Computer Methods and Programs in Biomedicine 219 (2022) 106783

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

Deep Gaussian processes for multiple instance learning: Application to CT intracranial hemorrhage detection☆

Miguel López-Pérez[a,*], Arne Schmidt[a], Yunan Wu[b], Rafael Molina[a], Aggelos K. Katsaggelos[b]

[a] Department of Computer Science and Artificial Intelligence, University of Granada, Granada 18010, Spain
[b] Department of Electrical Computer Engineering, Northwestern University, Evanston, IL, 60208 USA

ARTICLE INFO

Article history:
Received 29 October 2021
Revised 11 March 2022
Accepted 28 March 2022

Keywords:
Multiple instance learning
Deep Gaussian processes
Intracranial hemorrhage detection
Weakly supervised learning

ABSTRACT

Background and objective: Intracranial hemorrhage (ICH) is a life-threatening emergency that can lead to brain damage or death, with high rates of mortality and morbidity. The fast and accurate detection of ICH is important for the patient to get an early and efficient treatment. To improve this diagnostic process, the application of Deep Learning (DL) models on head CT scans is an active area of research. Although promising results have been obtained, many of the proposed models require slice-level annotations by radiologists, which are costly and time-consuming. *Methods:* We formulate the ICH detection as a problem of Multiple Instance Learning (MIL) that allows training with only scan-level annotations. We develop a new probabilistic method based on Deep Gaussian Processes (DGP) that is able to train with this MIL setting and accurately predict ICH at both slice- and scan-level. The proposed DGPMIL model is able to capture complex feature relations by using multiple Gaussian Process (GP) layers, as we show experimentally. *Results:* To highlight the advantages of DGPMIL in a general MIL setting, we first conduct several controlled experiments on the MNIST dataset. We show that multiple GP layers outperform one-layer GP models, especially for complex feature distributions. For ICH detection experiments, we use two public brain CT datasets (RSNA and CQ500). We first train a Convolutional Neural Network (CNN) with an attention mechanism to extract the image features, which are fed into our DGPMIL model to perform the final predictions. The results show that DGPMIL model outperforms VGPMIL as well as the attention-based CNN for MIL and other state-of-the-art methods for this problem. The best performing DGPMIL model reaches an AUC-ROC of 0.957 (resp. 0.909) and an AUC-PR of 0.961 (resp. 0.889) on the RSNA (resp. CQ500) dataset. *Conclusion:* The competitive performance at slice- and scan-level shows that DGPMIL model provides an accurate diagnosis on slices without the need for slice-level annotations by radiologists during training. As MIL is a common problem setting, our model can be applied to a broader range of other tasks, especially in medical image classification, where it can help the diagnostic process.

1. Introduction

Intracranial hemorrhage is a severe life-threatening emergency with high rates of mortality and permanent disability. It is initially caused by blood leaking inside the cranium, where the rapidly increasing blood pressure of the brain leads to severe brain damage or death [1]. It is reported that around 40,000 to 67,000 subjects suffer from ICH per year in the United States [2] and 30% of them eventually die [3]. To avoid death or remaining damages, early treatment is crucial. The study shows that, without timely brain surgery, nearly half of the deaths occur in the first 24 h and only 20% of the surviving patients have the chance to completely recover at the end [2], indicating the important role of a fast and accurate ICH diagnosis in improving the survival rates and chances of recovery. Computed Tomography (CT) is a widely used non-invasive imaging technique for the ICH diagnosis, that is accessible and cheap for patients and at the same time, convenient

☆ Project P20_00286 funded by FEDER/Junta de Andalucía-Consejería de Transformación Económica, Industria, Conocimiento y Universidades and the European Union's Horizon 2020 research and innovation programme under the Marie Skodowska Curie grant agreement No 860627 (CLARIFY Project). Funding for open access charge: Universidad de Granada / CBUA.

* Corresponding author.
E-mail addresses: mlopez@decsai.ugr.es (M. López-Pérez), arne@decsai.ugr.es (A. Schmidt), yunanwu2020@u.northwestern.edu (Y. Wu), rms@decsai.ugr.es (R. Molina), a-katsaggelos@northwestern.edu (A.K. Katsaggelos).

https://doi.org/10.1016/j.cmpb.2022.106783

and fast for radiologists. However, studies show that radiologists may misdiagnose after long hours of CT scans readings [4,5]. As Computer-aided diagnosis (CAD) methods can help to reduce the workload of radiologists and provide an accurate diagnosis, they are of high clinical importance.

