



Hyperbolic Secant representation of the logistic function: Application to probabilistic Multiple Instance Learning for CT intracranial hemorrhage detection

Francisco M. Castro-Macías^{a,b,*}, Pablo Morales-Álvarez^{c,b}, Yunan Wu^d,
Rafael Molina^a, Aggelos K. Katsaggelos^{d,e}

^a Department of Computer Science and Artificial Intelligence, University of Granada, Spain

^b Research Centre for Information and Communication Technologies, University of Granada, Spain

^c Department of Statistics and Operations Research, University of Granada, Spain

^d Department of Electrical and Computer Engineering, Northwestern University, USA

^e Center for Computational Imaging and Signal Analytics in Medicine, Northwestern University, USA

ARTICLE INFO

Keywords:

Multiple Instance Learning
Gaussian processes
Jaakkola bound
Pólya-Gamma
Hyperbolic Secant distribution
Variational inference
Intracranial hemorrhage detection

ABSTRACT

Multiple Instance Learning (MIL) is a weakly supervised paradigm that has been successfully applied to many different scientific areas and is particularly well suited to medical imaging. Probabilistic MIL methods, and more specifically Gaussian Processes (GPs), have achieved excellent results due to their high expressiveness and uncertainty quantification capabilities. One of the most successful GP-based MIL methods, VGPMIL, resorts to a variational bound to handle the intractability of the logistic function. Here, we formulate VGPMIL using Pólya-Gamma random variables. This approach yields the same variational posterior approximations as the original VGPMIL, which is a consequence of the two representations that the Hyperbolic Secant distribution admits. This leads us to propose a general GP-based MIL method that takes different forms by simply leveraging distributions other than the Hyperbolic Secant one. Using the Gamma distribution we arrive at a new approach that obtains competitive or superior predictive performance and efficiency. This is validated in a comprehensive experimental study including one synthetic MIL dataset, two well-known MIL benchmarks, and a real-world medical problem. We expect that this work provides useful ideas beyond MIL that can foster further research in the field.

1. Introduction

Multiple Instance Learning (MIL) [1] is a type of weakly supervised learning that has become very popular due to the reduced annotation effort it requires. In MIL binary classification [2] the training set consists of instances grouped into bags. Both bags and instances have labels, but we only observe them at the bag level while instance labels remain unknown. It is assumed that a bag label is positive if and only if it contains at least one positive instance. The goal is to achieve a method capable of accurately predicting both bag and instance labels using only bag labels.

* Corresponding author at: Department of Computer Science and Artificial Intelligence, University of Granada, Spain.
E-mail address: fcastro@ugr.es (F.M. Castro-Macías).

The MIL approach has been successfully applied to many different scientific domains [1], being particularly well suited to medical imaging [3]. In this work, we are particularly interested in the problem of IntraCranial Hemorrhage (ICH) detection. ICH is a severe life-threatening emergency with high mortality and morbidity rates caused by blood leakage inside the brain [4]. Computed Tomography (CT) scans are widely used to diagnose ICH because it is an inexpensive and non-invasive technique for patients. Each scan is made up of a significant number of slices, each representing a section of the head at a given height. A CT scan is labeled as ICH if at least one of its slices shows evidence of the injury, otherwise, it is normal. To apply a supervised learning approach, radiologists have to manually label every single slice in the dataset [5–7], which is a costly and tedious process. In contrast, the MIL approach significantly reduces the radiologists' workload as only one label for each scan is needed.

To learn in the MIL scenario, Deep Learning (DL) methods have become popular in practice due to their ability to deal with highly structured data [3,8–10]. The most successful models combine DL architectures with attention mechanisms to weigh the relevance of each instance, see [11]. However, these methods do not model the instance label *explicitly* (they just have an attention weight for each instance), which hampers the quantification of uncertainty at the instance level. Note that this is essential in MIL since instance labels are unknown. As a consequence, plenty of attention has been paid to probabilistic MIL methods in recent years. Among them, Gaussian Processes (GPs) have achieved very competitive results [12–17], due to their high expressiveness and uncertainty quantification capabilities. Most of these GP-based MIL methods build on the popular VGPMIL [14], which formulates the MIL problem through sparse GPs for classification using the logistic function.

In order to achieve mathematical tractability of the logistic function, VGPMIL introduces a variational bound known as the Jaakkola bound. Therefore, the training objective becomes a lower bound of the real one. As recently shown in [16], this theoretical approximation degrades the predictive performance in practice. An alternative and exact treatment of the logistic function has recently been introduced in the context of GPs for supervised classification, see [18–20]. The idea is to augment the model using Pólya-Gamma variables, obtaining an equivalent and tractable formulation. However, to the best of our knowledge, these ideas have never been adapted to the MIL scenario.

In this work, we first reformulate the VGPMIL model using Pólya-Gamma random variables. We find that this new model, called PG-VGPMIL, is equivalent to the existing VGPMIL when performing closed-form variational inference updates (they lead to the same update equations). This phenomenon, which was also observed in supervised classification [18,19], finds its justification in the properties of the Hyperbolic Secant density [21]. Thanks to them, the logistic observation model admits two representations [22], as a Super Gaussian and as a Gaussian Scale Mixture (GSM), that lead to the same variational optimization objective [18, Theorem 2.1]. We build upon the GSM representation to formulate ψ -VGPMIL, a general model where ψ is a differentiable GSM density. When ψ is the Hyperbolic Secant density, we recover the original VGPMIL. Although ψ -VGPMIL is formulated using GPs, it can be extended to other related probabilistic frameworks, such as Relevance Vector Machines [23,24].

Inspired by the above connection and by the definition of Pólya-Gamma variables [20], we replace the Pólya-Gamma distribution with the Gamma distribution, obtaining a new GSM density and thus a new particularization of ψ -VGPMIL, which we refer to as G-VGPMIL. The proposed algorithm is evaluated through a comprehensive set of experiments involving different datasets, baselines and metrics (both at the instance and bag level). First, we focus on the G-VGPMIL method in a controlled experiment built around the MNIST dataset. This allows us to understand its behavior in practice and devise its main properties. Second, we utilize two historically important MIL benchmark datasets (MUSK1 and MUSK2). This second experiment again shows that G-VGPMIL improves the existing VGPMIL approach in terms of efficiency and performance. Third, we show that G-VGPMIL achieves better results than state-of-the-art methods for the ICH detection problem.

In summary, the main contributions of this work are the following:

- We introduce PG-VGPMIL, a new probabilistic MIL model based on Pólya-Gamma random variables. To the best of our knowledge, Pólya-Gamma variables have never been used before in the context of MIL. We observe that PG-VGPMIL is equivalent to the existing VGPMIL when performing closed-form variational inference updates.
- Building on the theory behind this equivalence, we develop ψ -VGPMIL, a general inference framework for the logistic observation model. New inference models can be obtained using different GSM densities, denoted by ψ . PG-VGPMIL (and hence VGPMIL) becomes a particular case of this framework when ψ is the Hyperbolic Secant density.
- We use the Gamma distribution to obtain a new GSM density, which is used in the aforementioned general framework to obtain G-VGPMIL, a new inference model for MIL. To the best of our knowledge, the Gamma distribution has never been used in the context of MIL.
- The newly proposed G-VGPMIL is compared to state-of-the-art approaches using different metrics and datasets, including a real-world ICH detection problem. The experiments conducted show enhanced predictive performance and efficiency.

The rest of the paper is organized as follows. In Section 2 we introduce the MIL problem and its GP-based formulation. In Section 3 we introduce PG-VGPMIL and investigate its equivalence to VGPMIL. In Section 4 we devise ψ -VGPMIL, a new general framework of which VGPMIL is a particular realization. In Section 5 we introduce our new model G-VGPMIL as another particular realization of ψ -VGPMIL. In Section 6 we empirically validate the newly proposed method. In Section 7 we summarize the main conclusions of this work.

2. Gaussian Processes for Multiple Instance Learning

The goal of this section is to introduce the MIL problem and how GPs are used to address it.

2.1. Problem formulation: Multiple Instance Learning

We consider a dataset $\{(\mathbf{x}_n, y_n) : n \in \{1, \dots, N\}\} \subset \mathbb{R}^D \times \{0, 1\}$, which consists of the training instances \mathbf{x}_n and their binary labels y_n . We express this set in a matrix $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N] \in \mathbb{R}^{D \times N}$ and a vector $\mathbf{y} = [y_1, \dots, y_N]^\top \in \{0, 1\}^N$.

In the MIL scenario, the instance labels y_n are not observed. Instead, they are grouped into bags and only the maximum of the labels of the instances in a bag is observed. Formally, the index set $\{1, \dots, N\}$ is partitioned into B non-overlapping bags $\{\text{Bag}_1, \dots, \text{Bag}_B\}$ and the operator $\{\cdot\}_b$ is used to refer to the elements in bag b , so $\{\mathbf{y}\}_b = \{y_i : i \in \text{Bag}_b\}$. The operator $\{\cdot\}_{b \setminus n}$ is used to refer to the elements in bag b except for the n -th element, so $\{\mathbf{y}\}_{b \setminus n} = \{\mathbf{y}\}_b \setminus \{y_n\}$. Finally, for a bag b , the only observed label is $T_b = \max\{\mathbf{y}\}_b$. The goal is to predict the label for any new bag, as well as the labels of the instances in the bag.

Notice that the ICH detection problem presented in the introduction can be cast as an MIL one. A full scan is treated as a bag and slices are treated as instances. Only bag labels T_b are observed, while slice labels y_n are unobserved. A CT scan presents hemorrhage (positive class) if at least one of its slices show evidence of hemorrhage. A negative scan contains only normal slices.

2.2. Variational Gaussian Processes for Multiple Instance Learning

Before considering the GP MIL problem, we briefly summarize the general framework for GP Logistic classification [25]. Given a training dataset (\mathbf{X}, \mathbf{y}) , GP classification assumes the existence of a latent function $f : \mathbb{R}^D \rightarrow \mathbb{R}$ that determines the class of an instance. The model places a GP prior over this function, $p(\mathbf{f}) = \mathcal{N}(\mathbf{f} | \mathbf{0}, \mathbf{K}_{\mathbf{X}\mathbf{X}})$, and uses the observation model

$$p(y_n | f_n) = \text{Bernoulli}(y_n | \text{logit}(f_n)), \quad p(\mathbf{y} | \mathbf{f}) = \prod_{n=1}^N p(y_n | f_n), \quad (1)$$

where $\text{logit}(t) = 1/(1 + e^{-t})$ is the logistic function and $\mathbf{f} = [f_1, \dots, f_N]^\top = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)]^\top$ are the realizations of the latent function f over the training set. The matrix $\mathbf{K}_{\mathbf{X}\mathbf{X}}$ is defined as $(\mathbf{K}_{\mathbf{X}\mathbf{X}})_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, where $k : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$ is a positive-definite kernel [25]. In this work, we will use the popular Radial Basis Function (RBF) kernel, which is defined as $k(\mathbf{x}, \mathbf{x}') = v \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|}{2l}\right)$, where $v, l > 0$ will be treated as hyperparameters. In the following, we denote $\lambda = \{v, l\}$.

GP inference requires inverting an $N \times N$ matrix related to the kernel. This has a cost of $O(N^3)$, so the full model can not be used for large datasets. This has motivated the popularization of *sparse GP* methods. Following the *fully independent training conditional (FITC)* approximation [26], we introduce a set of M inducing points $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_M]$ and their corresponding output via f , $\mathbf{u} = f(\mathbf{Z})$. Mirroring the relation between \mathbf{X} and \mathbf{f} , we have

$$p(\mathbf{u}) = \mathcal{N}(\mathbf{u} | \mathbf{0}, \mathbf{K}_{\mathbf{Z}\mathbf{Z}}), \quad (2)$$

$$p(\mathbf{f} | \mathbf{u}) = \mathcal{N}(\mathbf{f} | \mathbf{a}, \tilde{\mathbf{K}}), \quad (3)$$

where $\mathbf{a} = \mathbf{K}_{\mathbf{X}\mathbf{Z}}\mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1}\mathbf{u}$ and $\tilde{\mathbf{K}} = \mathbf{K}_{\mathbf{X}\mathbf{X}} - \mathbf{K}_{\mathbf{X}\mathbf{Z}}\mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1}\mathbf{K}_{\mathbf{Z}\mathbf{X}}$. Here, $(\mathbf{K}_{\mathbf{Z}\mathbf{Z}})_{ij} = k(\mathbf{z}_i, \mathbf{z}_j)$, $(\mathbf{K}_{\mathbf{X}\mathbf{Z}})_{ij} = k(\mathbf{x}_i, \mathbf{z}_j)$ and $\mathbf{K}_{\mathbf{Z}\mathbf{X}} = \mathbf{K}_{\mathbf{X}\mathbf{Z}}^\top$. This approach reduces the cost to $O(M^2N)$.

