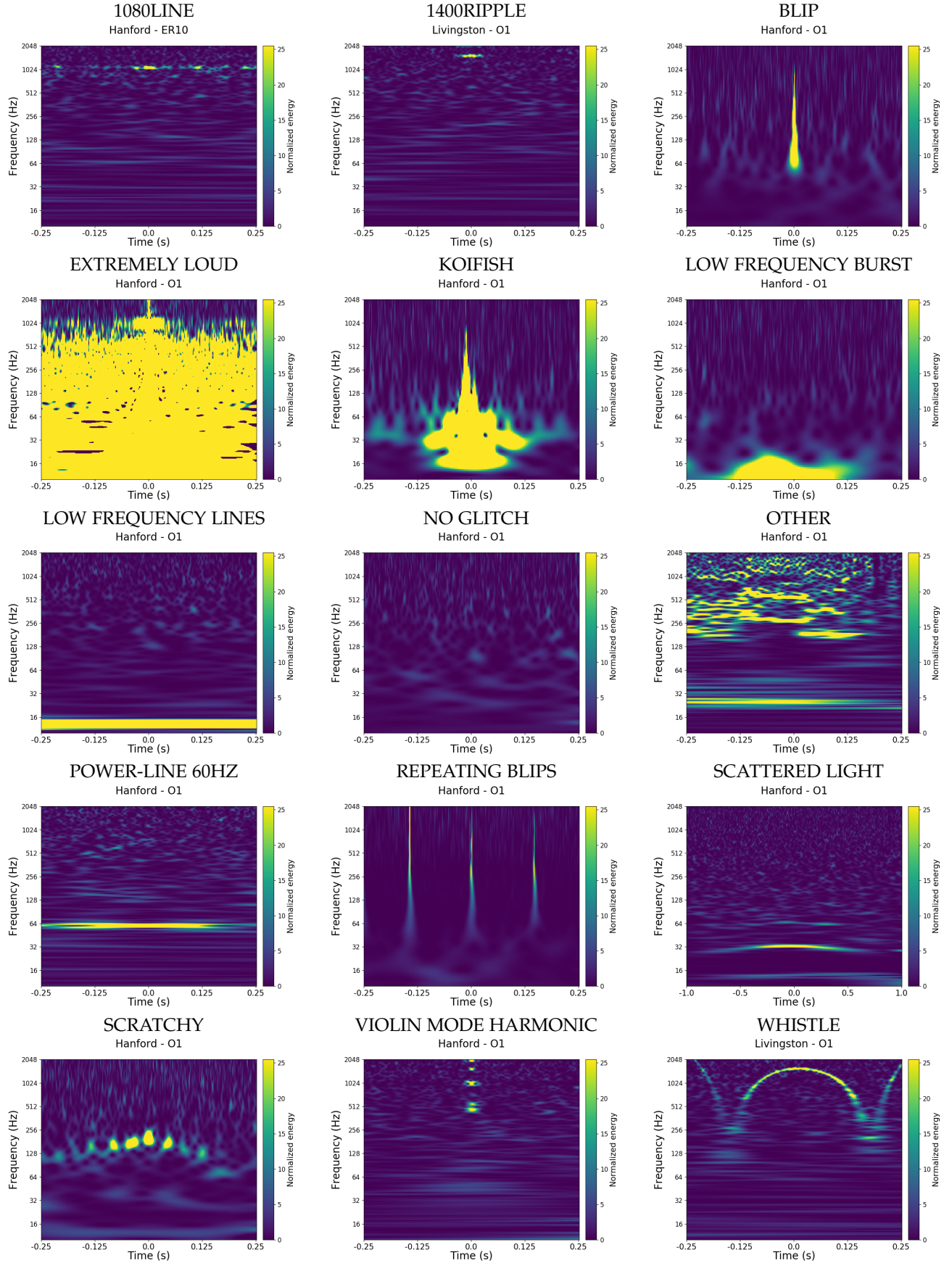


Fig. 3. Representative spectrograms for the 15 different types of glitches considered in this work. Hanford and Livingston refer to the two observatories that LIGO comprises, and ER10/O1 to two different observation runs. A brief description of each glitch is provided in the text. The goal of the GravitySpy project is to learn a machine learning system to automatically classify these glitches. The labels for the training set are obtained through crowdsourcing.

TABLE 2  
Per-class and global test performance of SVGPCR and SVGP-gold in the MNIST problem. In spite of the corrupted labels, the proposed method almost recovers the golden results.

	Test accuracy		Test likelihood	
	SVGP-gold	SVGPCR	SVGP-gold	SVGPCR
0	0.9867	<b>0.9898</b>	0.9777	<b>0.9781</b>
1	<b>0.9885</b>	0.9877	<b>0.9853</b>	0.9845
2	0.9525	<b>0.9535</b>	<b>0.9368</b>	0.9345
3	0.9703	<b>0.9733</b>	0.9475	<b>0.9555</b>
4	0.9715	<b>0.9735</b>	0.9548	<b>0.9570</b>
5	<b>0.9630</b>	0.9540	<b>0.9397</b>	0.9352
6	<b>0.9749</b>	<b>0.9749</b>	<b>0.9604</b>	0.9585
7	<b>0.9591</b>	0.9543	0.9418	<b>0.9432</b>
8	<b>0.9620</b>	<b>0.9620</b>	<b>0.9458</b>	0.9445
9	<b>0.9346</b>	0.9316	<b>0.9217</b>	0.9186
Global	<b>0.9665</b>	0.9657	<b>0.9515</b>	0.9514

is already prohibitive for standard GPs. Five decreasingly reliable annotators are considered. The first one has a 95% accuracy for each class, that is,  $r_{jj}^1 = 0.95$  for  $j = 0, \dots, 9$  (the rest of values,  $r_{ij}^1, i \neq j$ , are randomly assigned to add the remaining 0.05 probability by columns). The second and third ones are defined analogously, but with 90% and 80% accuracy, respectively. The fourth one is a *spammer* annotator, that is,  $r_{ij}^4 = 0.1$  for all  $i, j = 0, \dots, 9$ . This implies that, regardless of the real class, this annotator assigns a random label. The fifth one is an *adversarial* annotator. Specifically, in this case, with a 90% of probability, the annotator labels as  $(i + 1)$ -th class an instance whose real class is the  $i$ th (samples in class 9 are assigned to class 0). The confusion matrices for these annotators are depicted in the first row of figure 4. The five annotators label all the instances, which yields 300K annotations that are used to train SVGPCR.

Since we have available the true labels for the training instances, let us start by comparing SVGPCR with its theoretical upper bound, namely SVGP trained with the true labels, which we refer to as SVGP-gold. Table 2 shows the global and per-class test accuracy and test likelihood for both approaches. Importantly, notice that the results are very similar for all classes and both metrics, and SVGPCR almost reaches the same global performance as SVGP-gold (in spite of the corrupted labels provided by annotators).

This excellent performance of SVGPCR can be explained by its accurate prediction of the annotators behavior, which in turn allows SVGPCR to properly reconstruct the underlying true labels from the noisy annotations. Indeed, firstly, figure 4 shows the exceptional estimations obtained by SVGPCR for the annotators confusion matrices. Recall from eq. (14) that the expertise degree of annotators is estimated through posterior Dirichlet distributions. The bottom row of figure 4 shows the mean of those distributions. Interestingly, the maximum variance was 0.0016, which implies a high degree of certainty about the predictions in figure 4. Secondly, as previously mentioned, this allows SVGPCR to correctly puzzle out the underlying true labels from the noisy annotations. In fact, table 3 shows the excellent per-class and global performance of SVGPCR in this sense (recall that SVGPCR estimates the underlying true labels through the approximate posterior  $q(\mathbf{z})$  in eq. (11)).

More in depth, we have analyzed the 20 examples where SVGPCR fails to reconstruct the true label, and some of them can be certainly considered as not-easy ones. Figure 5 shows four of them, along with the probabilities assigned by SVGPCR for each one. In all cases, the true label is assigned the second highest probability by SVGPCR, and the digit presents some feature which certainly leads to confusion with the class that SVGPCR assigns more probability to.

Another key aspect of SVGPCR is the role of the inducing points. In this example we are using  $M = 100$ , and the next experiment will be devoted to analyze the influence of  $M$  in the performance of SVGPCR. But before, figure 6 shows the locations to which 30 out of the 100 inducing points have converged after training (recall that the ELBO in eq. (15) is also maximized w.r.t. the inducing locations  $\tilde{\mathbf{X}}$ ). For instance, the first column shows the locations of three inducing points which are classified as 0 by SVGPCR (according to the estimated  $\mathbf{m}_k$ , recall eq. (13)), and analogously for the rest of the columns. It is very interesting to notice that the inducing point locations comprise different calligraphic representations (in terms of shape, orientation and thickness) of the same digit. This is related to their intuitive role of entities that summarize the training data.

Next, let us study the influence of  $M$  (the number of inducing points) on the behavior of SVGPCR. Figure 7 shows the dependence on  $M$  of four different metrics: two measures of the test performance (accuracy and mean likelihood), and two related to the computational cost (at training and test steps). As expected from the theoretical formulation in section 2, a greater number of inducing points implies a higher performance at test (in both accuracy and mean likelihood), since the expressiveness of the model is higher. However, this also leads to heavier train and test costs, since there are more parameters to be estimated (inducing locations  $\tilde{\mathbf{X}}$ ,  $\mathbf{m}_k$ , and  $\mathbf{S}_k$ ), and the size of several matrices increase.

Moreover, for a given  $M$ , the model is expected to obtain better test performance as the training time evolves (i.e., when more epochs are run). In order to further investigate this, figure 8 shows the test accuracy of SVGPCR as the training time evolves, for different values of  $M$ . It is interesting to observe that, the more inducing points, the higher values of test accuracy can be potentially reached, but also a greater amount of training time is needed to reach that precision (notice that the steps which take  $M = 100, 250, 500$  to the level of their final precision happen increasingly later). The conclusion is that, for a given computational budget, the  $M$  to be selected is the highest one that can reach convergence in that time (logically, assuming that it allows for the inversion of the associated kernel matrix, i.e., usually  $M < 10^4$ ).

Finally, since the associated code can leverage GPU acceleration through GPflow [36], let us compare CPU and GPU implementations. Figure 9 shows that, for training, the GPU is usually the preferred choice, unless the minibatch size is very small, in which case the amount of memory copies from CPU to GPU does not compensate the advantage provided by the latter. In test, the GPU is always faster, since it involves much less data transfers from CPU to GPU.