With the rapid development of DL, several models have been proposed to detect ICH. CNNs foster self-learning filters to focus on regions of interest without the need for manual feature extractions. The simplest way is to apply DL models on a single slice directly and predict the ICH at slice-level. For instance, Phong et al. [6] compared three types of traditional CNN models and found that models with pre-trained weights on non-medical images improved the ICH diagnosis. Cho et al. [7] developed a cascade DL model based on CNNs and dual fully convolutional networks to improve the sensitivity in identifying ICH. Although these models achieved good classification performances, it is challenging to collect a large number of slice annotations because manual labeling is time-consuming and requires expert knowledge. The ground truth at scan-level is, however, relatively easy to obtain, as it can be generated directly from the clinical radiologists' report. Therefore, an emerging approach using only scan-level labels consists of predicting ICH on full 3D scans. For instance, Titano et al. [8] utilized a 3D Resnet-50 CNN to predict ICH on brain scans and Jnawali et al. [9] ensembled three different 3D CNNs to improve the detection rate of ICH. However, one major problem of 3D CNNs lies in their highly expensive computation, leading to out-of-memory errors during the training processes. In addition, 3D models are not able to indicate the specific slice that contains the possible ICH inside a scan. This is however crucial to facilitate the ICH localization.

Another approach that uses only scan-level labels is the MIL paradigm. MIL is a weakly-supervised learning method that has been proposed to solve the problems when only bag labels are available [10]. It has been applied in many medical domains. Campanella et al. [11] trained a MIL model to diagnose cancer in histopathological images with slide labels by finding the highest probability per bag and then applying a recurrent neural network on the extracted features of each instance to predict the whole slide. Recently, attention-based methods are gaining more and more popularity in the field of medical imaging for the MIL setting. Similarly to channel attention mechanisms that are weighting each channel of a CNN layer with attention weights [12], the attention weights in the case of MIL are assigned to the instances [13]. These instance attention weights provide insight into the contribution of each instance to the bag predictions. Several approaches have extended this attention mechanism to different medical applications: Han et al. [14] proposed an attention-based deep 3D MIL to diagnose COVID-19 from chest CT, where the attention mechanism is able to find key instances to interpret the specific infection areas of COVID-19. Qi et al. [15] developed another deep representation based MIL system to classify COVID-19 from normal pneumonia, which was first pre-trained to generate each instance feature and then generated predictions using the k-nearest neighbors. Similarly, they found that the attention weights highlight infected lesions, providing strong evidence for the diagnosis. Other approaches for the MIL problem are based on Gaussian Processes (GPs), which were first proposed as Variational Gaussian Processes for MIL (VGPMIL) obtaining promising results in many different scenarios. For instance, they performed well for the classification of histological images of Barrett's cancer [16]. In our previous work [17], VGPMIL combined GPs with an attention-based CNN to address ICH diagnosis in a MIL setting. We proved that GPs outperformed the attention mechanism of CNNs for the ICH problem and set a new state-of-the-art for ICH diagnosis using only scan labels for training. To the best of our knowledge, this was the first time that GPs have been applied to the ICH diagnosis problem.

Although Gaussian Processes have not been widely used for ICH yet, they have achieved promising results on many other classification tasks [18], such as non-parametric and probabilistic models, which are capable of dealing with uncertainty in modeling and prediction [19]. Prior information can be included in the kernel function acting as a regularizer. Thus, they are not prone to overfitting. The flexibility, expressiveness, and robustness to overfitting of GPs make them suitable for a wide range of problems, especially, when only limited data is available. For this reason, they are promising for medical applications. In spite of all the benefits previously mentioned, GPs suffer from an important drawback. Commonly, they are used with stationary kernels. These kernels work well in many scenarios but they are not able to capture complex patterns, e.g., a function that is flat in one region and varies rapidly in another. Moreover, high parametrized kernels, which represent richer functions using shallow GPs, are expensive to train so approximate methods may be at risk of overfitting [20]. To overcome this limitation, DGPs have been introduced [21]. They are hierarchical extensions of GPs enabling to model more complex functions while retaining all the benefits of shallow GPs. DGPs can learn a representation hierarchy non-parametrically with very few hyperparameters [20]. DGPs have been used in medical imaging problems, such as histology, with sound results [22] against GPs and DL methods. So far the existing DGP-approaches focused on fully supervised training mostly for regression [20,21], classification [20–22], or special cases like multi-view representation learning [23], a learning paradigm where multiple data sources with different data formats are taken into account. To the best of our knowledge, there is no existing DGP-model for the MIL setting with only bag labels available.