In the MIL scenario, instead of instance labels $\{y_n\}$ we have only bag labels T_b . To adapt the previous classifier to this setting, VGPMIL [14] introduces the following expression of the bag label likelihood given the labels of the instances,

$$p(T_b | \{\mathbf{y}\}_b) = \frac{H^{G_b}}{H+1}, \quad p(\mathbf{T} | \mathbf{y}) = \prod_{b=1}^B p(T_b | \{\mathbf{y}\}_b), \quad (4)$$

where $H > 0$ (set to 100 in [14]) and $G_b = T_b \max\{\mathbf{y}\}_b + (1 - T_b)(1 - \max\{\mathbf{y}\}_b)$. This replaces the deterministic value of T_b given $\max\{\mathbf{y}\}_b$ by a noisy probabilistic one. If $\max\{\mathbf{y}\}_b = 1$, then $p(T_b = 1 | \{\mathbf{y}\}_b) = H/(H+1)$ and $p(T_b = 0 | \{\mathbf{y}\}_b) = 1/(H+1)$ (a very small number). On the contrary, if $\max\{\mathbf{y}\}_b = 0$, then $p(T_b = 0 | \{\mathbf{y}\}_b) = H/(H+1)$ and $p(T_b = 1 | \{\mathbf{y}\}_b) = 1/(H+1)$ (again, a very small number). The complete VGPMIL model is given by the product of the distributions in Eq. (1), (2), (3), and (4).

In order to make predictions in VGPMIL it is necessary to compute the posterior distribution $p(\mathbf{u}, \mathbf{f}, \mathbf{y} | \mathbf{T})$, which is not analytically tractable. The original work [14] resorts to mean-field variational inference [27] to approximate it by a variational distribution $q(\mathbf{u}, \mathbf{f}, \mathbf{y}) = q(\mathbf{u})q(\mathbf{f} | \mathbf{u})q(\mathbf{y})$ with $q(\mathbf{y}) = \prod_{n=1}^N q(y_n)$, that is selected minimizing the Kullback-Leibler (KL) divergence between the variational distribution approximation and the true posterior. The solution for each factor is given by [27, Eq. (10.9)].

Since the logistic function used in Eq. (1) is not conjugate to the Gaussian distribution, it is impossible to obtain tractable expressions for the variational factors. To deal with this, VGPMIL uses the Jaakkola bound [28]. In the next subsection we explore an augmented version of VGPMIL based on Pólya-Gamma random variables [20], in which inference is tractable.

3. Pólya-Gamma variables for GPMIL

The class of Pólya-Gamma random variables was introduced in [20] to propose a *data augmentation* approach for stochastic Bayesian inference in models with binomial likelihood. Later, the multinomial distribution was reformulated in terms of Pólya-Gamma variables considering GPs with multinomial observations [29]. Finally, [19] proposed a stochastic variational approach to

GP classification using an augmented model based on Pólya-Gamma variables and inducing points. Next, we consider the Pólya-Gamma trick in the context of MIL GP classification.

The Pólya-Gamma distribution $\text{PG}(b, c)$, with $b > 0$ and $c \in \mathbb{R}$, has an important property that is closely related to our problem,

$$\text{logit}(x) = 2^{-1} \exp(x/2) \mathbb{E}_{\text{PG}(\omega|1,0)} [\exp(-x^2\omega/2)], \quad \forall x \in \mathbb{R}. \quad (5)$$

This equality ensures that the following joint density is well defined

$$p(y_n, \omega_n | f_n) = 2^{-1} \exp((y_n - 1/2)f_n) \exp(-f_n^2\omega_n/2) \text{PG}(\omega_n | 1, 0), \quad (6)$$

and that $p(y_n | f_n) = \int_0^\infty p(y_n, \omega_n | f_n) d\omega_n$ is the logistic observation model. Writing $\boldsymbol{\omega} = [\omega_1, \dots, \omega_N]^\top$, $\boldsymbol{\Omega} = \text{Diag}(\omega_1, \dots, \omega_N)$, and assuming independence between instances, we have

$$p(\mathbf{y}, \boldsymbol{\omega} | \mathbf{f}) = 2^{-N} \exp((\mathbf{y} - 2^{-1}\mathbf{1})^\top \mathbf{f} - 2^{-1}\mathbf{f}^\top \boldsymbol{\Omega} \mathbf{f}) \prod_{n=1}^N \text{PG}(\omega_n | 1, 0), \quad (7)$$

where $\mathbf{1} = [1, \dots, 1]^\top$. Note that we are considering an equivalent model where we have removed the use of the logistic function without introducing any approximation. The augmented VGPMIL model is defined by the product of the distributions in Eq. (2), (3), (4) and (7). We will refer to this model by *Pólya Gamma Variational Gaussian Process Multiple Instance Learning (PG-VGPMIL)*.

Inference in PG-VGPMIL. We mimic the inference procedure followed by VGPMIL. We approximate the posterior distribution $q(\mathbf{u}, \mathbf{f}, \mathbf{y}, \boldsymbol{\omega} | \mathbf{T})$ with a variational distribution $q(\mathbf{u}, \mathbf{f}, \mathbf{y}, \boldsymbol{\omega}) = q(\mathbf{u})p(\mathbf{f} | \mathbf{u})q(\mathbf{y})q(\boldsymbol{\omega})$, where $q(\mathbf{y}) = \prod_{n=1}^N q(y_n)$, and $q(\boldsymbol{\omega}) = \prod_{n=1}^N q(\omega_n)$, minimizing the KL divergence between them. Applying [27, Eq. (10.9)], we obtain

$$q(\mathbf{u}) = \mathcal{N}(\mathbf{u} | \mathbf{m}, \mathbf{S}), \quad (8)$$

$$q(y_n) = \text{Bernoulli}(y_n | \pi_n), \quad (9)$$

$$q(\omega_n) = \text{PG}(\omega_n | 1, c_n), \quad (10)$$

$$\pi_n = \text{logit} \left[\mathbf{K}_{\mathbf{x}_n \mathbf{Z}} \mathbf{K}_{\mathbf{Z} \mathbf{Z}}^{-1} \mathbf{m} + \log H(2T_b - 1) (1 - \mathbb{E}[\max\{\mathbf{y}\}_{b \setminus n}]) \right], \quad (11)$$

$$\mathbf{m} = \mathbf{S} \mathbf{K}_{\mathbf{Z} \mathbf{Z}}^{-1} \mathbf{K}_{\mathbf{Z} \mathbf{X}} (\boldsymbol{\pi} - 2^{-1}\mathbf{1}), \quad (12)$$

$$\mathbf{S} = (\mathbf{K}_{\mathbf{Z} \mathbf{Z}}^{-1} \mathbf{K}_{\mathbf{Z} \mathbf{X}} \boldsymbol{\Theta} \mathbf{K}_{\mathbf{X} \mathbf{Z}} \mathbf{K}_{\mathbf{Z} \mathbf{Z}}^{-1} + \mathbf{K}_{\mathbf{Z} \mathbf{Z}}^{-1})^{-1}, \quad (13)$$

$$\boldsymbol{\Theta} = \text{Diag}(\theta(c_1), \dots, \theta(c_n)), \quad c_n = \sqrt{\mathbb{E}_{q(f_n)}[f_n^2]}, \quad \theta(c) = \frac{\tanh(c/2)}{2c}, \quad (14)$$

where $q(f_n) = \int_{\mathbb{R}^M} q(\mathbf{u})p(f_n | \mathbf{u})d\mathbf{u}$ and $\boldsymbol{\pi} = [\pi_1, \dots, \pi_n]^\top$. It is worth mentioning that $\theta(c_n) = \mathbb{E}_{q(\omega_n)}[\omega_n]$. The derivation of these equations can be found in Appendix A. Note that training PG-VGPMIL boils down to iterating over the above equations. If we compare these updates with the ones from VGPMIL, we find that they are identical. We explore this in the following subsection.

3.1. On the equivalence between PG-VGPMIL and VGPMIL

We observe that the update equations for PG-VGPMIL are equivalent to those of VGPMIL. In both models, \mathbf{m} and π_n have the same expressions, recall [14, Eq. (12), Eq. (14)]. Also, the update for ξ_n in VGPMIL is the same as c_n here. Finally, from $\text{logit}(x) = (\tanh(x/2) + 1)/2$, we have that $\boldsymbol{\Lambda} = \boldsymbol{\Theta}$ and so the covariance matrix \mathbf{S} is also the same [14, Eq. (11)].

This phenomenon was also observed in the context of supervised classification [18,19], as a consequence of two different representations of the Hyperbolic Secant density, and generalizes naturally to our setting of MIL. Namely, note that the logistic likelihood can be written as

$$p(y_n | f_n) = \pi \exp((y_n - 2^{-1})f_n) \phi(f_n), \quad (15)$$

where $\phi(x) = (2\pi \cosh(x/2))^{-1}$ is the Hyperbolic Secant density proposed in [21]. This density prevents us from calculating variational updates analytically. It admits two representations: as a Super Gaussian (SG), and as a Gaussian Scale Mixture (GSM), the former being a consequence of the latter [22,30]. The SG representation leads to the Jaakkola bound used in VGPMIL. The GSM representation leads to PG-VGPMIL and is obtained from Eq. (5),

$$\phi(x) = \int_0^{+\infty} \mathcal{N}(x | 0, \omega^{-1}) \hat{\phi}(\omega) d\omega, \quad \forall x \in \mathbb{R}, \quad (16)$$

where $\hat{\phi}(\omega) = (2\pi\omega)^{-1} \text{PG}(\omega | 1, 0)$. These representations produce inference schemes that may appear different. However, both approaches lead to optimize the same objective, as demonstrated in [18, Theorem 2.1] for the supervised logistic regression model. This result follows straightforwardly for our setting involving MIL and GPs.

Besides being responsible for the equivalence between the two approaches presented so far, the Hyperbolic Secant density ϕ plays a fundamental role in the VGPMIL variational updates. Recall that the matrix $\boldsymbol{\Theta}$ in Eq. (14) is computed using the function θ , which

is completely determined by ϕ , see Appendix A. As pointed out in previous works, one could try to improve the inference procedure by modifying θ [31]. In the following section, we formalize this idea, and show that it corresponds to replacing ϕ by a different GSM density.

4. A general inference framework for the logistic observation model

The GP logistic observation model can be written as

$$p(\mathbf{u}, \mathbf{f}, \mathbf{y}) = Z^{-1} \mathcal{N}(\mathbf{f}, \mathbf{u}) \exp\left((\mathbf{y} - 2^{-1}\mathbf{1})^\top \mathbf{f}\right) \phi(\mathbf{f}), \quad (17)$$

where $Z = \pi^{-N}$, $\phi(\mathbf{f}) = \prod_{n=1}^N \phi(f_n)$, and $\mathcal{N}(\mathbf{f}, \mathbf{u}) = \mathcal{N}(\mathbf{f} | \mathbf{u}) \mathcal{N}(\mathbf{u})$ denotes the joint distribution given by equations (2) and (3). To extend this model, first we consider a differentiable density $\psi : \mathbb{R} \rightarrow \mathbb{R}$ that admits a GSM representation. Next, we replace ϕ by ψ ,

$$p(\mathbf{u}, \mathbf{f}, \mathbf{y}) = Z^{-1} \mathcal{N}(\mathbf{f}, \mathbf{u}) \exp\left((\mathbf{y} - 2^{-1}\mathbf{1})^\top \mathbf{f}\right) \psi(\mathbf{f}), \quad (18)$$

where $\psi(\mathbf{f}) = \prod_{n=1}^N \psi(f_n)$ and the normalization constant is

$$Z = \sum_{\mathbf{y} \in \{0,1\}^N} \int_{\mathbb{R}^N} \mathcal{N}(\mathbf{f}) \exp\left((\mathbf{y} - 2^{-1}\mathbf{1})^\top \mathbf{f}\right) \psi(\mathbf{f}) d\mathbf{f}, \quad (19)$$

where $\mathcal{N}(\mathbf{f}) = \int_{\mathbb{R}^M} \mathcal{N}(\mathbf{f}, \mathbf{u}) d\mathbf{u}$. Using $\sum_{\mathbf{y} \in \{0,1\}^N} \exp\left((\mathbf{y} - 2^{-1}\mathbf{1})^\top \mathbf{f}\right) = \pi^{-N} \phi(\mathbf{f})^{-1}$, we can write

$$Z = \pi^{-N} \int_{\mathbb{R}^N} \mathcal{N}(\mathbf{f}) \psi(\mathbf{f}) \phi(\mathbf{f})^{-1} d\mathbf{f}. \quad (20)$$

Given that ψ must be bounded, the above integral must be dominated by $\int_{\mathbb{R}^N} \mathcal{N}(\mathbf{f}) \phi(\mathbf{f})^{-1} d\mathbf{f}$. As $\mathcal{N}(\mathbf{f})$ is a Gaussian and $\phi(\mathbf{f})^{-1}$ is analytic (as a product of analytic functions), then $\mathcal{N}(\mathbf{f}) \phi(\mathbf{f})^{-1}$ is integrable [32]. Therefore, the integral in Eq. (20) is finite.