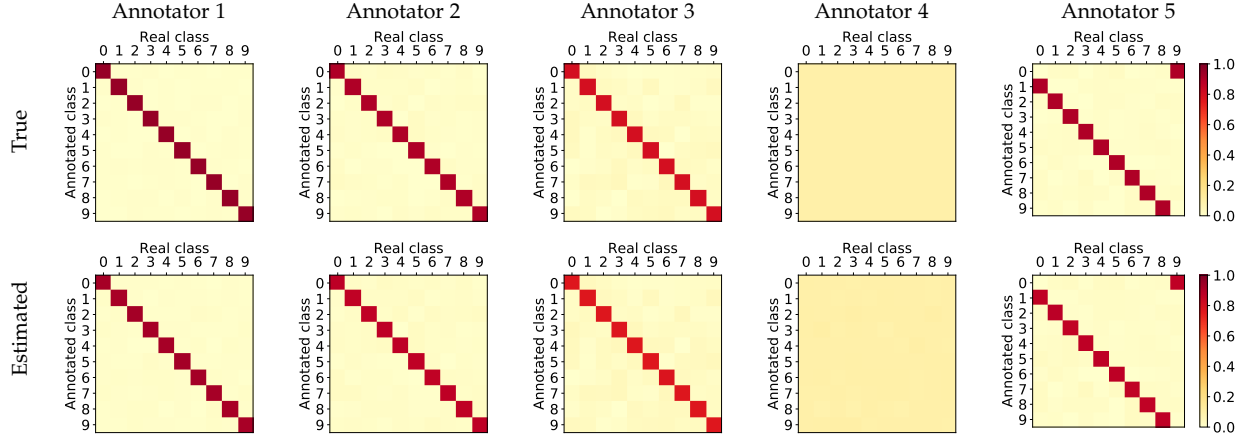


Fig. 4. Estimation of the expertise degree of the simulated annotators in the MNIST problem by SVGPCR. Upper row: true confusion matrices. Lower row: mean of the estimated distribution for the confusion matrices. Notice that the proposed method perfectly identifies adversarial (fifth) and spammer (fourth) annotators. Moreover, not only the structure of the matrices is well identified, but also the actual values (the intensity of color is very similar).

TABLE 3

Per-class and global performance of SVGPCR to reconstruct the underlying true label for training instances in the MNIST problem. An excellent result is obtained across all the classes, with only 20 (out of the 60000 training examples) not correctly predicted.

	0	1	2	3	4	5	6	7	8	9	Global
Accuracy	0.9998	0.9997	1.0000	0.9995	0.9995	0.9993	1.0000	0.9997	0.9997	0.9995	0.9997
Likelihood	0.9997	0.9996	0.9997	0.9992	0.9994	0.9993	0.9999	0.9995	0.9996	0.9991	0.9995

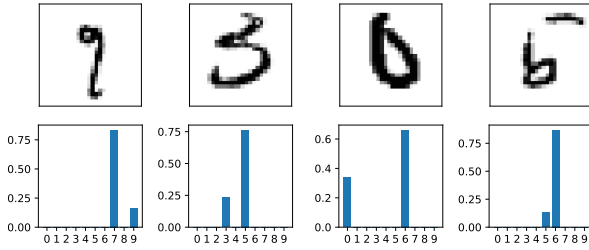


Fig. 5. Upper row: four (out of the 20) examples for which SVGPCR is not able to reconstruct the real underlying class in the MNIST problem. Lower row: the corresponding probabilities assigned by SVGPCR. In all cases, the proposed method assigns the second highest probability to the real class. Notice that some of these examples are not easy, and have some features which might lead to confusion.

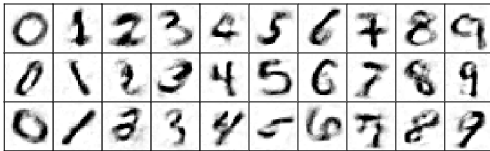


Fig. 6. Some of the inducing point locations learned by SVGPCR in the MNIST problem. They have been arranged by columns based on their classification. Notice that, for each digit (column), different representative patterns are learned in terms of shape, orientation and thickness.

## 4.2 Comparison to classical probabilistic approaches

As explained in section 1, the most popular approaches to crowdsourcing jointly model a classifier for the underlying true labels along with the annotators' behavior. The first works used basic logistic regression as the classifier, e.g. *Raykar* [19] and *Yan* [20] (the difference between them is the

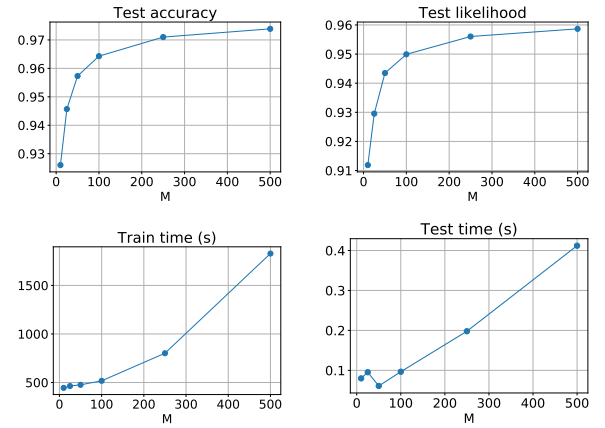


Fig. 7. Influence of  $M$  (number of inducing points) on the test performance (accuracy and likelihood) and computational cost (elapsed time at training and testing) for SVGPCR in the MNIST problem. As theoretically expected, more inducing points lead to better test performance at the expense of a higher computational cost.

noise model considered for the annotators). However, they struggled when dealing with complex non-linear data sets. Then, Gaussian Processes became the preferred choice, since their non-parametric form and accurate uncertainty quantification yielded much better results, e.g. *Rodr14* [21] and *VGPCR* [22] (they differ in the inference procedure used, Expectation Propagation [53] and Variational Inference [32], respectively). However, the poor scalability of GPs hampered the wide adoption of these approaches in practice. This motivated the development of the so-called RFF and VFF algorithms, which leverage Random Fourier features approximations to GPs to propose two more scalable GP-

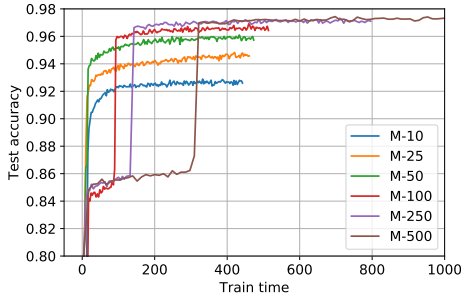


Fig. 8. Evolution of test accuracy as a function of the training time for different values of  $M$  in the MNIST problem. If there is no limit on the available training time, then high values of  $M$  must be selected (as long as it allows for the kernel matrix inversion). However, lower values would be more appropriate for a fast training, since the amount of parameters to be trained significantly reduces. Moreover, when a certain (problem-dependent)  $M$  has been reached, there is no a significant benefit by increasing it (observe the difference from  $M = 250$  to  $M = 500$ ).

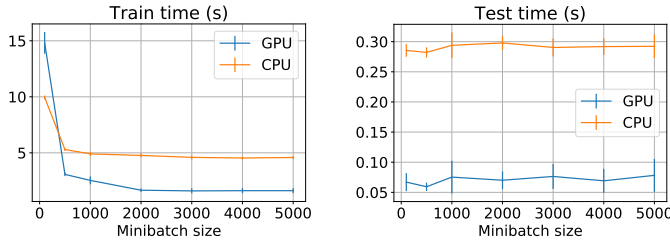


Fig. 9. Comparison between CPU and GPU implementations of SVGPCR in terms of computational cost for the MNIST problem. At training, the time depends on the minibatch size, since a greater minibatch implies computations with larger matrices and less memory copies, which benefits the GPU. With a minibatch of size 500 (the default used in this section), the GPU is around two times faster. These values are per epoch. At testing (production time), the GPU is over three times faster, and logically does not depend on the minibatch size. In this case, the shown values are for the whole test set.

based crowdsourcing methods [30]. These approaches significantly improve the scalability, reducing it from cubic  $\mathcal{O}(N^3)$  to linear  $\mathcal{O}(ND^2)$  (with  $D$  the number of Fourier frequencies used,  $D \ll N$ , see [30]). In practice, this implies moving from manageable data sets of  $N \approx 10^4$  up to  $N \approx 10^5/10^6$ . However, RFF and VFF do not factorize in mini-batches, which prevents them from reaching data sets of virtually any size.

In the last few years, these classical (mainly GP-based) approaches have been replaced by crowdsourcing methods based on Deep Learning (DL) [5], [23]. These achieve excellent scalability through mini-batches, and can handle data sets of almost any size. Because of this, they have become the state of the art approach for real-world crowdsourcing problems. In the next section 4.3, we will bring GP-based methods back to a state of the art level. We will show that SVGPCR is competitive with DL-based methods, and additionally provides a very accurate control of uncertainty. But before this, it is worth to analyze here the advances that SVGPCR introduces over its predecessors classical (mainly GP-based) crowdsourcing approaches.

More specifically, let us compare SVGPCR with the aforementioned *Raykar*, *Yan* (based on logistic-regression), *Rodr14*, VGPCR (based on GPs), and RFF, VFF (based on

scalable approximations to GP). Since most of them were formulated for binary problems, we consider a binary task relevant to astrophysicists in GravitySpy. Using the data set presented in section 3, the goal is to distinguish between the glitch called “Other” and the rest of types. This is important in order to identify potential overlaps between that catch-all class and the rest of glitches. Moreover, it introduces an imbalanced scenario, since “Other” represents only a 10.12% of the total amount of annotations. We will use the area under the ROC curve (AUC) as test performance metric.

Figure 10 compares the scalability of the compared methods as the training set grows. SVGPCR clearly stands out as the most scalable approach. This can be attributed to its training scheme through mini-batches, which considerably alleviates the dependence on the training set size. The rest of methods explode at different moments: the heavy EP inference of *Rodr14* only allows for training with up to  $N = 2500$ , the GP-based formulation of VGPCR and the complex annotators noise model of *Yan* make them reach  $N = 25000$  with difficulties. In spite of the GP approximation, VFF does not go beyond  $N = 10^5$  in this problem, because of the expensive optimization of Fourier features. Finally, *Raykar* (which is based on cheap logistic regression) and RFF (which does not optimize over the Fourier features) can cope with the full data set, although they are significantly slower than SVGPCR.