In our new framework, $p(\mathbf{y} | \mathbf{f})$ remains the same as in Eq. (1), but the GP prior distribution change according to

$$p(\mathbf{f} | \mathbf{u}) \propto \mathcal{N}(\mathbf{f}, \mathbf{u}) \psi(\mathbf{f}) \phi(\mathbf{f})^{-1}, \quad (21)$$

$$p(\mathbf{f}) \propto \mathcal{N}(\mathbf{f}) \psi(\mathbf{f}) \phi(\mathbf{f})^{-1}. \quad (22)$$

This means that we no longer consider a GP prior on $p(\mathbf{f})$, which is now a Gaussian weighted by the ratio $\psi(\mathbf{f})\phi(\mathbf{f})^{-1}$. This new approach increases the flexibility of our model, since we can explore different options for ψ and subsequently select the best for our problem. One way to accomplish this is considering a parametric family and looking for the optimal parameters. Recall that the original model can be recovered by setting $\psi = \phi$, which highlights that the new framework is an extension of the previous one.

To adapt this new framework to MIL, we multiply the joint distribution in Eq. (18) by the bag likelihood defined in Eq. (4). We will refer to this extended model as ψ Variational Gaussian Processes Multiple Instance Learning (ψ -VGPMIL).

Inference in ψ -VGPMIL. We leverage the GSM representation of ψ and perform inference in the augmented model, similar to PG-VGPMIL. Thus, we approximate the posterior distribution $p(\mathbf{u}, \mathbf{f}, \mathbf{y} | \mathbf{T})$ with a variational distribution $q(\mathbf{u}, \mathbf{f}, \mathbf{y}) = q(\mathbf{u})q(\mathbf{f} | \mathbf{u})q(\mathbf{y})$, where $q(\mathbf{f} | \mathbf{u}) = \mathcal{N}(\mathbf{f} | \mathbf{u})$, and $q(\mathbf{y}) = \prod_{n=1}^N q(y_n)$.

Our choice of $q(\mathbf{f} | \mathbf{u})$ is not optimal, but it allows us to derive tractable expressions for the variational updates. Note that $q(\mathbf{f} | \mathbf{u}) = p(\mathbf{f} | \mathbf{u})$ is a better approximation [33], but yields intractable expressions since $p(\mathbf{f} | \mathbf{u})$ is no longer a Gaussian, see Eq. (21). Our approach could be improved by giving more flexibility to $q(\mathbf{f} | \mathbf{u})$, e.g. using normalizing flows [34].

The optimal expressions for $q(\mathbf{u})$ and $q(\mathbf{y})$ are obtained minimizing the KL divergence between the posterior and the variational distributions. Applying [27, Eq. (10.9)], we obtain

$$q(\mathbf{u}) = \mathcal{N}(\mathbf{u} | \mathbf{m}, \mathbf{S}), \quad (23)$$

$$q(y_n) = \text{Bernoulli}(y_n | \pi_n), \quad (24)$$

$$\pi_n = \text{logit} \left[\mathbf{K}_{x_n \mathbf{Z}} \mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1} \mathbf{m} + \log H(2T_b - 1) (1 - \mathbb{E}[\max\{\mathbf{y}\}_{b \setminus n}]) \right], \quad (25)$$

$$\mathbf{m} = \mathbf{S} \mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1} \mathbf{K}_{\mathbf{Z}\mathbf{X}} (\boldsymbol{\pi} - 2^{-1}\mathbf{1}), \quad (26)$$

$$\mathbf{S} = (\mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1} \mathbf{K}_{\mathbf{Z}\mathbf{X}} \boldsymbol{\Theta} \mathbf{K}_{\mathbf{X}\mathbf{Z}} \mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1} + \mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1})^{-1}, \quad (27)$$

$$\boldsymbol{\Theta} = \text{Diag}(\theta(c_1), \dots, \theta(c_N)), \quad c_n = \sqrt{\mathbb{E}_{q(f_n)}[f_n^2]}, \quad \theta(c) = -\frac{\psi'(c)}{c\psi(c)}, \quad (28)$$

where $q(f_n) = \int_{\mathbb{R}^M} q(\mathbf{u}) p(f_n | \mathbf{u}) d\mathbf{u}$ and $\boldsymbol{\pi} = [\pi_1, \dots, \pi_N]^\top$. The expression of $\theta(c_n)$ reveals why ψ must be differentiable. As in PG-VGPMIL, $\theta(c_n)$ represents the expectation of the augmenting variables. We provide more details on the derivation of these equations in Appendix A. Again, note that if $\psi = \phi$, we recover the original VGPMIL model

Kernel hyperparameters estimation in ψ -VGPMIL. The variational framework allows us to estimate the kernel hyperparameters λ . Following [35], we aim to maximize the ELBO with respect to them. This is computationally equivalent to placing a flat improper prior $p(\lambda) \propto \text{const}$, and taking the mode of the approximated posterior $q(\lambda)$ as an estimate. The objective to be maximized is

$$\mathcal{J}(\lambda) = -\text{KL}(q(\mathbf{u}), p(\mathbf{u})) + (\boldsymbol{\pi} - 2^{-1}\mathbf{1})^\top \boldsymbol{\mu} + \sum_{n=1}^N \mathbb{E}_{q(f_n)} [\log \psi(f_n)] - \log Z, \quad (29)$$

where $\boldsymbol{\mu} = \mathbf{K}_{\mathbf{X}\mathbf{Z}}\mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1}\mathbf{m}$ is the mean of $q(\mathbf{f})$. We optimize $\mathcal{J}(\lambda)$ using gradient ascent for a fixed number of iterations, for which we approximate the expectation on the third and fourth terms using Monte Carlo sampling [36]. Note that Z depends on the kernel hyperparameters through $\mathcal{N}(\mathbf{f})$. Since $\log Z$ requires sampling from $\mathcal{N}(\mathbf{f})$, which is costly, we decide to further approximate those samples using Random Fourier Features [37].

It is worth mentioning that the hyperparameter estimation procedure was not implemented in the original VGPMIL. Thus, our work strictly generalizes the model proposed in [14].

The training procedure of ψ -VGPMIL is detailed in Algorithm 1. An iteration updates the variational parameters using Eq. (25) - (28), and then updates the kernel hyperparameters optimizing the objective in Eq. (29).

Algorithm 1: Training procedure of ψ -VGPMIL.

Input: Training instances $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, bag indices $\{\text{Bag}_1, \dots, \text{Bag}_B\}$, bag labels $\mathbf{T} = \{T_1, \dots, T_B\}$, number of inducing points, number of iterations K , kernel hyperparameters λ , function ψ .

- 1 Initialize the locations of the inducing points \mathbf{Z} . Compute the kernel matrices $\mathbf{K}_{\mathbf{X}\mathbf{X}}, \mathbf{K}_{\mathbf{X}\mathbf{Z}}, \mathbf{K}_{\mathbf{Z}\mathbf{X}}, \mathbf{K}_{\mathbf{Z}\mathbf{Z}}, \mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1}, \tilde{\mathbf{K}}$ using λ .
- 2 Initialize the components of \mathbf{m} and \mathbf{S} to random values drawn from $\mathcal{N}(0, 1)$. Initialize the components of $\boldsymbol{\pi}$ to random values drawn from Uniform(0, 1).
- 3 **for** $k = 1, \dots, K$ **do**
- 4 Update $\boldsymbol{\Theta}$ using Eq. (28).
- 5 Update \mathbf{S} using Eq. (27).
- 6 Update \mathbf{m} using Eq. (26).
- 7 For each $n \in \{1, \dots, N\}$, update π_n using Eq. (25).
- 8 Estimate $\hat{\lambda} = \arg \max_{\lambda} \mathcal{J}(\lambda)$, with $\mathcal{J}(\lambda)$ given in Eq. (29). Recompute the kernel matrices using $\hat{\lambda}$.

Output: Kernel hyperparameters $\hat{\lambda}$, distributions $q(\mathbf{u})$ and $q(\mathbf{y}) = \prod_{n=1}^N q(y_n)$ as in Eq. (23) and (24).

Making predictions in ψ -VGPMIL. Given a new bag $\{\mathbf{x}_1^*, \dots, \mathbf{x}_{N_*}^*\}$, we are interested in both instance and bag level predictions, which are denoted by $\mathbf{y}^* = \{y_1^*, \dots, y_{N_*}^*\}$ and T^* , respectively. To predict the latent function value f^* of an instance \mathbf{x}^* , we substitute the approximate posterior into the predictive distribution,

$$p(f^* | \mathbf{T}) \approx \mathcal{N}(f^* | \boldsymbol{\mu}^*, \boldsymbol{\sigma}^{*2}), \quad (30)$$

where $\boldsymbol{\mu}^* = \mathbf{K}_{\mathbf{x}^*\mathbf{Z}}\mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1}\mathbf{m}$ and $\boldsymbol{\sigma}^{*2} = \mathbf{K}_{\mathbf{x}^*\mathbf{x}^*} + \mathbf{K}_{\mathbf{x}^*\mathbf{Z}}\mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1}(\mathbf{S}\mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1} - \mathbf{I})\mathbf{K}_{\mathbf{Z}\mathbf{x}^*}$. To compute the distribution of the test label y^* , we integrate with respect to f^* in $p(y^*, f^* | \mathbf{T})$ to obtain

$$p(y^* = 1 | \mathbf{T}) = \int_{-\infty}^{+\infty} p(f^* | \mathbf{T}) \text{logit}(f^*) df^* = \mathbb{E}_{p(f^* | \mathbf{T})} [\text{logit}(f^*)], \quad (31)$$

which allows us to make instance-level predictions having trained the model exclusively with bag labels. As the expectation cannot be calculated in closed form, we approximate it by sampling from the GP predictive distribution,

$$p(y^* = 1 | \mathbf{T}) \approx \hat{p} = \frac{1}{L} \sum_{l=1}^L \text{logit}(f_l^*), \quad (32)$$

$$\text{Var}_{p(f^* | \mathbf{T})} [\text{logit}(f^*)] \approx \frac{1}{L} \sum_{l=1}^L (\text{logit}(f_l^*) - \hat{p})^2, \quad (33)$$

where $\{f_1^*, \dots, f_L^*\} \sim p(f^* | \mathbf{T})$. To obtain the bag label, we apply the MIL hypothesis,

$$p(T^* = 1 | \mathbf{T}) = 1 - \prod_{n=1}^{N_*} p(y_n^* = 0 | \mathbf{T}) = \mathbb{E}_{v(\mathbf{f}^* | \mathbf{T})} \left[1 - \prod_{n=1}^{N_*} (1 - \text{logit}(f_n^*)) \right], \quad (34)$$

where $v(\mathbf{f}^* | \mathbf{T}) = \prod_{n=1}^{N_*} p(f_n^* | \mathbf{T})$. Again, we estimate the above expectation using samples from the GP predictive distribution,

$$p(T^* = 1 | \mathbf{T}) \approx \tilde{p} = \frac{1}{L} \sum_{l=1}^L \left(1 - \prod_{n=1}^{N_*} (1 - \text{logit}(f_{nl}^*)) \right), \quad (35)$$

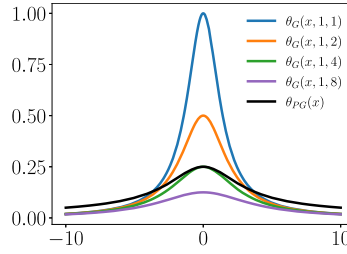


Fig. 1. Plot of the functions $\theta_{PG}(x) = \tanh(x/2)/(2x)$ (black) and $\theta_G(x; \alpha, \beta) = \alpha/(\beta + x^2/2)$ for $\alpha = 1$ and $\beta \in \{1, 2, 4, 8\}$ (blue, orange, green, purple).

Table 1

Label distribution at the instance and bag level for each dataset considered.

	Num. instances	Positive instances	Negative instances	Num. bags	Positive bags	Negative bags
MNIST	70000	8774	61226	7000	3500	3500
MUSK1	476	N/A	N/A	92	47	45
MUSK2	6598	N/A	N/A	102	39	63
RSNA	39750	5782	33968	1150	483	667
CQ500	193317	N/A	N/A	490	205	285

$$\text{Var}_{v(f^*|\mathbf{T})} \left[1 - \prod_{n=1}^{N_*} (1 - \text{logit}(f_n^*)) \right] \approx \frac{1}{L} \sum_{l=1}^L \left(\tilde{p} - \left(1 - \prod_{n=1}^{N_*} (1 - \text{logit}(f_{nl}^*)) \right) \right)^2, \quad (36)$$

where $\{f_{n1}^*, \dots, f_{nL}^*\} \sim p(f_n^* | \mathbf{T})$ for each $n \in \{1, \dots, N_*\}$.