Moreover, figure 11 shows that their test performance is pretty far from that of SVGPCR. Indeed, the logistic regression model underlying *Raykar* is not sufficient for the non-linear problem at hand, and the GP approximation provided by RFF is known to be poor when the dimensionality of the problem is high [30] (like here, where we are working with 256 features, recall section 3). The rest of methods are also clearly outperformed, since their limited scalability prevents them from processing the full data set. Interestingly, figure 11 shows an intuitive and logical structure: the more simple logistic-regression based methods are located on the left (less test AUC), the classical GP-based ones in the central part, and SVGPCR on the right.

### 4.3 Comparison with state of the art DL-based methods

In the last years, Deep Learning has emerged as a scalable alternative to model crowdsourcing problems. Two of the most popular approaches are AggNet [5] and the various crowd layers proposed in [23]. The former considers a deep neural network (DNN) as underlying classifier, and a probabilistic noise model for annotators based on per-user confusion matrices. Then, the training step follows an iterative expectation-maximization (EM) scheme between both parts of the model [43, Section 9.4]. Alternatively, the crowd layers in [23] allow for end-to-end training of the DNN, without the need for the EM scheme. This is significantly cheaper in terms of computational cost, although the probabilistic formulation of AggNet allows for a better uncertainty quantification. The three crowd layers studied in [23] will be considered here: CL-VW, CL-VWB and CL-MW. They differ in the parametric form of the annotator noise model, which is increasingly complex: a vector of per-class weights for CL-VW, an additional bias for CL-VWB, and a whole confusion matrix for CL-MW.

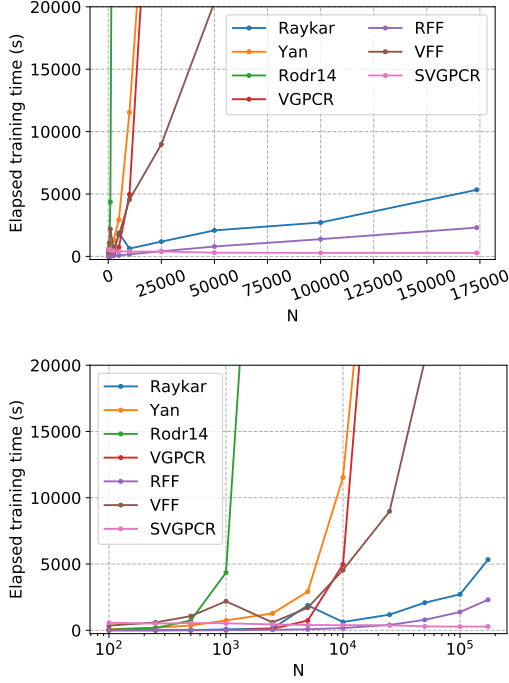


Fig. 10. Elapsed training time as a function of the training set size  $N$  in the binary LIGO data set. The mean over five independent runs is shown. Above, a standard linear scale is used for the x-axis. Notice that SVGPCR exhibits a significantly better scalability than classical probabilistic methods, which is due to its factorization in mini-batches. Moreover, among previous approaches, we can distinguish between those that have already exploded for  $N = 25000$  (*Yan*, *Rodr14*, *VGPCR*, *VFF*), and those which have not yet for the full set size (*Raykar*, *RFF*). In order to better appreciate the differences, a logarithmic scale is used for the x-axis in the figure below. This further shows that *Rodr14* shoots up as early as  $N = 1000$ , *Yan*, *VGPCR* and *VFF* do it around  $N = 10000$ , and *Raykar* and *RFF* are starting beyond  $N = 10^5$ .

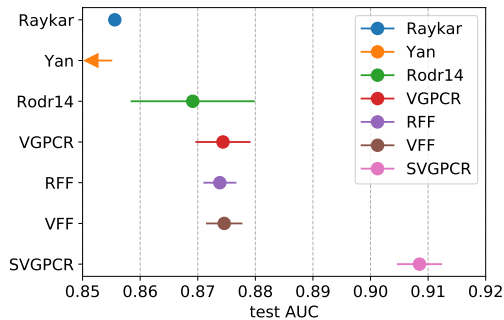


Fig. 11. Test AUC achieved in the binary LIGO data set by the compared models (in each case, the largest one that could be trained). The mean and one standard deviations are shown. SVGPCR clearly outperforms methods which cannot cope with the full data set (*Yan*, *Rodr14*, *VGPCR*, *VFF*). Moreover, *RFF* and *Raykar* are also beaten because of their more limited formulation (see the main text).

These four DL-based methods (AggNet, CL-VW, CL-VWB, CL-MW) are compared against three increasingly complex SVGPCR models: SVGPCR-10, SVGPCR-50, SVGPCR-100, where each number represents the amount of inducing points used. As all these approaches are defined for multi-class tasks, the full LIGO problem in section 3 can be addressed now.

Tables 4 and 5 show the global and per-class test performance of the compared methods. Table 4 is devoted to the test accuracy, which relies only on the mode of the predictive distribution and is less influenced by the uncertainty quantification of the model. Table 5 shows the test likelihood, which additionally depends on the uncertainty of the predictive distribution, and therefore depends more heavily on its accurate control within the model.

In both tables, SVGPCR stands out as the best-performing method globally. The difference is greater in the case of the test likelihood, which is logically explained by the better uncertainty quantification of GPs. Indeed, the better control of uncertainty also justifies that AggNet outperforms CL-based methods in test likelihood (whereas they are very similar in accuracy). Moreover, observe that the global superiority of SVGPCR is not due to a great result in only one or two very populated classes. SVGPCR performs consistently well across the 15 glitch types in both tables, winning in few of them (a bit more in test likelihood, as logically expected), and avoiding dramatic failures on difficult classes. This will be also observed for the alternative GP-based methods introduced in Section 5. According to astrophysicists at GravitySpy, this regularity across classes is a desirable property for a reliable glitch detection system.

It is also worth to notice that  $M = 50$  inducing points seem enough for the problem at hand. In both tables 4 and 5, a significant improvement is observed from  $M = 10$  to  $M = 50$ , but  $M = 100$  produces very similar results. This small value of  $M = 50$  hints at a not very complex internal structure of the data. It is also interesting to observe that, in general, the most difficult classes are “Repeating Blips” and “Other” (recall the 15 types in figure 3). This discovery is not surprising for astrophysicists in GravitySpy, since the former is usually confused with “Blips”, and the latter is a catch-all class to which some conservative annotators resort too often. The case of “Other” is also related to the interest of astrophysicists to study it separately in the experiment of previous section 4.2.

It is also important to highlight that all these methods are scalable enough so as to cope with the full LIGO data set. More specifically, figure 12 shows the elapsed time at training and testing for the compared methods. In general, the proposed SVGPCR is competitive with DL-based methods in these aspects. At training, SVGPCR is significantly faster than AggNet due to the heavy iterative EM scheme of the latter, and is slower than CL-MW<sup>6</sup>. Nonetheless, less than one hour of training is a competitive result for a data set with 173565 instances (recall section 3). At testing, SVGPCR is the fastest approach, which is convenient for real-time applications the system might be used for.

6. Results of CL-VW/CL-VWB being worse than CL-MW in figure 12 might be attributed to implementation inefficiency, since the former include for loops whereas matrix multiplication is used in the latter.



TABLE 4

Per-class and global test accuracy for the compared methods in the LIGO experiment. Mean and standard deviation over ten runs are shown. Globally, SVGPCR with enough inducing points outperforms DL-based methods by 2%. In per-class results, notice the regularity of SVGPCR, which performs well across all classes without standing out in many of them.

	AggNet	CL-VW	CL-VWB	CL-MW	SVGPCR (M=10)	SVGPCR (M=50)	SVGPCR (M=100)
1080LINE	<b>.9791(.0045)</b>	.9676(.0063)	.9732(.0059)	.9746(.0063)	.9781(.0075)	.9720(.0076)	.9720(.0069)
1400RIPPLE	<b>.9447(.0237)</b>	.1666(.3335)	.3243(.4001)	.0000(.0000)	.0000(.0000)	.8528(.0203)	.8577(.0171)
BLIP	.9737(.0078)	.9754(.0079)	.9741(.0135)	<b>.9826(.0062)</b>	.8712(.2904)	.9643(.0044)	.9622(.0052)
EXTR.LOUD	<b>.8704(.0566)</b>	.7193(.0652)	.8034(.0660)	.5272(.3502)	.3681(.3705)	.7261(.0398)	.7295(.0427)
KOIFISH	.8760(.0634)	.8749(.0151)	<b>.9043(.0296)</b>	.9023(.0203)	.8007(.2673)	.8844(.0129)	.8828(.0115)
L.F.BURST	.7649(.0802)	.8639(.0299)	.9023(.0211)	<b>.9146(.0174)</b>	.7912(.2638)	.8864(.0087)	.8861(.0105)
L.F.LINE	.8603(.0742)	.8603(.0274)	.8514(.0293)	.8566(.0279)	.8354(.2785)	.9150(.0105)	<b>.9156(.0111)</b>
NOGLITCH	<b>.9174(.0354)</b>	.8835(.0201)	.8887(.0241)	.9141(.0274)	.7048(.2366)	.7941(.0223)	.7951(.0162)
OTHER	.2786(.0294)	<b>.4167(.0348)</b>	.3660(.0376)	.3503(.0297)	.3622(.1213)	.3977(.0083)	.4011(.0091)
P.L.60HZ	.8885(.0676)	.8591(.0290)	<b>.9468(.0104)</b>	.9365(.0179)	.7417(.2476)	.8438(.0102)	.8425(.0127)
REP.BLIPS	.5487(.0730)	.0470(.1410)	.0572(.1717)	.0000(.0000)	.0000(.0000)	.6649(.0215)	<b>.6700(.0210)</b>
SCATT.LIGHT	.9088(.0492)	.9497(.0126)	.9601(.0109)	<b>.9645(.0053)</b>	.8657(.2886)	.9600(.0050)	.9562(.0056)
SCRATCHY	.8980(.0400)	.4433(.4437)	.4426(.4452)	.4440(.4444)	.8093(.2702)	.8953(.0206)	<b>.9000(.0165)</b>
VIOLIN	.9755(.0057)	<b>.9932(.0032)</b>	.9921(.0033)	.9930(.0018)	.8915(.2971)	.9899(.0027)	.9914(.0017)
WHISTLE	.9175(.0236)	.9359(.0136)	<b>.9438(.0137)</b>	.9377(.0230)	.8122(.2712)	.9166(.0070)	.9201(.0047)
GLOBAL	.8957(.0227)	.8886(.0126)	.8985(.0105)	.8956(.0104)	.8355(.1919)	<b>.9184(.0031)</b>	.9183(.0027)

TABLE 5

Per-class and global test likelihood for the compared methods in the LIGO experiment. Mean and standard deviation over ten independent runs are shown. Globally, SVGPCR with enough inducing points outperforms DL-based methods by almost 3%. It also exhibits a desirable regularity across the different classes. Moreover, notice that, compared to the accuracy in Table 4, there exists here a greater advantage against methods that do not quantify uncertainty (i.e. CL-based ones).

	AggNet	CL-VW	CL-VWB	CL-MW	SVGPCR (M=10)	SVGPCR (M=50)	SVGPCR (M=100)
1080LINE	<b>.9781(.0048)</b>	.9515(.0091)	.9597(.0076)	.9649(.0074)	.8811(.2715)	.9689(.0082)	.9688(.0075)
1400RIPPLE	<b>.9416(.0242)</b>	.1644(.3290)	.3196(.3937)	.0118(.4874)	.0067(.0199)	.8475(.0182)	.8509(.0156)
BLIP	.9709(.0077)	.9753(.0079)	.9735(.0123)	<b>.9777(.0043)</b>	.8746(.2694)	.9606(.0044)	.9587(.0055)
EXTR.LOUD	<b>.8626(.0574)</b>	.7214(.0645)	.8024(.0684)	.5276(.3503)	.3651(.3541)	.7266(.0435)	.7242(.0408)
KOIFISH	.8688(.0679)	.8752(.0161)	<b>.9013(.0300)</b>	.8959(.0190)	.8022(.2457)	.8788(.0109)	.8784(.0117)
L.F.BURST	.7575(.0799)	.7249(.0139)	.8035(.0177)	.8419(.0207)	.7966(.2434)	<b>.8851(.0083)</b>	.8838(.0098)
L.F.LINE	.8502(.0786)	.5808(.0192)	.6200(.0178)	.6917(.0242)	.8378(.2571)	<b>.9125(.0094)</b>	.9118(.0103)
NOGLITCH	<b>.9091(.0402)</b>	.8109(.0247)	.8196(.0342)	.8196(.0244)	.7062(.2146)	.7919(.0238)	.7932(.0146)
OTHER	.2692(.0285)	<b>.4123(.0322)</b>	.3439(.0362)	.3268(.0275)	.3686(.1012)	.3959(.0095)	.3999(.0091)
P.L.60HZ	.8700(.0665)	.8565(.0291)	<b>.8993(.0108)</b>	.8929(.0117)	.7435(.2260)	.8384(.0107)	.8380(.0107)
REP.BLIPS	.5323(.0664)	.0466(.1400)	.0584(.1752)	.4878(.8447)	.0067(.0199)	.6581(.0191)	<b>.6651(.0198)</b>
SCATT.LIGHT	.8716(.0480)	.8782(.0165)	.8947(.0196)	.9416(.0067)	.8683(.2673)	<b>.9558(.0047)</b>	.9520(.0057)
SCRATCHY	.8823(.0399)	.4425(.4429)	.4411(.4440)	.4414(.4419)	.7971(.2441)	.8911(.0215)	<b>.8953(.0176)</b>
VIOLIN	.9738(.0058)	.9823(.0034)	.9815(.0024)	.9863(.0022)	.8960(.2765)	.9875(.0020)	<b>.9886(.0014)</b>
WHISTLE	.9167(.0215)	.9341(.0149)	<b>.9427(.0141)</b>	.9388(.0213)	.8124(.2490)	.9155(.0064)	.9179(.0046)
GLOBAL	.8871(.0240)	.8175(.0106)	.8387(.0113)	.8528(.0101)	.8126(.2487)	<b>.9154(.0033)</b>	.9149(.0027)

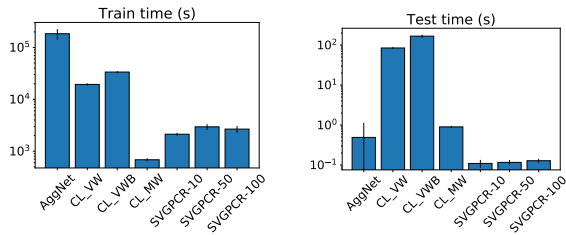


Fig. 12. Train and test times (mean and standard deviation over ten independent runs) for the compared methods in the LIGO experiment. Notice the logarithmic scale in y-axis. The proposed SVGPCR is competitive with all the DL-based methods in terms of computational cost at both training and testing.

As already pointed out in Table 5, the underlying GP model of SVGPCR implies an advantage over DL-based methods in terms of uncertainty quantification. The test likelihood metric is a global measure of the quality of the predictive distribution obtained for each individual test instance. To clearly understand the benefits of the GP modelling, figure 13 shows the predictive distributions for

some test instances which are behind the better global performance of SVGPCR. Only the best method (in terms of test likelihood) of each type (i.e. CL-based ones, SVGPCR ones, and AggNet) is considered, which yields the three columns in figure 13. Each row represents a different test instance.

Interestingly, we observe that the three approaches correctly classify the four instances, that is, they assign the highest probability to the correct class (which is highlighted in red). In particular, this means that these four instances contribute equally to the test accuracy of the three methods. However, notice that the quality of the predictive distribution worsens from left to right (i.e., from better to worse uncertainty quantification theoretical properties), since the methods become less certain about the correct answer and assign more probability to wrong ones. This is precisely what is accounted for in the test likelihood metric. From a practical perspective, this better quality of the predictive distributions has been particularly appreciated by astrophysicists at GravitySpy, in addition to the improvement in test accuracy (recall table 4). We stress that test instances

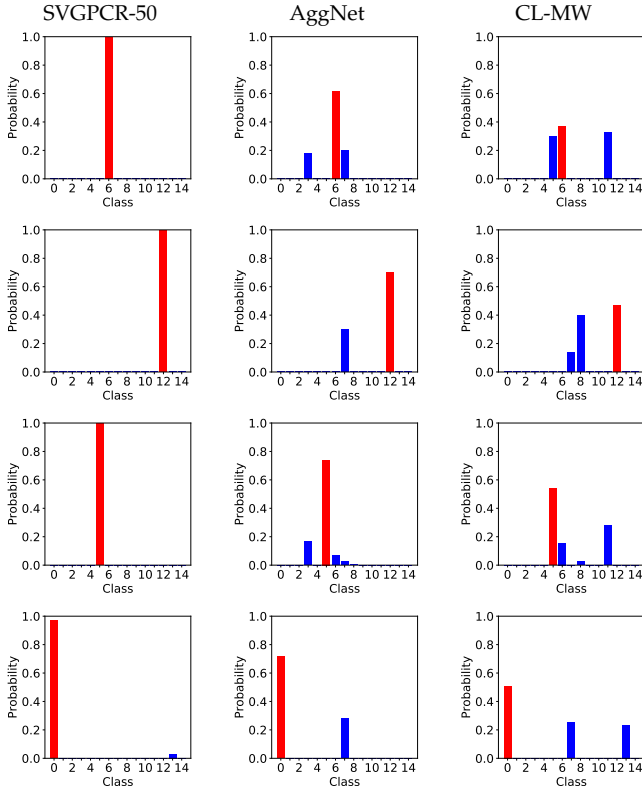


Fig. 13. Test predictive distribution obtained by the compared methods for four different test instances in the LIGO experiment. All the methods assign the highest probability to the correct class (which is highlighted in red). However, from left to right, the quality of the predictive distribution decreases, as greater probability is assigned to wrong classes. This is related to the uncertainty quantification capabilities of each method, an aspect at which the GP modelling of SVGPCR stands out. Moreover, these differences in the predictive distributions are behind the superiority of SVGPCR in test likelihood (table 5).

in figure 13 are among those that most contribute to the outperformance in test likelihood. In general, when it is not certain about the outcome, SVGPCR provides uncertainty for its estimations. This is illustrated in the third section of the supplementary material. The figure there is analogous to this one, but using randomly selected test points.

Finally, a key aspect of crowdsourcing methods is the identification of the different annotators behavior. Unlike in section 4.1, where we had simulated annotators to check the good estimations of SVGPCR, in this real experiment we do not have available a ground-truth. Nonetheless, let us compare the predictions obtained by the different methods. We will see that they capture similar patterns, some of which can be explained from the experience of astrophysicists. Figure 14 shows the confusion matrices predicted by the compared methods for five different annotators. In the CL-based family we only consider CL-MW, as it is the best in test likelihood and the only one which provides a confusion matrix.

One of the most distinctive features for all instances and methods is the predominance of high values in the diagonal. This was considered as a positive feedback by astrophysicists, as it means that annotators have been generally well instructed to distinguish among glitches. Additionally, other patterns out of the diagonal are worth an analysis. For the

first column (first volunteer), SVGPCR and AggNet detect that glitches of type 1 (i.e. “1400Ripple”, recall figure 3) are classified as class 13 (“Violin Mode Harmonic”). This is a very frequent mistake according to experts, since the general appearance of both glitches is similar. We also observe that CL-MW does not agree on this prediction. This discrepancy of CL-MW for some particular patterns is recurrent across different annotators, and can be attributed to the different modelling of the annotators noise (non-probabilistic one, but through weights in the DNN). The second column shows a typical conservative annotator, who resorts too frequently to the catch-all “Other” class. This is reflected in the persistent high values of the row number 8 in the matrices, regardless of the column (the real class). For the third column, the three methods identify the confusion from “Violin Mode Harmonic” to “1400Ripple”. Notice that this is the opposite to the first annotator, where the confusion was the other way round. In the fourth annotator, SVGPCR detects the same issue with “Violin Mode Harmonic” and “1400Ripple”, whereas the others are less certain about this. Moreover, AggNet exhibits a noisy behavior compared to SVGPCR and CL-MW. Although perhaps less explicitly, this can be also observed across different annotators, and might be due to the iterative nature of AggNet, which does not allow for an end-to-end learning and leaves some extra noise after training. In the fifth annotator, the three methods identify a very common confusion, which is labelling instances whose real class is “Blip” as “Koifish” (classes 2 and 4, respectively). Although these glitches seem pretty different in the paradigmatic examples shown in figure 3, wider “Blip” and narrower “Koifish” are frequent in the data set, and might mislead a non-expert volunteer.