5. Gamma variables for GP-MIL

As explained in the previous section, new models that generalize VGPMIL can be obtained by replacing the Hyperbolic Secant density with a different density that also admits a GSM representation. In this section, we focus on a concrete realization of ψ -VGPMIL, which uses Gamma variables and will prove to work better in practice.

Our motivation to use Gamma variables is as follows. A Pólya-Gamma variable $\omega \sim \text{PG}(1, 0)$ is defined as an infinite weighted sum of independent Gamma variables [20, Definition 1], $\omega = (2\pi^2)^{-1} \sum_{m=1}^{\infty} g_m / (m - 0.5)^2$, where $g_m \sim \text{Gamma}(1, 1)$. We alternatively consider $g_m = g \sim \text{Gamma}(1, 1)$ for every m , which leads to $\omega = g/4$ and, therefore, $\omega \sim \text{Gamma}(1, 4)$. Thus, we replace the Pólya-Gamma density in Eq. (16) by a Gamma density with parameters α and β , obtaining

$$\psi(x) = Z(\alpha, \beta)^{-1} (\beta + x^2/2)^{-\alpha}, \quad \forall x > 0, \quad (37)$$

where $Z(\alpha, \beta)$ is the normalization constant. It is worth mentioning that although this constant can be calculated, it is not needed to carry out the updates in Algorithm 1. Thus, we arrive at a concrete realization of ψ -VGPMIL, which we call *Gamma Variational Gaussian Processes Multiple Instance Learning (G-VGPMIL)*.

We note that G-VGPMIL differs from PG-VGPMIL in the function θ used to compute the Θ matrix. In PG-VGPMIL this function is defined as $\theta_{PG}(x) = \tanh(x/2)/(2x)$, while in G-VGPMIL we obtain $\theta_G(x; \alpha, \beta) = \alpha/(\beta + x^2/2)$. Both functions are plotted in Fig. 1 for $\alpha = 1$ and different values of β . We leave the estimation of α and β for future work.

6. Results

In this section, we carry out an empirical validation of the newly formulated G-VGPMIL model by means of three different experiments. Table 1 summarizes the five datasets that we use. First, using an MIL version of the well-known MNIST dataset, we perform a controlled and visual experiment to understand the behavior of our method at both the instance and bag levels. Second, we employ two classical benchmark datasets for MIL algorithms, the MUSK1 and MUSK2 datasets. Third, we tackle the real-world medical problem of ICH detection, showing enhanced performance against state-of-the-art approaches. We compare VGPMIL (Algorithm 1 with $\psi = \phi$) and G-VGPMIL (Algorithm 1 with ψ as in Eq. (37)). To ensure a fair comparison, both models were trained with identical set-ups, initial parameters, and grid-searches in every experiment. Both models have been implemented in JAX,¹ and will be available at <https://github.com/Franblueee/psi-VGPMIL> upon the acceptance of the paper.

¹ <https://jax.readthedocs.io/en/latest/>.

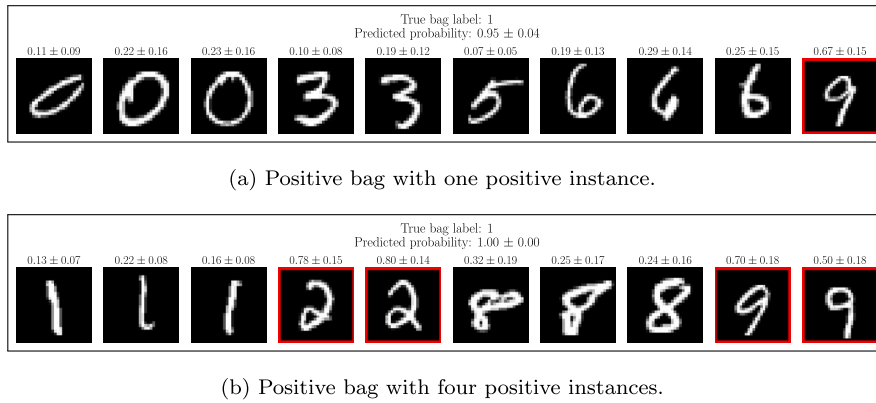


Fig. 2. MNIST bags and G-VGPML predictions (probability to be positive). Positive instances are highlighted with a red frame.

6.1. MNIST

Overview. In this section we analyze the behavior of the newly proposed G-VGPML in a controlled environment. To this end, we transform the MNIST dataset [38] into an MIL one. This experiment allows us to understand the performance of our method at both instance and bag level. The results show that G-VGPML performs, at least, as well as the state-of-the-art VGPML [14] while taking less time to complete the training.

Dataset description. The MNIST dataset consists of 70000 images (60000 for training and 10000 for testing) of handwritten digits. We choose the digits 2 and 9 to be the positive class, while the rest of the digits belong to the negative class. This way, a bag is positive if it contains at least a 2 or a 9, and negative otherwise. We randomly group the digits into bags of 10 instances each, ensuring that there is a balanced distribution of positive and negative bags, with each positive bag containing 1 to 4 positive instances. The resulting dataset has 61226 positive instances, 8774 negative instances and 7000 bags. Fig. 2 shows two of the generated bags, as well as the predictions obtained by G-VGPML. We also show the standard deviation for each prediction, which was calculated from equations (33) and (36).

We consider two versions of this dataset. In the first one, called MNIST RAW, we use all the $28 \times 28 = 784$ features of the instances without any kind of preprocessing. Given that shallow models may perform poorly when dealing with high-dimensional data, in the second version, i.e., MNIST PCA, we apply Principal Component Analysis (PCA) and retain the first 30 principal components of each instance.

Experimental details. For both VGPML and G-VGPML models, we fix $H = 100$ and choose the RBF kernel. The initial kernel hyperparameters are $(\nu, l) = (0.5, 30)$ for MNIST PCA and $(\nu, l) = (0.5, 784)$ for MNIST RAW. We conduct a grid search considering $\{50, 100, 200\}$ for the number of inducing points, $\{0.5, 1.0\}$ for the α hyperparameter and $\{1.0, 2.5, 4.0\}$ for the β hyperparameter. We create five different train-test stratified splits in which we evaluate all previous configurations. To guide training, we monitor the Bag AUC score in a validation subset and halt the process if no improvement occurs during ten epochs (an epoch is a complete update of the variational parameters). Based on the Bag Accuracy score, we select the best model and report the corresponding cross-validation metrics.

Result 1: competitive performance and training time. In Fig. 3 we have selected, for each number of inducing points, the best model according to the Bag Accuracy score, and plotted it against the training time in seconds. VGPML and G-VGPML exhibit similar training times. They perform comparably in MNIST PCA, while G-VGPML is superior in MNIST RAW, suggesting enhanced performance in handling high-dimensional data. Overall, applying PCA improves the performance as it eliminates the existing redundancy in MNIST. Notice that, as theoretically expected, both the training time and the model performance increase with the number of inducing points.

Result 2: instance level results and other metrics. In the previous result, we focused on the Bag AUC metric. Now, we analyze the AUC, Accuracy and F1-score performance at bag and instance levels. This is especially important given the high imbalance of the instance labels (61226 negatives vs 8774 positives, recall Table 1). In Fig. 4 we collect the metrics corresponding to the best model in each scenario. These metrics are also available in Tables 4 and 5. In MNIST PCA there are few differences between VGPML and G-VGPML, while in MNIST RAW the differences are significantly higher in favor of G-VGPML, at both the instance and bag level.

6.2. MUSK (1 and 2)

Overview. In this section, we evaluate G-VGPML in two historically important MIL benchmark datasets. Notice that these datasets come from a real problem (see next paragraph). Yet, we show that G-VGPML performs similarly to the previous experiment, achieving once again better performance in a similar training time as VGPML.

Dataset description. MUSK1 and MUSK2 [2] are two tabular datasets that belong to the domain of drug activity prediction. The goal of this problem is to predict if a molecule will bind to a target binding site (a cavity into which the molecule fits). This is determined by the shapes that the molecule can adopt by rotating its bonds. If at least one of these shapes actually binds to the

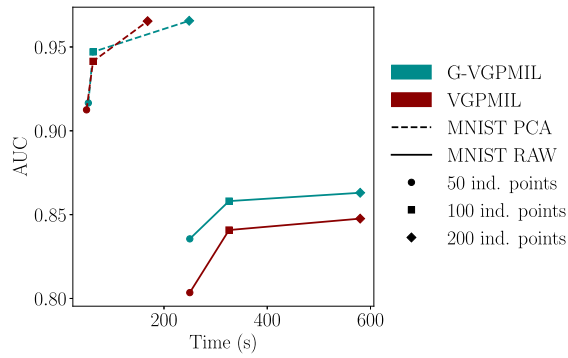


Fig. 3. Training time vs AUC in the MNIST dataset.

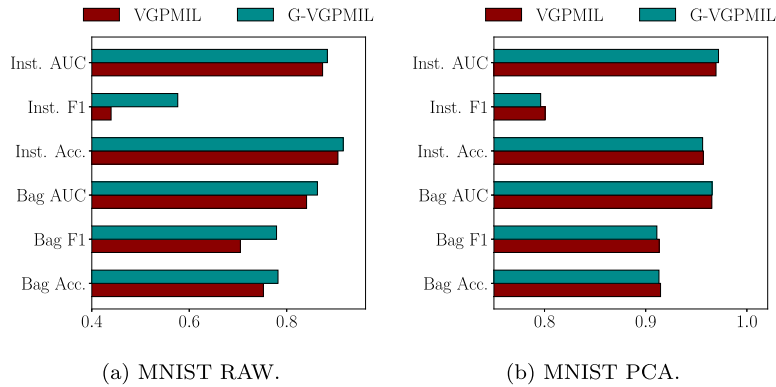


Fig. 4. Bag level and instance level performance in MNIST.

binding site, the molecule is labeled as positive. Otherwise, it is labeled as negative. This way, a bag represents a whole molecule, while an instance is each of the shapes that the molecule can adopt. MUSK1 has 476 instances and 92 bags (47 positive and 45 negative). MUSK2 has 6598 instances and 102 bags (39 positive and 63 negative). Instances are represented by a 166-dimensional real vector and their labels are unknown, so we will only analyze the metrics at the bag level.

Experimental details. Again, we fix $H = 100$ and choose the RBF kernel. The initial kernel hyperparameters are $(\nu, l) = (0.5, 166)$. We perform a grid search considering $\{50, 100, 200\}$ for the number of inducing points, $\{0.5, 1.0\}$ for the α hyperparameter and $\{1.0, 2.5, 4.0\}$ for the β hyperparameter. We create five different train-test stratified splits in which we evaluate each of the previous configurations. To guide the training process we monitor the AUC metric in a validation subset and halt the training when it has not improved for ten epochs (an epoch is a complete update of the variational parameters). Based on the Bag Accuracy score, we select the best model and report the corresponding cross-validation metrics.

Result 1: better performance in similar training time. In Fig. 5 we show a plot analogous to that of the MNIST experiment. We have selected the best model (according to the Bag AUC metric) for each number of inducing points considered, and have plotted it against the training time in seconds. For easier comparison, MUSK1 and MUSK2 results are shown together. Observe that the blue dots (G-VGPML) are above the red ones (VGPMIL), which means that our method shows superior performance. In terms of the training time, G-VGPML is clearly faster in MUSK1, while it is competitive with VGPMIL in MUSK2.

Result 2: other bag level metrics. Remember that in the MUSK1 and MUSK2 datasets we do not have the instance level labels (recall Table 1), so we cannot assess the performance at this level. However, we analyze other bag level metrics to ensure that our method performs correctly in all aspects. For MUSK2 the F1 metric is very important given the imbalance between positive bags and negative bags (see Table 1). Fig. 6 shows the metrics corresponding to the best model for each of the datasets (also available in Tables 6 and 7). Observe that G-VGPML attains better values than VGPMIL, especially in MUSK2. The superiority of G-VGPML is consistent with the behavior we observed in MNIST: our approach performs better when dealing with high-dimensional data, which is exactly the case here.

6.3. Intracranial hemorrhage detection (RSNA, CQ500)

Overview. So far, we have focused on understanding the behavior of G-VGPML (we have compared only with VGPMIL) and we have worked with two relatively easy problems.

Here, we consider a more complex real-world medical problem of detecting ICH from brain CT scans, for which we need to perform a more complex feature extraction process, which will be explained below. Moreover, we will provide a wider comparison

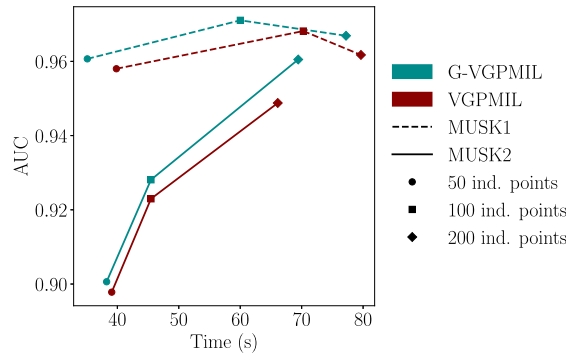


Fig. 5. Training time vs AUC for MUSK1 and MUSK2.

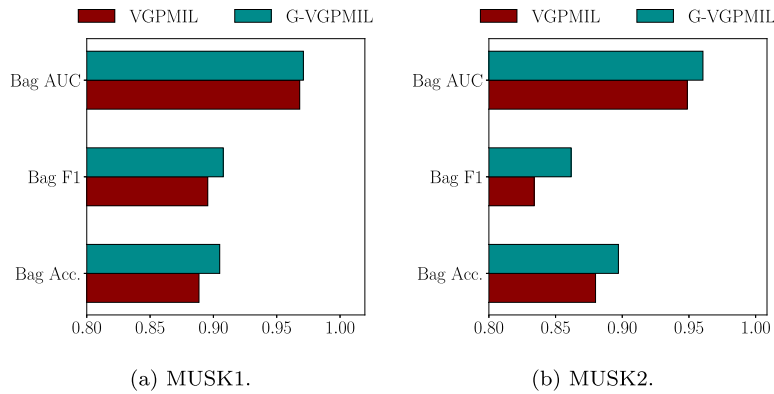


Fig. 6. Bag level metrics for MUSK1 and MUSK2.

against various state-of-the-art approaches that have been used in this same task. In summary, we will see that G-VGPML achieves very good results in predictive performance, efficiency and stability.

Dataset description. We will use two datasets to train and evaluate our method. The first one was published by the Radiological Society of North America (RSNA)² in 2019. It includes a total of 39750 slices from 1150 patients. The slice (instance) labels are known: there are 5782 abnormal slices (positive) and 33968 normal slices (negative). Regarding the scans (bags), there are 483 abnormal scans (positive) and 667 normal scans. The slices are of size 512×512 and the number of them in each scan varies from 24 to 57. The results for this dataset are shown in Subsection 6.3.1. We will also use the CQ500 dataset [6], which was acquired from different institutions in New Delhi, India, as an independent test set. It includes 193317 slices and 491 bags, with only labels at bag level (205 abnormal and 286 normal). The number of slices in each scan varies from 16 to 128. The results for this dataset are shown in Subsection 6.3.2.

Preprocessing. We follow the same approach as in [15]. To imitate the way radiologists read CT images we apply three windows to each CT slice to enhance the display of the brain, blood, and soft tissue. These three windows are concatenated to form a three-dimensional matrix and normalized to $[0, 1]$ (see Fig. 7).

Feature extraction for the ICH problem. In Sections 2, 3 and 5 we have discussed different GP-based MIL methods, including our novel G-VGPML. Although these methods can be directly applied to tabular data, in the case of medical images (and more in general, high dimensional or highly structured data), it is convenient to extract a few meaningful features as a previous step. For the ICH detection problem we use an attention-based CNN that has been employed in previous work [15,17]. The interested reader may consult [15] for details. The architecture is named AttCNN and corresponds to the composition of three functions,

$$\text{AttCNN}(\mathbf{X}^b) = (f_c \circ f_{\text{Att}} \circ f_{\text{CNN}})(\mathbf{X}^b), \quad (38)$$

where $\mathbf{X}^b \in \mathbb{R}^{N_b \times 3HW}$ is a matrix where the preprocessed instances of bag b are collected. First, a CNN $f_{\text{CNN}} : \mathbb{R}^{3HW} \rightarrow \mathbb{R}^D$ is applied to each instance in the bag to obtain the latent D dimensional representation of each instance. In this work, we have considered the values 8, 32, and 128 for D (the number of features extracted). Then, an attention layer $f_{\text{Att}} : \mathbb{R}^{N_b \times D} \rightarrow \mathbb{R}^D$ computes an aggregated representation of the bag. Lastly, a fully connected layer with a sigmoid output, $f_c : \mathbb{R}^D \rightarrow [0, 1]$, is applied to obtain bag

² <https://www.kaggle.com/competitions/rsna-intracranial-hemorrhage-detection/>.

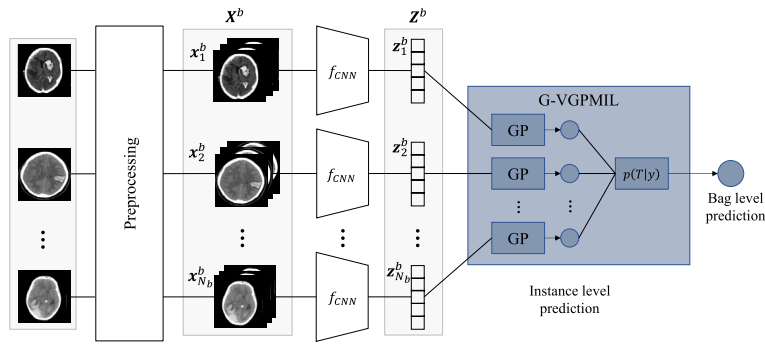


Fig. 7. Attention based CNN (adapted from [15]).

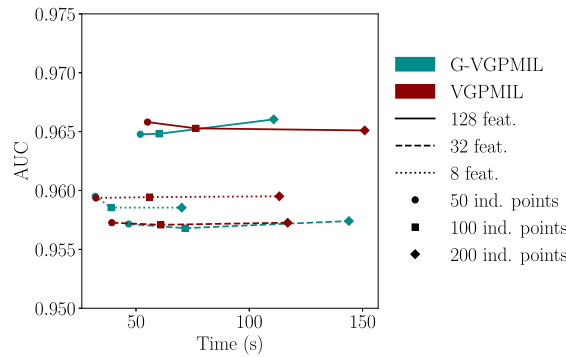


Fig. 8. Training time vs AUC in the RSNA dataset.

level predictions of $p(T_b = 1 | \mathbf{X}^b)$. AttCNN is trained to minimize the cross-entropy between its predictions and the true bag labels. Following [15,17], after the training procedure has been completed, we remove f_{Att} and f_c and replace them with our G-VGPMIL model, obtaining the architecture represented in Fig. 7.

Experimental details. We fix $H = 100$ and choose the RBF kernel. The initial kernel hyperparameters are $(\nu, l) = (0.5, D)$, where $D \in \{8, 32, 128\}$ is the number of features extracted. We perform a grid search considering $\{50, 100, 200\}$ for the number of inducing points, $\{0.5, 1.0\}$ for the α hyperparameter and $\{1.0, 2.5, 4.0\}$ for the β hyperparameter. We use the same five train-test splits that were used in [15] and [17]. To guide the training process we monitor the AUC metric in a validation subset and halt the training when it has not improved for ten epochs (an epoch is a complete update of the variational parameters). Based on the Bag Accuracy score, we select the best model and report the corresponding cross-validation metrics.

6.3.1. Training and testing on RSNA

As we have already mentioned, we first extract relevant features using AttCNN for $D \in \{8, 32, 128\}$. With these features, we train the different configurations of VGPMIL and G-VGPMIL. In this section, we report and analyze the results when testing in the RSNA test splits.

Result 1: competitive performance and training time. Fig. 8 show an AUC vs time plot analogous to those discussed in previous experiments. For each value of the number of features (D) and inducing points, we have selected the best model according to Bag AUC. In term of the bag AUC score, G-VGPMIL and VGPMIL demonstrate comparable performance, being VGPMIL slightly ahead when $D = 8$. As for the training time, significant differences are present when $D \in \{32, 128\}$, with G-VGPMIL showing lower training times retaining similar discriminative performance.

Result 2: robustness to other metrics. Once we have examined the behavior of our method using the AUC metric, we now focus on other classification metrics. As before, the metrics corresponding to the best model of each type (according to the Bag AUC metric) are collected in Fig. 9. Notice that we are also considering AttCNN as a baseline. As expected, AttCNN performs much worst than the probabilistic solutions based on GPs. Clearly, VGPMIL is the most effective method, being closely followed by the proposed G-VGPMIL. This is also true for each of the values of D we have considered (see Table 8).

Result 3: an example of a scan prediction. Fig. 10 shows an example of how our method predicts a positive scan from RSNA. As in Fig. 2, the standard deviation is calculated via sampling. Our method assigns a high probability to almost all positive instances. When the presence of hemorrhage is clear, uncertainty levels (standard deviation) are very low. In slices where there is hardly any visible hemorrhage, the uncertainty is higher. Also, as a considerable number of slices are predicted as positive, uncertainty at the bag level is almost zero. Note that, in addition to the prediction, the associated uncertainty is a very important piece of information for the user. This exemplifies the type of information that our model can provide to radiologists.

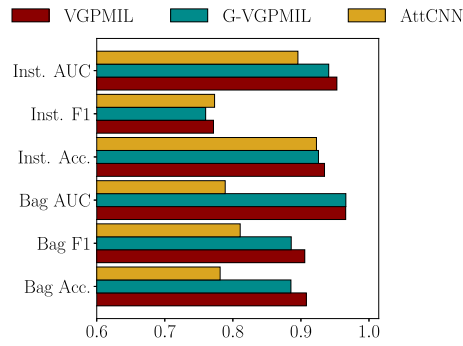


Fig. 9. Results obtained by the best model of each type in the RSNA dataset.

True scan label: 1
Predicted probability: 1.00 ± 0.00

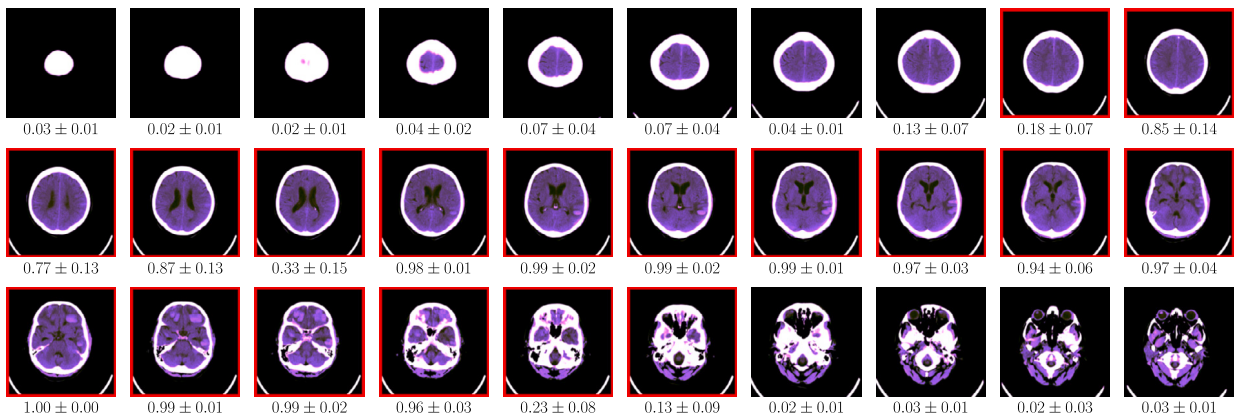


Fig. 10. RSNA scan and G-VGPMIL predictions (probability to be positive). Positive slices are highlighted with a red frame.

Table 2

Comparison of different approaches for binary ICH detection. VGPMIL, DGPML and G-VGPMIL results are obtained using the RSNA dataset for training and testing.

Source	Dataset size	Labeling type	Method	Bag AUC
Saab et al. [40]	4340 scans	Scan	MIL	0.91
Jnawali et al. [41]	40357 scans	Scan	3D CNNs	0.87
Titano et al. [42]	37236 scans	Scan	3D CNNs	0.88
Sato et al. [39]	126 scans	Scan	3D Autoencoder	0.87
Arbabshirani et al. [43]	45583 scans	Scan	3D CNNs	0.85
VGPMIL (Wu et al. [15])	1150 scans	Scan	MIL	0.9644 ± 0.0086
DGPML2 (López-Pérez et al. [17])	1150 scans	Scan	MIL	0.957
G-VGPMIL	1150 scans	Scan	MIL	0.966 ± 0.0065

Result 4: comparison with the state of the art. We complete the analysis of the RSNA dataset by comparing our method to other approaches in the literature. In Table 2 we collect information about other methods in the literature: the size of the dataset they use, a brief description of the method, and the Bag AUC obtained when evaluating in their test set. In our case, we report the metrics obtained when evaluating RSNA. Our method outperforms the others although it uses very few scans (only [39] uses less) and a relatively simple architecture (compared to 3D CNNs and Autoencoders). Also, observe that GP-based models (VGPMIL, DGPML, G-VGPMIL) obtain the best performance.