Most importantly, the identification of all these wrong behaviors allows crowdsourcing methods to take full advantage of the noisy annotations. It is also worth noticing that the Bayesian nature of SVGPCR provides uncertainties for the confusion matrices obtained here (recall the full posterior Dirichlet distributions in eq. (14)), which is not available for the DL-based methods.

## 5 EXPLORING RECENT INFERENCE TECHNIQUES

Once SVGPCR has been successfully developed, this section explores modern GP inference approaches beyond the standard SVGP. As motivated in the introduction, Section 5.1 uses Normalizing Flows [39] to represent more complex posterior distributions for SVGPCR. Then, Section 5.2 adapts the GP-Net model [37] to crowdsourcing.

### 5.1 Normalizing Flows

Normalizing Flows (NF), originally introduced in [39], see also the recent review [40], have become a very popular technique to represent complex distributions that can be used as approximate posteriors in VI. The idea is to transform a simple base distribution (typically a Gaussian) through a sequence of invertible transformations. Sampling from such a distribution is straightforward, and its density depends on the determinant of the transformations Jacobian, which must be cheap to compute. Different transformations yield different NFs, such as planar and radial [39],

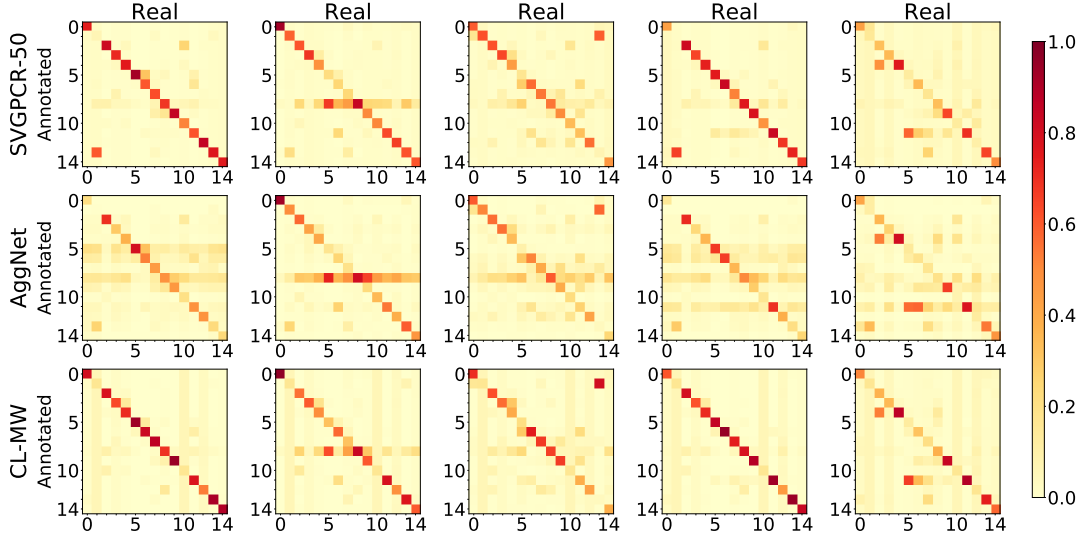


Fig. 14. Confusion matrices estimated by the three families of algorithms (rows) and for five different annotators (columns) in the LIGO experiment. In each matrix, the real class is represented in the x-axis, and the annotated one in the y-axis (that is, the matrices add up to one by columns). For every annotator, the general structure of the three estimated matrices is similar. In particular, the highest values are located in the diagonal, which means that annotators have been correctly instructed in general. Moreover, recall that SVGPCR provides a full probability distribution (in particular uncertainties) for these predictions.

TABLE 6

Per-class and global test accuracy for recent inference methods on GPs. Mean and standard deviation over ten runs are shown. The best performing SVGPCR method is shown for comparison. Both NF and GP-Net slightly outperform SVGPCR. Moreover, unlike inducing points based ones, GP-Net achieves very competitive results for  $M = 10$ , as theoretically expected.

	SVGPCR (M=50)	SVGPCR-NF (M=10)	SVGPCR-NF (M=50)	SVGPCR-NF (M=100)	GPNETCR (M=10)	GPNETCR (M=50)	GPNETCR (M=100)
1080LINE	.9720(.0076)	<b>.9785(.0128)</b>	.9725(.0064)	.9729(.0078)	.9740(.0045)	.9732(.0037)	.9727(.0042)
1400RIPPLE	<b>.8528(.0203)</b>	.4065(.4067)	.8447(.0220)	.8496(.0300)	.8398(.0247)	.8195(.0272)	.8398(.0499)
BLIP	.9643(.0044)	.7742(.3872)	.9622(.0033)	.9626(.0060)	.9665(.0089)	<b>.9679(.0038)</b>	.9645(.0066)
EXTR.LOUD	.7261(.0398)	.4045(.3338)	.7114(.0477)	.7193(.0627)	.7455(.0629)	.7420(.0653)	<b>.7523(.0794)</b>
KOIFISH	.8844(.0129)	.7116(.3558)	.8837(.0110)	.8888(.0129)	.8825(.0141)	.8845(.0206)	<b>.8912(.0130)</b>
L.FBURST	.8864(.0087)	.7062(.3532)	.8903(.0081)	.8840(.0078)	.8896(.0162)	<b>.8950(.0136)</b>	.8864(.0108)
L.FLINE	.9150(.0105)	.7380(.3691)	.9158(.0094)	<b>.9266(.0072)</b>	.9096(.0184)	.9077(.0178)	.9116(.0105)
NOGLITCH	.7941(.0223)	.6239(.3128)	<b>.7958(.0160)</b>	.7926(.0195)	.7739(.0285)	.7797(.0136)	.7800(.0240)
OTHER	.3977(.0083)	.3282(.1644)	.4034(.0102)	.3962(.0119)	.4061(.0253)	.4011(.0157)	<b>.4069(.0152)</b>
P.L.60HZ	.8438(.0102)	.6574(.3290)	.8472(.0098)	.8430(.0120)	.8574(.0096)	<b>.8634(.0055)</b>	.8579(.0127)
REP.BLIP	<b>.6649(.0215)</b>	.0000(.0000)	.6470(.0331)	.6427(.0528)	.5812(.0608)	.6043(.0465)	.6068(.0606)
SCATT.LIGHT	.9600(.0050)	.7707(.3854)	.9598(.0049)	<b>.9637(.0049)</b>	.9630(.0088)	.9606(.0067)	.9622(.0068)
SCRATCHY	.8953(.0206)	.7193(.3604)	<b>.9027(.0191)</b>	.8880(.0183)	<b>.9027(.0191)</b>	.8880(.0183)	.8947(.0217)
VIOLIN	.9899(.0027)	.7919(.3959)	.9904(.0019)	.9898(.0024)	.9924(.0028)	<b>.9935(.0012)</b>	.9921(.0020)
WHISTLE	.9166(.0070)	.7281(.3642)	.9237(.0068)	.9202(.0083)	<b>.9281(.0156)</b>	.9272(.0096)	.9254(.0106)
GLOBAL	.9184(.0031)	.7765(.2584)	.9189(.0021)	<b>.9197(.0027)</b>	.9185(.0021)	.9186(.0018)	.9184(.0021)

inverse autoregressive flow [54], or masked autoregressive flow [55]. Also, augmented normalizing flows, which overcome the constraint on intermediate dimensionality, have been recently explored in [56].

In our model of SVGPCR, we used an approximate Gaussian posterior  $q(\mathbf{U})$  for the inducing points, recall eq. (13). This yields a tractable KL term in the ELBO between two Gaussians (prior and posterior), recall eq. (15). However, the true posterior might not be Gaussian, since we are using a non-conjugate likelihood for  $\mathbf{z}|\mathbf{F}$  (in addition to the crowdsourcing likelihood for  $\mathbf{Y}|\mathbf{z}$ ).

In this section, we consider an approximate posterior for  $\mathbf{U}$  based on a planar NF with Leaky ReLU (LReLU) nonlinearities. More specifically, each  $\mathbf{u}_k$  is defined through  $\mathbf{u}_k^0 \rightarrow \dots \rightarrow \mathbf{u}_k^L = \mathbf{u}_k$ , where  $\mathbf{u}_k^0 \sim \mathcal{N}(\mathbf{m}_k^0, \mathbf{S}_k^0)$  and  $\mathbf{u}_k^l = f_k^l(\mathbf{u}_k^{l-1})$  with  $f_k^l(\mathbf{a}) = \mathbf{a} + \mathbf{v} \cdot \text{LReLU}(\mathbf{w}^\top \mathbf{a} + b)$  (logically, each  $\mathbf{v}$ ,  $\mathbf{w}$  and  $b$  depend on  $k$  and  $l$ , but this

is omitted to lighten the notation). As explained in [39, Section 4.1], the determinant of the Jacobian matrix for this transformation,  $\partial f_k^l / \partial \mathbf{a}$ , can be computed in  $\mathcal{O}(M)$ , where  $M$  is the dimensionality of  $\mathbf{u}_k^l$ , i.e. the number of inducing points. Therefore, the new variational parameters are  $\mathbf{V} = \{\mathbf{m}_k^0, \mathbf{S}_k^0, \mathbf{v}_k^l, \mathbf{w}_k^l, b_k^l\}_{k,l}$ . This approach will be called SVGPCR-NF.

Following the approach in [39], the new KL term in the ELBO is as follows:

$$\text{KL}(q(\mathbf{u}_k)||p(\mathbf{u}_k)) = \mathbb{E}_{q(\mathbf{u}_k^0)} \log q(\mathbf{u}_k^0) - \sum_{l=1}^L \mathbb{E}_{q(\mathbf{u}_k^0)} \log \left| \det \frac{\partial f_k^l}{\partial \mathbf{u}_k^{l-1}} \right| - \mathbb{E}_{q(\mathbf{u}_k^0)} \log p(\mathbf{u}_k^L). \quad (17)$$

The first term can be computed in closed form (the entropy of a Gaussian is well-known), the second term can be stochastically approximated in  $\mathcal{O}(LM)$  by sampling from



TABLE 7

Per-class and global test likelihood for recent inference methods on GPs. Mean and standard deviation over ten runs are shown. The best performing SVGPCR method is shown for comparison. Both NF and GP-Net slightly outperform SVGPCR. Moreover, unlike inducing points based ones, GP-Net achieves very competitive results for  $M = 10$ , as theoretically expected.