6.3.2. Evaluation on CQ500

To finish, we evaluate our recently trained models using an external database called CQ500. We discuss how well our model generalizes to examples never seen before and of a different nature from the ones it was trained with.

Result 1: better performance with less training time. Fig. 11 shows a plot similar to that of RSNA, now focusing on the Bag AUC in the CQ500 dataset. G-VGPMIL is always above VGPMIL, which indicates better performance. Also, for $D \in \{8, 128\}$, G-VGPMIL appears to the left, which implies reduced training time.

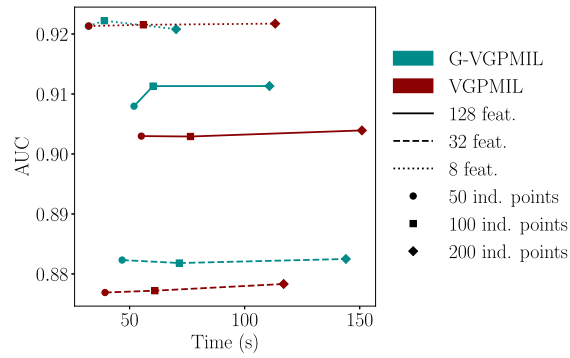


Fig. 11. Training time vs AUC in the CQ500 dataset.

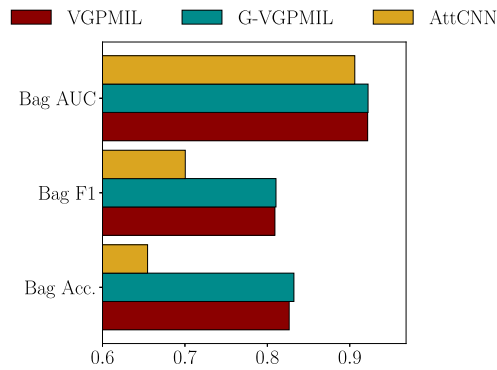


Fig. 12. Results obtained by the best model of each type in the CQ500 dataset.

Table 3

Performance of different approaches in the CQ500 dataset. VGPMIL, DGPML2 and G-VGPML use the RSNA dataset for training.

Source	Labeling type	Method	AUC
Chilamkurthy et al. [45]	Slice	2D CNNs	0.94
Nguyen et al. [46]	Slice	2D CNN + LSTM	0.96
Monteiro et al. [44]	Scan	voxel-based CNN	0.83
VGPMIL (Wu et al. [15])	Scan	MIL	0.9062 ± 0.007
DGPML2 (López-Pérez et al. [17])	Scan	MIL	0.909
G-VGPML	Scan	MIL	0.923 ± 0.0066

Result 2: other classifications metrics. As in the case of RSNA, we report other classification metrics at bag level in Table 9. For each value of D , G-VGPML obtains the best value. In general, the gap between G-VGPML and VGPMIL increases as the number of used features grows. Fig. 12 represents the metrics of the best model of each type (according to Bag AUC). G-VGPML and VGPMIL remain at a competitive performance (being G-VGPML one step ahead), while the top performance of AttCNN drops significantly compared to the behavior observed in RSNA. This suggests that the generalization ability of AttCNN is worst than that of the models built upon GPs.

Result 3: comparison with the state of the art. Finally, we compare our approach with other methods using the Bag AUC metric in the CQ500 dataset, see Table 3. Among the methods that use only bag labels to train the models [44,15,17], our method obtains the highest score. Also, it is highly competitive with those approaches that are trained using the slice labels [45,46]. Notice that using instance-level information turns this problem into a much easier one, but it requires additional effort from the radiologists. Our model is able to compete with those methods in a less demanding manner.

7. Conclusion

Motivated by the problems associated with the logistic observation model in the GP-based MIL formulation, we have reformulated VGPMIL using Pólya-Gamma random variables. We found that this new model, PG-VGPML, leads to the same update equations as VGPMIL. This is a consequence of the two equivalent representations that the Hyperbolic Secant density admits. This reveals that

VGPMIL/PG-VGPMIL is a realization of a more general framework, ψ -VGPMIL, which is achieved by replacing the Hyperbolic Secant density by a general GSM density ψ .

An interesting challenge that arises is the choice of a convenient density. In this work, we have explored the natural choice of the Gamma distribution, arriving at the newly proposed G-VGPMIL. Our experiments show that G-VGPMIL improves upon VGPMIL in terms of predictive performance and training time. G-VGPMIL shows competitive results with fully supervised models, thus closing the gap with them. As the features used to train G-VGPMIL may not be optimal (they have been extracted in two phases), we believe that there is still room for improvement.

Another interesting challenge is the extension of our model to a multi-class problem, where one has to deal with the intractable term that the softmax function introduces. Following the line of our work, we would need to express the partition function of the softmax as a GSM, which is not obvious and requires further investigation.

Finally, we believe that this work provides interesting ideas that can be useful beyond MIL and leaves open questions of both theoretical and practical nature. We hope that our ideas will be useful to the rest of the community and can enhance further research.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The used data is publicly available and the code will be published in GitHub upon acceptance of the paper.

Acknowledgements

This work was supported by project PID2019-105142RB-C22 funded by MCIN, by project PID2022-140189OB-C22 funded by MCIN / AEI / 10.13039 / 501100011033, and by project B-TIC-324-UGR20 funded by FEDER/Junta de Andalucía and Universidad de Granada. The work by Francisco M. Castro-Macías was supported by Ministerio de Universidades under FPU contract FPU21/01874. Pablo Morales-Álvarez acknowledges grant C-EXP-153-UGR23 funded by Consejería de Universidad, Investigación e Innovación and by ERDF Andalusia Program 2021-2027. The authors acknowledge Universidad de Granada / CBUA for funding the open access charge.

Appendix A. ψ -VGPMIL variational updates

In this section, we derive the variational updates for the ψ -VGPMIL model presented in Section 4. The updates for PG-VGPMIL, from Section 3, are obtained by setting ψ to the Hyperbolic Secant distribution defined by Eq. (15). Similarly, for G-VGPMIL, as discussed in Section 5, the updates are obtained by defining ψ as the density specified in Eq. (37).

To perform inference, we consider the augmented ψ -VGPMIL model. Recall that ψ is a GSM density, so there exists a mixing density $\hat{\psi} : [0, +\infty[\rightarrow \mathbb{R}$ such that,

$$\psi(x) = \int_0^{+\infty} \mathcal{N}(x | 0, \omega^{-1}) \hat{\psi}(\omega) d\omega, \quad \forall x \in \mathbb{R}. \quad (39)$$

The augmented ψ -VGPMIL model is given by

$$p(\mathbf{u}, \mathbf{f}, \mathbf{y}, \boldsymbol{\omega}, \mathbf{T}) = p(\mathbf{u}, \mathbf{f}, \mathbf{y}, \boldsymbol{\omega}) p(\mathbf{T} | \mathbf{y}), \quad (40)$$

$$p(\mathbf{u}, \mathbf{f}, \mathbf{y}, \boldsymbol{\omega}) = Z^{-1} \mathcal{N}(\mathbf{f}, \mathbf{u}) \exp((\mathbf{y} - 2^{-1} \mathbf{1})^T \mathbf{f}) \mathcal{N}(\mathbf{f} | \boldsymbol{\omega}) \hat{\psi}(\boldsymbol{\omega}), \quad (41)$$

where $Z = \pi^{-N} \int \mathcal{N}(\mathbf{f}, \mathbf{u}) \psi(\mathbf{f}) \phi(\mathbf{f})^{-1} d(\mathbf{f}, \mathbf{u})$, $\hat{\psi}(\boldsymbol{\omega}) = \prod_{n=1}^N \hat{\psi}(\omega_n)$, $\mathcal{N}(\mathbf{f} | \boldsymbol{\omega}) = \mathcal{N}(\mathbf{f} | \mathbf{0}, \boldsymbol{\Omega}^{-1})$, $\boldsymbol{\Omega} = \text{Diag}(\boldsymbol{\omega})$, and $p(\mathbf{T} | \mathbf{y})$ is the bag likelihood given by Eq. (4).

Following the variational inference approach, we approximate the posterior $p(\mathbf{u}, \mathbf{f}, \mathbf{y}, \boldsymbol{\omega} | \mathbf{T})$ with a variational distribution $q(\mathbf{u}, \mathbf{f}, \mathbf{y}, \boldsymbol{\omega}) = q(\mathbf{u}) \mathcal{N}(\mathbf{f} | \mathbf{u}) q(\mathbf{y}) q(\boldsymbol{\omega})$, where $q(\mathbf{y}) = \prod_{n=1}^N q(y_n)$ and $q(\boldsymbol{\omega}) = \prod_{n=1}^N q(\omega_n)$. We achieve this by minimizing the KL divergence between the posterior and the variational distributions. Using [27, Eq. (10.9)], we can compute the optimal solution for each distribution keeping the others fixed,

$$\log q = \mathbb{E}_{Q \setminus q} [\log p(\mathbf{T}, \mathbf{u}, \mathbf{f}, \mathbf{y}, \boldsymbol{\omega})] + \text{const}, \quad (42)$$

where Q denotes the joint variational distribution.

Update of $q(y_{j \neq n})$. We denote $q(y_{j \neq n}) = \prod_{j \neq n} \pi_j^{y_j} (1 - \pi_j)^{1-y_j}$. Let n be fixed and let b be the index of the bag to which instance n belongs. We have

$$\log q(y_n) = \underbrace{\mathbb{E}_{q(\mathbf{u})\mathcal{N}(\mathbf{f}|\mathbf{u})q(y_{j \neq n})} \left[(\mathbf{y} - 2^{-1}\mathbf{1})^\top \mathbf{f} \right]}_{A_1} + \underbrace{\mathbb{E}_{q(y_{j \neq n})} [\log p(\mathbf{T} | \mathbf{y})]}_{A_2} + \text{const.} \quad (43)$$

We analyze each term separately,

$$A_1 = \mathbb{E}_{q(y_{j \neq n})} [\mathbf{y}]^\top \mathbb{E}_{q(\mathbf{u})p(\mathbf{f}|\mathbf{u})} [\mathbf{f}] + \text{const} \quad (44)$$

$$= y_n \mathbb{E}_{q(\mathbf{u})p(\mathbf{f}|\mathbf{u})} [f_n] + \text{const} = \quad (45)$$

$$= y_n \mathbf{K}_{\mathbf{x}_i \mathbf{Z}} \mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1} \mathbf{m} + \text{const} \quad (46)$$

$$A_2 = \log H \mathbb{E}_{q(y_{j \neq n})} [G_b] + \text{const} = \quad (47)$$

$$= \log H (2T_b - 1) \mathbb{E}_{q(y_{j \neq n})} [\max \{\mathbf{y}\}_b] + \text{const} = \quad (48)$$

$$= y_n \log H (2T_b - 1) (1 - \mathbb{E} [\max \{\mathbf{y}\}_{b \setminus n}]) + \text{const}, \quad (49)$$

where we have used [14, Eq. 13] to compute the expected value involved. Putting it all together,

$$\log q(y_n) = y_n \left[\mathbf{K}_{\mathbf{x}_i \mathbf{Z}} \mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1} \mathbf{m} + \log H (2T_b - 1) (1 - \mathbb{E} [\max \{\mathbf{y}\}_{b \setminus n}]) \right] + \text{const}, \quad (50)$$

from which we deduce $q(y_n) = \text{Bernouilli}(y_n | \pi_n)$ with

$$\pi_n = \text{logit} \left[\mathbf{K}_{\mathbf{x}_i \mathbf{Z}} \mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1} \mathbf{m} + \log H (2T_b - 1) (1 - \mathbb{E} [\max \{\mathbf{y}\}_{b \setminus n}]) \right] \quad (51)$$

Update of $q(\mathbf{u})$.

$$\log q(\mathbf{u}) = \underbrace{\log \mathcal{N}(\mathbf{u})}_{B_1} + \underbrace{\mathbb{E}_{\mathcal{N}(\mathbf{f}|\mathbf{u})q(\mathbf{y})q(\omega)} \left[((\mathbf{y} - 2^{-1}\mathbf{1})^\top \mathbf{f}) + \log \mathcal{N}(\mathbf{f} | \omega) \right]}_{B_2} + \text{const.} \quad (52)$$