	SVGPCR (M=50)	SVGPCR-NF (M=10)	SVGPCR-NF (M=50)	SVGPCR-NF (M=100)	GPNETCR (M=10)	GPNETCR (M=50)	GPNETCR (M=100)
1080LINE	.9689(.0082)	.7886(.3611)	.9690(.0066)	.9698(.0077)	<b>.9712(.0043)</b>	.9707(.0037)	.9704(.0043)
1400RIPPLE	<b>.8475(.0182)</b>	.4167(.3909)	.8399(.0227)	.8416(.0275)	.8373(.0233)	.8167(.0261)	.8331(.0490)
BLIP	.9606(.0044)	.7835(.3585)	.9586(.0037)	.9583(.0065)	.9632(.0089)	<b>.9645(.0041)</b>	.9615(.0060)
EXTR.LOUD	.7266(.0435)	.4157(.3160)	.7025(.0475)	.7027(.0529)	.7385(.0659)	.7342(.0611)	<b>.7447(.0740)</b>
KOIFISH	.8788(.0109)	.7215(.3275)	.8789(.0104)	.8814(.0141)	.8806(.0145)	.8813(.0192)	<b>.8839(.0130)</b>
L.FBURST	.8851(.0083)	.7186(.3261)	.8881(.0082)	.8810(.0073)	.8877(.0162)	<b>.8919(.0136)</b>	.8835(.0107)
L.FLINE	.9125(.0094)	.7477(.3407)	.9120(.0093)	<b>.9225(.0079)</b>	.9065(.0182)	.9045(.0181)	.9083(.0104)
NOGLITCH	<b>.7919(.0238)</b>	.6337(.2844)	<b>.7919(.0161)</b>	.7854(.0211)	.7688(.0283)	.7776(.0131)	.7799(.0220)
OTHER	.3959(.0095)	.3376(.1359)	.3989(.0088)	.3921(.0125)	<b>.4056(.0259)</b>	.3999(.0153)	.4041(.0145)
P.L.60HZ	.8384(.0107)	.6680(.3010)	.8414(.0106)	.8384(.0124)	.8546(.0115)	<b>.8595(.0057)</b>	.8542(.0134)
REP.BLIPS	<b>.6581(.0191)</b>	.0134(.0266)	.6406(.0347)	.6341(.0568)	.5834(.0663)	.6031(.0412)	.6043(.0596)
SCATT.LIGHT	.9568(.0047)	.7801(.3568)	.9555(.0046)	.9590(.0054)	<b>.9597(.0080)</b>	.9578(.0061)	.9593(.0066)
SCRATCHY	.8911(.0215)	.7187(.3271)	.8936(.0190)	.8829(.0190)	<b>.8985(.0201)</b>	.8826(.0163)	.8904(.0245)
VIOLIN	.9875(.0020)	.8027(.3681)	.9879(.0018)	.9871(.0025)	.9898(.0026)	<b>.9909(.0011)</b>	.9900(.0018)
WHISTLE	.9155(.0064)	.7362(.3349)	.9209(.0062)	.9156(.0080)	<b>.9297(.0123)</b>	.9253(.0076)	.9224(.0092)
GLOBAL	.9154(.0033)	.7346(.3341)	.9152(.0022)	.9157(.0027)	.9157(.0020)	<b>.9158(.0019)</b>	.9155(.0023)

$q(\mathbf{u}_k^0)$  and evaluating the known Jacobian, recall [39, Section 4.1], and the same applies for the third one. The rest of the ELBO is as in eq. (15), where  $q(\mathbf{f}_{n,:})$  does not have a closed-form expression anymore, but it is approximated by sampling  $S$  values from the flow and averaging the  $S$  conditional Gaussians of  $\mathbf{f}$  given  $\mathbf{u}$ . The same applies for predicting. Notice that the computational complexity is still dominated by the SVGPCR operations.

We apply SVGPCR-NF on the full LIGO dataset with  $M = 10, 50, 100$ , as was done for the standard SVGPCR. To better understand the dependence on the flow length  $L$ , we assess the performance when moving from  $L = 2$  to  $L = 11$  (when the performance begins to decrease). This is shown in figure 15, which reveals that  $L = 9$  is the optimal choice in terms of both accuracy and test likelihood.

Using this value of  $L$ , the full results in terms of test accuracy and likelihood are shown in Tables 6 and 7, respectively. The performance is very similar to SVGPCR, exhibiting a slight improvement. This suggests that, in this particular case, the true posterior distribution might be unimodal and a Gaussian could be enough to represent it (indeed, Fig. 2 in the supplementary material shows the distribution obtained for  $q(\mathbf{U})$ ). Interesting properties of SVGPCR, such as the regularity across classes and avoiding dramatic failures in difficult ones, are also maintained. Moreover, just like for SVGPCR, results are clearly improved when growing from  $M = 10$  to  $M = 50$ , but stay similar for  $M = 100$ . This is to be expected, since the underlying SVGP model is still the same, and it is well-known that few inducing points limit the expressiveness of the predictive distribution. Next we explore a recent model that adopts a different approach.

## 5.2 Inference Networks for GPs

In this section we adapt GP-Net [37] to the crowdsourcing scenario. Since the expressiveness of inducing points based sparse GP approaches is limited by the amount of such points, GP-Net explores the use of inference networks to approximate the posterior distribution of GPs. Inference is performed in the function space directly, see also [57], and

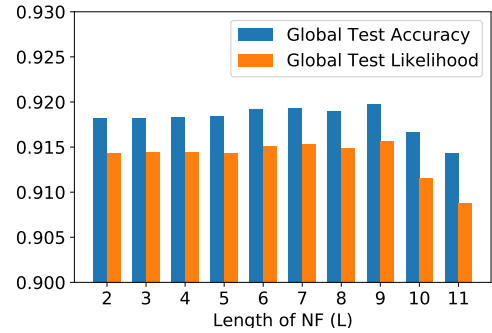


Fig. 15. Global test accuracy and global test likelihood for SVGPCR-NF with different lengths for the flow ( $L$ ). The amount of inducing points is  $M = 100$ . The results are the mean over ten runs.

the posterior stochastic process  $q(f)$  is modelled with a parametric inference network  $q_\gamma(f)$ . The joint distribution must be Gaussian for a finite set of points, see [37, Section 4] for different choices, such as the random feature expansions that will be used here. To learn the parameters  $\gamma$ , a stochastic functional mirror-descent algorithm is tracked, which iteratively adapts the true stochastic posterior [37, Section 3.1]. In each step, the inference network is matched to such posterior on a measurement set  $\mathbf{X}_M$  of size  $M$ , which allows for learning meaningful correlations [37, Section 3.2]. Importantly, notice that this  $M$  does not limit the expressiveness of the inference network  $q_\gamma$ .

In the crowdsourcing scenario, recall that eq. (7) defines the full probabilistic model before inducing points are introduced. Now, we consider an approximate posterior  $q(\mathbf{F}, \mathbf{Z}, \mathbf{R}) = q(\mathbf{F})q(\mathbf{Z})q(\mathbf{R})$  where  $q(\mathbf{F})$  is given by  $K$  inference networks  $q_{\gamma_1}(f_1), \dots, q_{\gamma_K}(f_K)$ , one for each class. We use random feature expansions with 100 hidden units, recall [37, Section 4]. Then, VI can be applied to minimize  $\text{KL}(q(\mathbf{F}, \mathbf{Z}, \mathbf{R}) || p(\mathbf{F}, \mathbf{Z}, \mathbf{R} | \mathbf{Y}, \Theta))$  in turns. Fixing  $q(\mathbf{F})$ , and following the derivation of eq. (15), a gradient step must be given on  $\{q_{n_k}\}$  and  $\{\tilde{\alpha}_{ij}^a\}$  (the variational parameters of

$q(\mathbf{Z})$  and  $q(\mathbf{R})$  respectively) to maximize

$$\begin{aligned} \mathcal{F}(\{q_{nk}\}, \{\tilde{a}_{ij}^a\}) = & \sum_{n=1}^N \sum_{k=1}^K q_{nk} \mathbb{E}_{q_\gamma(\mathbf{f}_{n,:})} [\log p(\mathbf{e}_k | \mathbf{f}_{n,:})] + \\ & + \sum_{n=1}^N \sum_{a \in A_n} \sum_{\mathbf{y} \in \mathbf{Y}_n^a} \sum_{k=1}^K q_{nk} \mathbb{E}_{q(\mathbf{r}_k^a)} [\log p(\mathbf{y} | \mathbf{e}_k, \mathbf{r}_k^a)] - \\ & - \sum_{n=1}^N \sum_{k=1}^K q_{nk} \log q_{nk} - \sum_{a=1}^A \sum_{k=1}^K \text{KL}(q(\mathbf{r}_k^a) || p(\mathbf{r}_k^a)). \end{aligned} \quad (18)$$

This is analogous to eq. (15), but the inducing points term is not present anymore, and  $q_\gamma(\mathbf{f}_{n,:})$  is given by the inference networks (fixed in this step;  $\gamma$  denotes  $\gamma_1, \dots, \gamma_K$  jointly). Then, fixing  $q(\mathbf{U})$  and  $q(\mathbf{R})$ , and following the derivations in [37, Section 3.2] (non-conjugate likelihood case), the inference networks parameters  $\gamma$  must be updated by maximizing

$$\begin{aligned} \mathcal{L}(\gamma) = & \mathbb{E}_{q_\gamma(\mathbf{F}_M, \mathbf{F})} \left[ \beta_t \sum_{n=1}^N \sum_{k=1}^K q_{nk} \log p(\mathbf{e}_k | \mathbf{f}_{n,:}) + \right. \\ & \left. + \beta_t \log p(\mathbf{F}_M, \mathbf{F}) + (1 - \beta_t) \log q_{\gamma_t}(\mathbf{F}_M, \mathbf{F}) - \log q_\gamma(\mathbf{F}_M, \mathbf{F}) \right]. \end{aligned}$$

Here,  $q_{\gamma_t}$  refers to the previous value of the inference network,  $\mathbf{F}_M$  denotes the evaluation of the  $K$  inference networks on the measurement set  $\mathbf{X}_M$  (which is randomly sampled from the training set as in [37]), and  $\beta_t$  is the learning rate of the functional mirror-descent algorithm (which is set to  $\beta_t = \beta_0(1 + \xi\sqrt{t})^{-1}$  as in [37]). In practice, the first term is approximated with a mini-batch, as in eqs. (15) and (18). The GP hyperparameters  $\Theta$  are optimized as in [37], by minimizing the KL divergence between  $q_{\gamma_t}$  and the GP prior on the current mini-batch. This method is called GPNETCR.