We analyze each term separately,

$$B_1 = -2^{-1} \mathbf{u}^\top \mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1} \mathbf{u} + \text{const}, \quad (53)$$

$$B_2 = \mathbb{E}_{\mathcal{N}(\mathbf{f}|\mathbf{u})q(\mathbf{y})q(\omega)} [(\mathbf{y} - 1/2)^\top \mathbf{f} - 2^{-1} \mathbf{f}^\top \mathbf{\Omega} \mathbf{f}] + \text{const} = \quad (54)$$

$$= \mathbb{E}_{\mathcal{N}(\mathbf{f}|\mathbf{u})q(\omega)} \left[(\boldsymbol{\pi} - 2^{-1}\mathbf{1})^\top \mathbf{f} - 2^{-1} \mathbf{f}^\top \mathbf{\Omega} \mathbf{f} \right] + \text{const} = \quad (55)$$

$$= (\boldsymbol{\pi} - 2^{-1}\mathbf{1})^\top \mathbf{K}_{\mathbf{X}\mathbf{Z}} \mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1} \mathbf{u} - 2^{-1} \mathbf{u}^\top \mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1} \mathbf{K}_{\mathbf{Z}\mathbf{X}} \mathbf{\Theta} \mathbf{K}_{\mathbf{X}\mathbf{Z}} \mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1} \mathbf{u} + \text{const}, \quad (56)$$

where $\mathbf{\Theta} = \text{Diag}(\theta_1, \dots, \theta_N)$ and $\theta_n = \mathbb{E}_{q(\omega_n)} [\omega_n]$. Putting all together

$$\log q(\mathbf{u}) = (\boldsymbol{\pi} - 2^{-1}\mathbf{1})^\top \mathbf{K}_{\mathbf{X}\mathbf{Z}} \mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1} \mathbf{u} - 2^{-1} \mathbf{u}^\top (\mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1} \mathbf{K}_{\mathbf{Z}\mathbf{X}} \mathbf{\Theta} \mathbf{K}_{\mathbf{X}\mathbf{Z}} \mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1} + \mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1}) \mathbf{u} + \text{const}. \quad (57)$$

This proves that $q(\mathbf{u}) = \mathcal{N}(\mathbf{u} | \mathbf{m}, \mathbf{S})$ with

$$\mathbf{m} = \mathbf{S} \mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1} \mathbf{K}_{\mathbf{Z}\mathbf{X}} (\boldsymbol{\pi} - 2^{-1}\mathbf{1}) \quad (58)$$

$$\mathbf{S} = (\mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1} \mathbf{K}_{\mathbf{Z}\mathbf{X}} \mathbf{\Theta} \mathbf{K}_{\mathbf{X}\mathbf{Z}} \mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1} + \mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1})^{-1} \quad (59)$$

Update of $q(\omega_n)$. Let n be fixed.

$$\log q(\omega_n) = \mathbb{E}_{q(f_n)} [\log \mathcal{N}(f_n | 0, \omega_n^{-1})] + \log \hat{\psi}(\omega_n) = \quad (60)$$

$$= \log \mathcal{N}(c_n | 0, \omega_n^{-1}) + \log \hat{\psi}(\omega_n), \quad (61)$$

where $c_n = \sqrt{\mathbb{E}_{q(f_n)} [f_n^2]}$. Thus, if $\psi(c_n) \neq 0$,

$$q(\omega_n) = \psi(c_n)^{-1} \mathcal{N}(c_n | 0, \omega_n^{-1}) \hat{\psi}(\omega_n). \quad (62)$$

In the PG-VGPMIL scenario, we can deduce $q(\omega_n) = \text{PG}(\omega_n | 1, c_n)$ using that the general Pólya-Gamma density $\text{PG}(\cdot | b, c)$ is a exponential tilting of the $\text{PG}(\cdot | b, 0)$ density [20]. However, in the general case, we only need the expectation of $q(\omega_n)$. To calculate it, we first differentiate under the integral sign in Eq. (39),

$$\psi'(x) = -x \int_0^{+\infty} \omega \mathcal{N}(x | 0, \omega^{-1}) \hat{\psi}(\omega) d\omega. \quad (63)$$

Then, we evaluate this expression in $x = c_n$ and use Eq. (62),

Table 4
MNIST PCA.

Num. ind. points	Model	Inst. Accuracy	Inst. F1	Inst. AUC	Bag Accuracy	Bag F1	Bag AUC
50	G-VGPML	0.9342 ± 0.004	0.6769 ± 0.0305	0.9232 ± 0.0056	0.8481 ± 0.0153	0.843 ± 0.018	0.9166 ± 0.0136
	VGPML	0.9296 ± 0.0013	0.6352 ± 0.0083	0.9246 ± 0.0071	0.8438 ± 0.011	0.8327 ± 0.0117	0.9125 ± 0.0098
100	G-VGPML	0.9431 ± 0.0029	0.7183 ± 0.0178	0.9559 ± 0.0032	0.8841 ± 0.0139	0.878 ± 0.0151	0.947 ± 0.0069
	VGPML	0.9425 ± 0.0033	0.7171 ± 0.0209	0.95 ± 0.0032	0.8756 ± 0.0188	0.8699 ± 0.0203	0.9414 ± 0.0071
200	G-VGPML	0.9563 ± 0.0023	0.7962 ± 0.0137	0.972 ± 0.0009	0.9131 ± 0.0132	0.911 ± 0.014	0.9656 ± 0.003
	VGPML	0.957 ± 0.004	0.8006 ± 0.0247	0.9695 ± 0.002	0.9147 ± 0.0137	0.9133 ± 0.0148	0.9654 ± 0.0056

Table 5
MNIST RAW.

Num. ind. points	Model	Inst. Accuracy	Inst. F1	Inst. AUC	Bag Accuracy	Bag F1	Bag AUC
50	G-VGPML	0.9016 ± 0.0036	0.5365 ± 0.0448	0.8703 ± 0.015	0.7384 ± 0.0193	0.75 ± 0.0274	0.8355 ± 0.0234
	VGPML	0.8961 ± 0.0039	0.3636 ± 0.0346	0.8549 ± 0.0094	0.7016 ± 0.0189	0.6311 ± 0.0301	0.8035 ± 0.0168
100	G-VGPML	0.9113 ± 0.0018	0.5681 ± 0.0147	0.8827 ± 0.0056	0.7653 ± 0.0147	0.771 ± 0.0141	0.8581 ± 0.0114
	VGPML	0.9052 ± 0.0016	0.4397 ± 0.0224	0.8737 ± 0.0059	0.7522 ± 0.0074	0.705 ± 0.0154	0.8408 ± 0.0079
200	G-VGPML	0.916 ± 0.0024	0.5764 ± 0.0241	0.8835 ± 0.0096	0.7822 ± 0.0114	0.7792 ± 0.016	0.863 ± 0.0152
	VGPML	0.9051 ± 0.0016	0.4338 ± 0.0107	0.876 ± 0.0065	0.7491 ± 0.0075	0.6986 ± 0.0099	0.8476 ± 0.0082

Table 6
MUSK1.

Num. ind. points	Model	Bag Accuracy	Bag F1	Bag AUC
50	G-VGPML	0.8832 ± 0.0202	0.8917 ± 0.0182	0.9607 ± 0.0133
	VGPML	0.8832 ± 0.018	0.8916 ± 0.0175	0.958 ± 0.0136
100	G-VGPML	0.905 ± 0.0224	0.9078 ± 0.0209	0.9711 ± 0.016
	VGPML	0.8886 ± 0.0178	0.8956 ± 0.0171	0.9682 ± 0.0137
200	G-VGPML	0.9023 ± 0.0258	0.908 ± 0.0233	0.9669 ± 0.0159
	VGPML	0.8806 ± 0.0271	0.89 ± 0.0218	0.9617 ± 0.0141

Table 7
MUSK2.

Num. ind. points	Model	Bag Accuracy	Bag F1	Bag AUC
50	G-VGPML	0.8308 ± 0.0298	0.7806 ± 0.0382	0.9006 ± 0.0247
	VGPML	0.8358 ± 0.0412	0.779 ± 0.0609	0.8978 ± 0.0219
100	G-VGPML	0.8603 ± 0.021	0.8228 ± 0.0174	0.9281 ± 0.0228
	VGPML	0.853 ± 0.0316	0.797 ± 0.0492	0.923 ± 0.0303
200	G-VGPML	0.8971 ± 0.0293	0.8617 ± 0.0385	0.9605 ± 0.01
	VGPML	0.88 ± 0.0234	0.834 ± 0.0365	0.9488 ± 0.0156

$$\psi'(c_n) = -c_n \psi(c_n) \mathbb{E}_{q(\omega_n)} [\omega_n], \quad (64)$$

from where we conclude that $\theta_n = -\psi'(c_n)/(c_n \psi(c_n))$.

Appendix B. Tables

The top value of each metric within each group is highlighted in bold, and the best in each column is underlined (Tables 4–9).

References

- [1] M.-A. Carboneau, V. Cheplygina, E. Granger, G. Gagnon, Multiple instance learning: a survey of problem characteristics and applications, *Pattern Recognit.* 77 (2018) 329–353.
- [2] T.G. Dietterich, R.H. Lathrop, T. Lozano-Pérez, Solving the multiple instance problem with axis-parallel rectangles, *Artif. Intell.* 89 (1–2) (1997) 31–71.
- [3] G. Quéllec, G. Cazuguel, B. Cochener, M. Lamard, Multiple-instance learning for medical image and video analysis, *IEEE Rev. Biomed. Eng.* 10 (2017) 213–234.
- [4] F.O. Otite, P. Khandelwal, A.M. Malik, S. Chaturvedi, R.L. Sacco, J.G. Romano, Ten-year temporal trends in medical complications after acute intracerebral hemorrhage in the United States, *Stroke* 48 (3) (2017) 596–603.
- [5] T.D. Phong, H.N. Duong, H.T. Nguyen, N.T. Trong, V.H. Nguyen, T. Van Hoa, V. Snasel, Brain hemorrhage diagnosis by using deep learning, in: *Proceedings of the 2017 International Conference on Machine Learning and Soft Computing*, 2017, pp. 34–39.
- [6] S. Chilamkurthy, R. Ghosh, S. Tanamala, M. Biviji, N.G. Campeau, V.K. Venugopal, V. Mahajan, P. Rao, P. Warier, Deep learning algorithms for detection of critical findings in head CT scans: a retrospective study, *Lancet* 392 (10162) (2018) 2388–2396.

Table 8
RSNA.

Num. features	Num. ind. points	Model	RSNA inst. Accuracy	RSNA inst. F1	RSNA inst. AUC	RSNA bag Accuracy	RSNA bag F1	RSNA bag AUC
8	50	G-VGPML	0.9395 ± 0.002	0.7791 ± 0.0125	0.9508 ± 0.0053	0.8907 ± 0.0172	0.8855 ± 0.0154	0.9595 ± 0.0081
		VGPMIL	0.941 ± 0.0016	0.7926 ± 0.0054	0.9592 ± 0.0034	0.884 ± 0.01	0.8823 ± 0.0095	0.9594 ± 0.0075
	100	G-VGPML	0.9399 ± 0.0018	0.7823 ± 0.0123	0.9434 ± 0.0045	0.8867 ± 0.0202	0.8818 ± 0.019	0.9585 ± 0.008
		VGPMIL	0.941 ± 0.0015	0.7903 ± 0.0049	0.9597 ± 0.0036	0.888 ± 0.016	0.8851 ± 0.0156	0.9594 ± 0.0076
	200	G-VGPML	0.9401 ± 0.0018	0.7835 ± 0.0109	0.9408 ± 0.0052	0.8867 ± 0.0174	0.882 ± 0.0164	0.9585 ± 0.0079
		VGPMIL	0.9411 ± 0.0013	0.7875 ± 0.0041	0.9603 ± 0.0037	0.8933 ± 0.0207	0.8883 ± 0.0211	0.9595 ± 0.0074
32	50	G-VGPML	0.9251 ± 0.0103	0.7554 ± 0.0307	0.9418 ± 0.0078	0.8613 ± 0.0229	0.8648 ± 0.0186	0.9572 ± 0.0044
		VGPMIL	0.9337 ± 0.008	0.7647 ± 0.0292	0.9523 ± 0.0054	0.8933 ± 0.0189	0.8904 ± 0.0176	0.9573 ± 0.0049
	100	G-VGPML	0.9254 ± 0.0105	0.7563 ± 0.0311	0.9425 ± 0.0075	0.86 ± 0.0276	0.8641 ± 0.0239	0.9568 ± 0.0053
		VGPMIL	0.9329 ± 0.008	0.76 ± 0.0292	0.9527 ± 0.0053	0.8907 ± 0.0182	0.8873 ± 0.017	0.9571 ± 0.0053
	200	G-VGPML	0.9266 ± 0.012	0.7562 ± 0.0344	0.9426 ± 0.0074	0.864 ± 0.0248	0.8657 ± 0.0204	0.9574 ± 0.005
		VGPMIL	0.9325 ± 0.0079	0.7568 ± 0.0287	0.9531 ± 0.0052	0.8893 ± 0.0187	0.8854 ± 0.018	0.9573 ± 0.0049
128	50	G-VGPML	0.9245 ± 0.0069	0.7586 ± 0.0162	0.9407 ± 0.0044	0.8773 ± 0.0124	0.8793 ± 0.0119	0.9648 ± 0.0073
		VGPMIL	0.9347 ± 0.007	0.7716 ± 0.0217	0.9529 ± 0.0047	0.908 ± 0.0154	0.9057 ± 0.0151	0.9658 ± 0.0067
	100	G-VGPML	0.9276 ± 0.0057	0.7509 ± 0.0178	0.9412 ± 0.0061	0.9067 ± 0.0112	0.9029 ± 0.0121	0.9648 ± 0.0075
		VGPMIL	0.9353 ± 0.007	0.7721 ± 0.0219	0.9532 ± 0.0046	0.912 ± 0.0129	0.9089 ± 0.013	0.9653 ± 0.0067
	200	G-VGPML	0.9258 ± 0.0068	0.7601 ± 0.0171	0.9409 ± 0.0047	0.8853 ± 0.0181	0.8858 ± 0.0168	0.9652 ± 0.0065
		VGPMIL	0.935 ± 0.0079	0.7683 ± 0.0254	0.954 ± 0.0047	0.9133 ± 0.0184	0.9097 ± 0.0186	0.9651 ± 0.0068

Table 9
CQ500.