As with SVGPCR and SVGPCR-NF, we apply GPNETCR on the full LIGO dataset with  $M = 10, 50, 100$  (here  $M$  is the measurement set size). The results in terms of test accuracy and likelihood are shown in Table 6 and 7, respectively. The performance for  $M = 50, 100$  is very similar to SVGPCR-NF, i.e. just slightly better than SVGPCR. This confirms the robustness and convenience of the GP-based crowdsourcing formulations, which show good results across classes and avoid noticeable failures in difficult ones. However, the results for  $M = 10$  are very different. Whereas 10 inducing points do not yield a sufficiently expressive posterior in SVGPCR(-NF), the GPNETCR inference network capacity is not constrained by  $M$ . In fact, the GPNETCR with  $M = 10$  performs better than SVGPCR for any amount of  $M$ . This confirms that the properties of GP-Net can be extended to crowdsourcing.

## 6 CONCLUSIONS AND FUTURE WORK

In this work we have first introduced SVGPCR, an extension of SVGP to crowdsourcing that can scale up to very large data sets through its mini-batch training scheme. The motivation is the problem of glitch classification in the laureate LIGO project, which is addressed with crowdsourcing techniques in the GravitySpy sub-project. To that end, and in order to obtain accurate predictive distributions, astrophysicists were interested in combining the excellent uncertainty

quantification of GP-based crowdsourcing methods with the scalability of those based on deep learning (DL). The proposed SVGPCR brings back GP-based methods to the state of the art in crowdsourcing.

SVGPCR is competitive with DL-based approaches in terms of test accuracy and computational cost, and stands out in terms of predictive distribution quality. Moreover, its behavior naturally follows its theoretical formulation: it provides very accurate estimations for the annotators expertise degree, and the inducing points influence the test performance and the computational cost as expected. We further leveraged recent inference techniques to propose SVGPCR-NF and GPNETCR.

In the LIGO problem, the glitches were given by relevant features extracted by astrophysicists. However, in the case of more complex data such as images, audio or natural language, DL-based methods can benefit from convolutional layers in the deep neural network. From a probabilistic perspective, this could be addressed through Deep Gaussian Processes [58] and the very recent attempts to introduce convolutional structure in GPs [59], [60]. Moreover, recent successful models combining the benefits of GPs and Neural Networks, such as (Conditional) Neural Processes [61], [62], could be extended to crowdsourcing. In fact, these can also be endowed with the aforementioned convolutional structure [63].

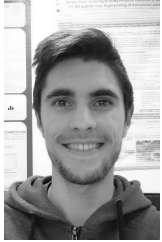
## REFERENCES

- [1] A. Irwin, "No phds needed: how citizen science is transforming research," *Nature*, vol. 562, pp. 480–482, 2018.
- [2] C. J. Guerrini, M. A. Majumder, M. J. Lewellyn, and A. L. McGuire, "Citizen science, public policy," *Science*, vol. 361, no. 6398, pp. 134–136, 2018.
- [3] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast, but is it good?: evaluating non-expert annotations for natural language tasks," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2008, pp. 254–263.
- [4] M. Buhrmester, T. Kwang, and S. D. Gosling, "Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data?" *Perspectives on psychological science*, vol. 6, no. 1, pp. 3–5, 2011.
- [5] S. Albarqouni, C. Baur, F. Achilles, V. Belagiannis, S. Demirci, and N. Navab, "AggNet: Deep Learning From Crowds for Mitosis Detection in Breast Cancer Histology Images," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1313–1321, 2016.
- [6] J. Saez-Rodriguez, J. C. Costello, S. H. Friend, M. R. Kellen, L. Man-gravite, P. Meyer, T. Norman, and G. Stolovitzky, "Crowdsourcing biomedical research: leveraging communities as innovation engines," *Nature Reviews Genetics*, vol. 17, no. 8, pp. 470–486, 2016.
- [7] S. Fritz, L. See, C. Perger, I. McCallum, C. Schill, D. Schepaschenko, M. Duerauer, M. Karner, C. Dresel, J.-C. Laso-Bayas *et al.*, "A global dataset of crowdsourced land cover and land use reference data," *Scientific data*, vol. 4, p. 170075, 2017.
- [8] F. Rodrigues, M. Lourenco, B. Ribeiro, and F. Pereira, "Learning supervised topic models for classification and regression from crowds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2409–2422, 2017.
- [9] E. Heim, A. Seitel, J. Andrulis, F. Isensee, C. Stock, T. Ross, and L. Maier-Hein, "Clickstream analysis for crowd-based object segmentation with confidence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2814–2826, 2018.
- [10] M. Zevin, S. Coughlin, S. Bahaadini, E. Besler, N. Rohani, S. Allen *et al.*, "Gravity spy: integrating advanced ligo detector characterization, machine learning, and citizen science," *Classical and Quantum Gravity*, vol. 34, no. 6, p. 064003, 2017.
- [11] A. Abramovici, W. E. Althouse, R. W. P. Drever, Y. Gürsel, S. Kawamura, F. J. Raab, D. Shoemaker, L. Sievers, R. E. Spero, K. S. Thorne, R. E. Vogt, R. Weiss, S. E. Whitcomb, and M. E. Zucker, "Ligo: The laser interferometer gravitational-wave observatory," *Science*, vol. 256, no. 5055, pp. 325–333, 1992.

- [12] B. P. Abbott et al. (LIGO Scientific Collaboration and Virgo Collaboration), "Observation of gravitational waves from a binary black hole merger," *Physical Review Letters*, vol. 116, p. 061102, Feb 2016.
- [13] D. Castelvecchi and A. Witze, "Einstein's gravitational waves found at last," *Nature news*, 2016.
- [14] V. S. Sheng, F. Provost, and P. G. Ipeirotis, "Get another label? improving data quality and data mining using multiple, noisy labelers," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 614–622.
- [15] P. Donmez and J. G. Carbonell, "Proactive learning: cost-sensitive active learning with multiple imperfect oracles," in *ACM Conference on Information and Knowledge Management*, 2008, pp. 619–628.
- [16] A. P. Dawid and A. M. Skene, "Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 20–28, 1979.
- [17] P. G. Ipeirotis, F. Provost, and J. Wang, "Quality Management on Amazon Mechanical Turk," in *ACM SIGKDD Workshop on Human Computation (HCOMP'10)*, 2010, pp. 64–67.
- [18] T.-F. Whitehill, J. and Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *Advances in Neural Information Processing Systems (NIPS)*, 2009, pp. 2035–2043.
- [19] V. Raykar, S. Yu, L. Zhao, G. Hermosillo Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," *The Journal of Machine Learning Research*, vol. 11, no. Apr, pp. 1297–1322, 2010.
- [20] Y. Yan, R. Rosales, G. Fung, R. Subramanian, and J. Dy, "Learning from multiple annotators with varying expertise," *Machine Learning*, vol. 95, no. 3, pp. 291–327, 2014.
- [21] F. Rodrigues, F. Pereira, and B. Ribeiro, "Gaussian process classification and active learning with multiple annotators," in *International Conference on Machine Learning (ICML)*, 2014, pp. 433–441.
- [22] P. Ruiz, P. Morales-Álvarez, R. Molina, and A. K. Katsaggelos, "Learning from crowds with variational gaussian processes," *Pattern Recognition*, vol. 88, pp. 298 – 311, 2019.
- [23] F. Rodrigues and F. Pereira, "Deep learning from crowds," in *Conference on Artificial Intelligence (AAAI)*, 2018, pp. 1611–1618.
- [24] M. Y. Guan, V. Gulshan, A. M. Dai, and G. E. Hinton, "Who said what: Modeling individual labelers improves classification," in *Conference on Artificial Intelligence (AAAI)*, 2018, pp. 3109–3118.
- [25] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT, 2006.
- [26] M. Bauer, M. van der Wilk, and C. E. Rasmussen, "Understanding probabilistic sparse gaussian process approximations," in *Advances in Neural Information Processing Systems (NIPS)*, 2016, pp. 1533–1541.
- [27] E. Snelson and Z. Ghahramani, "Sparse gaussian processes using pseudo-inputs," in *Advances in Neural Information Processing Systems (NIPS)*, 2006, pp. 1257–1264.
- [28] M. Titsias, "Variational learning of inducing variables in sparse gaussian processes," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 5, 2009, pp. 567–574.
- [29] P. Morales-Álvarez, A. Pérez-Suay, R. Molina, and G. Camps-Valls, "Remote sensing image classification with large-scale gaussian processes," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 1103–1114, 2018.
- [30] P. Morales-Álvarez, P. Ruiz, R. Santos-Rodríguez, R. Molina, and A. K. Katsaggelos, "Scalable and efficient learning from crowds with gaussian processes," *Information Fusion*, vol. 52, pp. 110 – 127, 2019.
- [31] J. Hensman, A. Matthews, and Z. Ghahramani, "Scalable Variational Gaussian Process Classification," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 38, 2015, pp. 351–360.
- [32] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.
- [33] J. Hensman, N. Fusi, and N. D. Lawrence, "Gaussian processes for big data," in *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013, pp. 282–290.
- [34] D. Burt, C. E. Rasmussen, and M. Van Der Wilk, "Rates of convergence for sparse variational gaussian process regression," in *International Conference on Machine Learning*, 2019, pp. 862–871.
- [35] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1303–1347, 2013.
- [36] D. G. Matthews, G. Alexander, M. Van Der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrà, Z. Ghahramani, and J. Hensman, "Gpflow: A gaussian process library using tensorflow," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 1299–1304, 2017.
- [37] J. Shi, M. E. Khan, and J. Zhu, "Scalable training of inference networks for gaussian-process models," in *International Conference on Machine Learning*, 2019, pp. 5758–5768.
- [38] F. P. Casale, A. Dalca, L. Saglietti, J. Listgarten, and N. Fusi, "Gaussian process prior variational autoencoders," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 369–10 380.
- [39] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *International Conference on Machine Learning*, 2015, pp. 1530–1538.
- [40] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference," *arXiv preprint arXiv:1912.02762*, 2019.
- [41] J. Hensman, A. G. Matthews, M. Filippone, and Z. Ghahramani, "Mcmc for variationally sparse gaussian processes," in *Advances in Neural Information Processing Systems*, 2015, pp. 1648–1656.
- [42] M. Tegner, B. Bloem-Reddy, and S. Roberts, "Sequential sampling of gaussian process latent variable models," *arXiv preprint arXiv:1807.04932*, 2018.
- [43] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
- [44] K. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT, 2012.
- [45] D. Hernández-Lobato, J. M. Hernández-Lobato, and P. Dupont, "Robust multi-class gaussian process classification," in *Advances in Neural Information Processing Systems (NIPS)*, 2011, pp. 280–288.
- [46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference for Learning Representations (ICLR)*, 2015.
- [47] F. W. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, *NIST handbook of mathematical functions*. Cambridge university press, 2010.
- [48] D. J. Kennefick, *Traveling at the speed of thought: Einstein and the quest for gravitational waves*. Princeton university press, 2016.
- [49] S. Bahaadini, V. Noroozi, N. Rohani, S. Coughlin, M. Zevin, J. Smith, V. Kalogera, and A. Katsaggelos, "Machine learning for gravity spy: Glitch classification and dataset," *Information Sciences*, vol. 444, pp. 172–186, 2018.
- [50] B. P. Abbott, R. Abbott, T. Abbott, M. Abernathy, F. Acernese, K. Ackley, M. Adamo, C. Adams, T. Adams, P. Addesso et al., "Characterization of transient noise in advanced ligo relevant to gravitational wave signal gw150914," *Classical and Quantum Gravity*, vol. 33, no. 13, p. 134001, 2016.
- [51] L. K. Nuttall, T. J. Massinger, J. Areeda, J. Betzwieser, S. Dwyer, A. Effler, R. P. Fisher, P. Fritschel, J. S. Kissel, A. P. Lundgren et al., "Improving the data quality of advanced ligo based on early engineering run results," *Classical and Quantum Gravity*, vol. 32, no. 24, p. 245005, 2015.
- [52] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [53] T. P. Minka, "A family of algorithms for approximate bayesian inference," Ph.D. dissertation, University of Cambridge, 2001.
- [54] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improved variational inference with inverse autoregressive flow," in *Advances in neural information processing systems*, 2016, pp. 4743–4751.
- [55] G. Papamakarios, T. Pavlakou, and I. Murray, "Masked autoregressive flow for density estimation," in *Advances in Neural Information Processing Systems*, 2017, pp. 2338–2347.
- [56] J. Chen, C. Lu, B. Chenli, J. Zhu, and T. Tian, "Vflow: More expressive generative flows with variational data augmentation," in *International Conference on Machine Learning*, 2020.
- [57] S. Sun, G. Zhang, J. Shi, and R. Grosse, "Functional variational bayesian neural networks," in *International Conference on Learning Representations*, 2019.
- [58] H. Salimbeni and M. Deisenroth, "Doubly stochastic variational inference for deep gaussian processes," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 4588–4599.
- [59] M. Van der Wilk, C. E. Rasmussen, and J. Hensman, "Convolutional gaussian processes," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 2849–2858.
- [60] K. Blomqvist, S. Kaski, and M. Heinonen, "Deep convolutional gaussian processes," *arXiv preprint arXiv:1810.03052*, 2018.