Num. features	Num. ind. points	Model	CQ500 bag Accuracy	CQ500 bag F1	CQ500 bag AUC
8	50	G-VGPML	0.831 ± 0.0115	0.8115 ± 0.0108	0.9213 ± 0.0058
		VGPMIL	0.8041 ± 0.0131	0.7923 ± 0.0122	0.9213 ± 0.0063
	100	G-VGPML	0.8322 ± 0.0065	0.8105 ± 0.0073	0.923 ± 0.0066
		VGPMIL	0.811 ± 0.0117	0.7966 ± 0.0109	0.9215 ± 0.0062
	200	G-VGPML	0.831 ± 0.0093	0.8103 ± 0.0079	0.9208 ± 0.0058
		VGPMIL	0.8265 ± 0.0082	0.8092 ± 0.0077	0.9217 ± 0.0062
32	50	G-VGPML	0.8233 ± 0.0271	0.79 ± 0.0394	0.8823 ± 0.0284
		VGPMIL	0.8022 ± 0.0273	0.7752 ± 0.0377	0.8769 ± 0.0298
	100	G-VGPML	0.8245 ± 0.0263	0.7904 ± 0.0391	0.8818 ± 0.028
		VGPMIL	0.8063 ± 0.0284	0.7778 ± 0.0384	0.8772 ± 0.0294
	200	G-VGPML	0.829 ± 0.023	0.7927 ± 0.0385	0.8825 ± 0.0288
		VGPMIL	0.8127 ± 0.0309	0.7829 ± 0.0423	0.8783 ± 0.0293
128	50	G-VGPML	0.847 ± 0.021	0.8216 ± 0.0297	0.908 ± 0.0174
		VGPMIL	0.8221 ± 0.0279	0.8044 ± 0.0328	0.903 ± 0.018
	100	G-VGPML	0.8448 ± 0.017	0.8207 ± 0.0307	0.9113 ± 0.0185
		VGPMIL	0.829 ± 0.0237	0.8096 ± 0.03	0.9029 ± 0.0181
	200	G-VGPML	0.8485 ± 0.019	0.823 ± 0.0321	0.9113 ± 0.0185
		VGPMIL	0.8327 ± 0.0235	0.8114 ± 0.0306	0.9039 ± 0.018

- [7] J. Cho, K.-S. Park, M. Karki, E. Lee, S. Ko, J.K. Kim, D. Lee, J. Choe, J. Son, M. Kim, et al., Improving sensitivity on identification and delineation of intracranial hemorrhage lesion using cascaded deep learning models, *J. Digit. Imag.* 32 (3) (2019) 450–461.
- [8] K. Saab, J. Dunnmon, R. Goldman, A. Ratner, H. Sagreya, C. Ré, D. Rubin, Doubly weak supervision of deep learning models for head ct, in: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2019, pp. 811–819.
- [9] S. Remedios, Z. Wu, C. Bermudez, C.I. Kerley, S. Roy, M.B. Patel, J.A. Butman, B.A. Landman, D.L. Pham, Extracting 2D Weak Labels from Volume Labels Using Multiple Instance Learning in CT Hemorrhage Detection, *Medical Imaging 2020: Image Processing*, vol. 11313, SPIE, 2020, pp. 66–75.
- [10] S. Qi, C. Xu, C. Li, B. Tian, S. Xia, J. Ren, L. Yang, H. Wang, H. Yu, DR-MIL: deep represented multiple instance learning distinguishes COVID-19 from community-acquired pneumonia in CT images, *Comput. Methods Programs Biomed.* 211 (2021) 106406.
- [11] M. Ilse, J. Tomczak, M. Welling, Attention-based deep multiple instance learning, in: *International Conference on Machine Learning*, PMLR, 2018, pp. 2127–2136.
- [12] M. Kim, F.D.I. Torre, Gaussian processes multiple instance learning, in: *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 2010, pp. 535–542.
- [13] M. Kandemir, M. Haussmann, F. Diego, K.T. Rajamani, J. van der Laak, F.A. Hamprecht, Variational weakly supervised Gaussian processes, in: *Proceedings of the British Machine Vision Conference*, vol. 2016, 2016.
- [14] M. Haußmann, F.A. Hamprecht, M. Kandemir, Variational Bayesian multiple instance learning with Gaussian processes, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6570–6579.
- [15] Y. Wu, A. Schmidt, E. Hernández-Sánchez, R. Molina, A.K. Katsaggelos, Combining attention-based multiple instance learning and Gaussian processes for CT hemorrhage detection, in: *Medical Image Computing and Computer Assisted Intervention*, 2021, pp. 582–591.

- [16] F. Wang, A. Pinar, The multiple instance learning Gaussian process probit model, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 3034–3042.
- [17] M. López-Pérez, A. Schmidt, Y. Wu, R. Molina, A.K. Katsaggelos, Deep Gaussian processes for multiple instance learning: application to CT intracranial hemorrhage detection, *Comput. Methods Programs Biomed.* 219 (2022) 106783.
- [18] D. Durante, T. Rigon, Conditionally conjugate mean-field variational Bayes for logistic models, *Stat. Sci.* 34 (3) (2019) 472–485.
- [19] F. Wenzel, T. Galy-Fajou, C. Donner, M. Kloft, M. Opper, Efficient Gaussian process classification using Pólya-Gamma data augmentation, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 5417–5424.
- [20] N.G. Polson, J.G. Scott, J. Windle, Bayesian inference for logistic models using Pólya-Gamma latent variables, *J. Am. Stat. Assoc.* 108 (504) (2013) 1339–1349.
- [21] W. Harkness, M. Harkness, Generalized hyperbolic secant distributions, *J. Am. Stat. Assoc.* 63 (321) (1968) 329–337.
- [22] J.A. Palmer, Variational and scale mixture representations of non-Gaussian densities for estimation in the Bayesian linear model: Sparse coding, independent component analysis, and minimum entropy segmentation, Ph.D. thesis, 2006.
- [23] M.E. Tipping, Sparse Bayesian learning and the relevance vector machine, *J. Mach. Learn. Res.* 1 (Jun) (2001) 211–244.
- [24] L. Martino, J. Read, A joint introduction to Gaussian processes and relevance vector machines with connections to Kalman filtering and other kernel smoothers, *Inf. Fusion* 74 (2021) 17–38.
- [25] C.E. Rasmussen, C.K. Williams, et al., *Gaussian Processes for Machine Learning*, vol. 1, Springer, 2006.
- [26] E. Snelson, Z. Ghahramani, Sparse Gaussian processes using pseudo-inputs, *Adv. Neural Inf. Process. Syst.* 18 (2005).
- [27] C.M. Bishop, N.M. Nasrabadi, *Pattern Recognition and Machine Learning*, vol. 4, Springer, 2006.
- [28] T.S. Jaakkola, M.I. Jordan, Bayesian parameter estimation via variational methods, *Stat. Comput.* 10 (1) (2000) 25–37.
- [29] S. Linderman, M.J. Johnson, R.P. Adams, Dependent multinomial models made easy: stick-breaking with the Pólya-Gamma augmentation, *Adv. Neural Inf. Process. Syst.* 28 (2015).
- [30] T. Galy-Fajou, F. Wenzel, M. Opper, Automated augmented conjugate inference for non-conjugate Gaussian process models, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 3025–3035.
- [31] S.D. Babacan, R. Molina, M.N. Do, A.K. Katsaggelos, Bayesian blind deconvolution with general sparse image priors, in: European Conference on Computer Vision, Springer, 2012, pp. 341–355.
- [32] J. Zinn-Justin, *Quantum Field Theory and Critical Phenomena*, vol. 171, Oxford University Press, 2021.
- [33] M.K. Titsias, Variational learning of inducing variables in sparse Gaussian processes, in: Artificial Intelligence and Statistics (AISTATS), vol. 5, 2009, pp. 567–574.
- [34] I. Kobyzev, S.J. Prince, M.A. Brubaker, Normalizing flows: an introduction and review of current methods, *IEEE Trans. Pattern Anal. Mach. Intell.* 43 (11) (2020) 3964–3979.
- [35] J. Hensman, A.G. Matthews, M. Filippone, Z. Ghahramani, Mcmc for variationally sparse Gaussian processes, *Adv. Neural Inf. Process. Syst.* 28 (2015).
- [36] S. Mohamed, M. Rosca, M. Figurnov, A. Mnih, Monte Carlo gradient estimation in machine learning, *J. Mach. Learn. Res.* 21 (1) (2020) 5183–5244.
- [37] J. Hensman, N. Durrande, A. Solin, et al., Variational Fourier features for Gaussian processes, *J. Mach. Learn. Res.* 18 (1) (2017) 5537–5588.
- [38] L. Deng, The MNIST database of handwritten digit images for machine learning research, *IEEE Signal Process. Mag.* 29 (6) (2012) 141–142.
- [39] D. Sato, S. Hanaoka, Y. Nomura, T. Takenaga, S. Miki, T. Yoshikawa, N. Hayashi, O. Abe, A primitive study on unsupervised anomaly detection with an autoencoder in emergency head CT volumes, in: Medical Imaging 2018: Computer-Aided Diagnosis, 2018, p. 60.
- [40] K. Saab, J. Dunnmon, R. Goldman, A. Ratner, H. Sagreya, C. Ré, D. Rubin, Doubly weak supervision of deep learning models for head CT, in: Medical Image Computing and Computer Assisted Intervention, vol. 11766, 2019, pp. 811–819.
- [41] K. Jnawali, M.R. Arbabshirani, N. Rao, A.A.P. M.d, Deep 3d Convolution Neural Network for CT Brain Hemorrhage Classification, *Medical Imaging 2018: Computer-Aided Diagnosis*, vol. 10575, International Society for Optics and Photonics, 2018, p. 105751C.
- [42] J.J. Titano, M. Badgeley, J. Schefflein, M. Pain, A. Su, M. Cai, N. Swinburne, J. Zech, J. Kim, J. Bederson, J. Mocco, B. Drayer, J. Lehar, S. Cho, A. Costa, E.K. Oermann, Automated deep-neural-network surveillance of cranial images for acute neurologic events, *Nat. Med.* 24 (9) (2018) 1337–1341.
- [43] M.R. Arbabshirani, B.K. Fornwalt, G.J. Mongelluzzo, J.D. Suever, B.D. Geise, A.A. Patel, G.J. Moore, Advanced machine learning in action: identification of intracranial hemorrhage on computed tomography scans of the head with clinical workflow integration, *npj Digit. Med.* 1 (1) (2018) 9.
- [44] M. Monteiro, V.F.J. Newcombe, F. Mathieu, K. Adatia, K. Kamnitsas, E. Ferrante, T. Das, D. Whitehouse, D. Rueckert, D.K. Menon, B. Glocker, Multiclass semantic segmentation and quantification of traumatic brain injury lesions on head CT using deep learning: an algorithm development and multicentre validation study, *Lancet Digit. Health* 2 (6) (2020) e314–e322.
- [45] S. Chilamkurthy, R. Ghosh, S. Tanamala, M. Biviji, N.G. Campeau, V.K. Venugopal, V. Mahajan, P. Rao, P. Warier, Development and validation of deep learning algorithms for detection of critical findings in head CT scans, *Lancet* (2018) 2388–2396.
- [46] N.T. Nguyen, D.Q. Tran, N.T. Nguyen, H.Q. Nguyen, A CNN-LSTM architecture for detection of intracranial hemorrhage on CT scans, in: Medical Imaging with Deep Learning, 2020.