- [61] M. Garnelo, D. Rosenbaum, C. Maddison, T. Ramalho, D. Saxton, M. Shanahan, Y. W. Teh, D. Rezende, and S. A. Eslami, "Conditional neural processes," in *International Conference on Machine Learning*, 2018, pp. 1704–1713.
- [62] M. Garnelo, J. Schwarz, D. Rosenbaum, F. Viola, D. J. Rezende, S. Eslami, and Y. W. Teh, "Neural processes," *arXiv preprint arXiv:1807.01622*, 2018.
- [63] J. Gordon, W. P. Bruinsma, A. Y. Foong, J. Requeima, Y. Dubois, and R. E. Turner, "Convolutional conditional neural processes," *arXiv preprint arXiv:1910.13556*, 2019.



**Pablo Morales-Álvarez** received the B.Sc. degree in Maths from the University of Granada (UGR), Spain, in 2014. He obtained the First End of Studies Award by the Spanish Ministry of Science to the most outstanding undergraduate in mathematics. Then, he received the M.Sc. degrees in Mathematical Physics (2015) and Data Science (2016), both from UGR. Funded by the highly competitive Ph.D. fellowship from *La Caixa* Foundation, he is currently pursuing the Ph.D. degree at UGR under the supervision

of Prof. Rafael Molina, co-supervised by Prof. Aggelos K. Katsaggelos (Northwestern University, USA). He was selected (15% international acceptance ratio) to attend the 2017 Machine Learning Summer School at the MPI for Intelligent Systems at Tübingen (Germany), organized by Prof. Bernhard Schölkopf's group. He was visiting the Machine Learning Group at the University of Cambridge (UK) for five months between 2019 and 2020, to work with Prof. José Miguel Hernández-Lobato and his team. In March 2020 he was selected for an internship at Microsoft Research Cambridge (UK) to work with Cheng Zhang. He has served as a reviewer for top-tier conferences (ICML 2020, NeurIPS 2020, also invited for ICLR 2021), and journals (Science). His main research interests are probabilistic machine learning methods, specially (Deep) Gaussian Processes and Bayesian Neural Networks, and their applications to crowdsourcing and citizen science problems.



**Pablo Ruiz** received his BS Degree in Mathematics (2008), Master Degree in Multimedia Technologies (2009) and PhD in Information and Communication Technologies (2015), all of them from the University of Granada. Currently he is in his third year as a postdoctoral fellow at Northwestern University under the supervision of Prof. Katsaggelos. In his research he studies probabilistic models and inference applied to the solution of inverse problems. Using these tools he is addressing a number of signal and image

processing problems such as image restoration, blind deconvolution, image and 3D volume restoration from images acquired with computational tomography, phase retrieval, as well as Machine Learning problems, where he is using the Gaussian Processes theory to solve regression and classification problems, active learning, feature learning and crowdsourcing among others.



**Scott Coughlin** received a BA degree from Northwestern University, in 2014. He is currently working as a research specialist at the Center for Interdisciplinary Exploration and Research in Astrophysics (CIERA, Northwestern University), and enrolled as a graduate student at the School of Physics and Astronomy at Cardiff University (UK). He leads and manages the citizen science project known as Gravity Spy. Gravity Spy is an interdisciplinary research project that combines his interests of gravitational-wave data analysis,

machine learning and citizen science with the goal of classifying non-astronomical noise artifacts in gravitational wave data.



**Rafael Molina** received the degree in mathematics (statistics) and the Ph.D. degree in optimal design in linear models from the University of Granada, Granada, Spain, in 1979 and 1983, respectively. He was the Dean of the Computer Engineering School, University of Granada, from 1992 to 2002. In 2000, he joined the University of Granada, as a Professor of computer science and artificial intelligence. He was the Head of the Computer Science and Artificial Intelligence Department, University of Granada, from 2005 to

2007. He has co-authored the paper that received the runner-up prize at reception for early stage researchers at the House of Commons. His research interest focuses mainly on using Bayesian modeling and inference in problems like image restoration (applications to astronomy and medicine), super-resolution of images and video, blind deconvolution, computational photography, source recovery in medicine, compressive sensing, low-rank matrix decomposition, active learning, fusion, and machine learning. He was a recipient of the IEEE International Conference on Image Processing Paper Award in 2007 and the ISPA Best Paper Award in 2009. He served as an Associate Editor for *Applied Signal Processing* from 2005 to 2007 and the *IEEE TRANSACTIONS ON IMAGE PROCESSING* from 2010 to 2014. He has been serving as an Associate Editor for *Progress in Artificial Intelligence* since 2011. He has been serving as an Area Editor for *Digital Signal Processing* since 2011.



**Aggelos K. Katsaggelos** received the Diploma degree in electrical and mechanical engineering from the Aristotelian University of Thessaloniki, Greece, in 1979, and the M.S. and Ph.D. degrees in electrical engineering from the Georgia Institute of Technology in 1981 and 1985, respectively. In 1985, he joined the Department of Electrical Engineering and Computer Science, Northwestern University. He was the Ameritech Chair of information technology and the AT&T Chair. He is currently a Professor of the Joseph

Cummings Chair, Northwestern University. He is also an Academic Staff Member with NorthShore University Health System and an Affiliated Faculty Member with the Department of Linguistics. He has an appointment with the Argonne National Laboratory. He has published extensively in the areas of multimedia signal processing and communications, computational imaging, and machine learning (over 250 journal papers, 600 conference papers, and 40 book chapters). He holds 25 international patents. He has co-authored *Rate-Distortion Based Video Compression* (Kluwer, 1997), *Super-Resolution for Images and Video* (Claypool, 2007), *Joint Source-Channel Video Transmission* (Claypool, 2007), and *Machine Learning Refined* (Cambridge University Press, 2016). He has supervised 56 Ph.D. dissertations. Among his many professional activities, he was the Editor-in-Chief of the *IEEE Signal Processing Magazine* from 1997 to 2002, a BOG Member of the *IEEE Signal Processing Society* from 1999 to 2001, a member of the Publication Board of the *IEEE PROCEEDINGS* from 2003 to 2007, and a member of the Award Board of the *IEEE Signal Processing Society*. He is a fellow of the IEEE in 1998, SPIE in 2009, EURASIP in 2017, and OSA in 2018. He was a recipient of the IEEE Third Millennium Medal in 2000, the IEEE Signal Processing Society Meritorious Service Award in 2001, the IEEE Signal Processing Society Technical Achievement Award in 2010, the IEEE Signal Processing Society Best Paper Award in 2001, the IEEE ICME Paper Award in 2006, the IEEE ICIP Paper Award in 2007, the ISPA Paper Award in 2009, and the EUSIPCO Paper Award in 2013. He was a Distinguished Lecturer of the IEEE Signal Processing Society from 2007 to 2008.