This work aims to extend our previous conference paper [17], which uses an attention-based MIL combined with GPs for ICH detection. We overcome the limitation of the originally applied shallow GP, which is only capable of modeling functions with limited complexity. Therefore, instead of using GPs, we propose a novel MIL method based on DGPs called DGPMIL. The new DGPMIL is more flexible than VGPMIL and improves the performance of the classifier. In this work, we also use the attention-based CNN proposed in [17] to extract the features, but this time, the hierarchical structure of DGPs enables us to capture richer patterns. In addition, the inducing locations of DGPMIL are optimized per layer in contrast to VGPMIL, the model used in [17], where they were fixed after a k-means estimation. The main contributions are:

- We introduce DGPMIL, a novel probabilistic model based on DGPs for MIL classification. To the best of our knowledge, DGPs have never been proposed before for MIL in any domain. We outline the detailed theoretical derivation and make the implementation of the model publicly available at https://github.com/wizmik12/DGPMIL. It is based on GPytorch, a framework for GPs on top of Pytorch, and can leverage GPU computation for fast inference.
- We study the behavior of this new MIL approach in a controlled experiment using the MNIST database. This experiment shows how the greater expressiveness of deep GPs achieves better results than shallow GPs in MIL.
- Finally, we apply the DGPMIL model combined with an attention CNN to ICH detection with labels at the scan level. These experiments demonstrate the suitability of this method for medical imaging. We report competitive or superior results to current state-of-the-art methods. Remarkably, the precision obtained at detecting ICH is notably better than previous approaches for this problem.

The rest of the paper is organized as follows. Section 2 describes the proposed model. We explain the feature extraction process using an attention-based CNN and also describe DGPMIL.

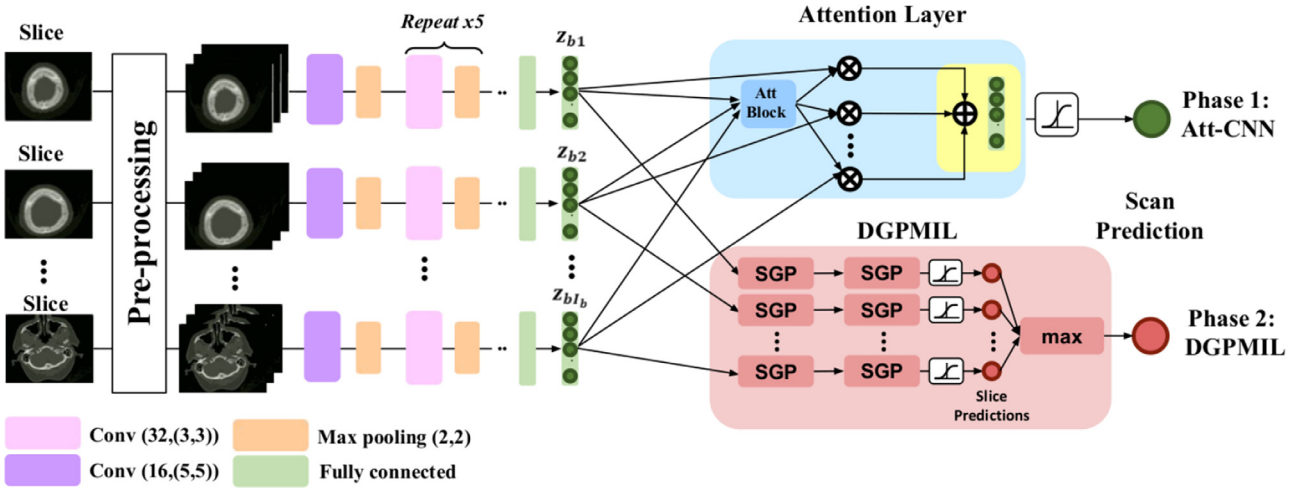


Fig. 1. The proposed architecture for the ICH detection with scan labels. In phase 1 the feature extractor is trained using an attention module for bag level predictions (Att-CNN). In phase 2 the weights of the feature extractor are frozen and DGPMIL is trained to predict slice and scan level labels. We depict only a two-layer DGPMIL here although in the experiments we use a varying amount of layers to find the optimal configuration.

Section 3 validates the method. We first create a synthetic MIL problem of digit classification to show the behavior of DGPMIL and then we perform a comprehensive validation for ICH detection on CT scans. Section 4 analyzes the main findings of the reported results and Section 5 concludes our work.

2. Methods

2.1. Problem formulation

Mathematically, we model the ICH detection as a MIL problem. We denote the set of all CT slices as $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ and the true (unobserved) slice labels as $\mathbf{Y} = \{y_1, y_2, \ldots, y_N\}$ with $y_i \in \{0, 1\}$, where the class label 1 is assigned when the slice or scan is ICH positive and otherwise 0 if no ICH is present, and N is the total number of slices in a given bag. Note that N can vary depending on the bag. In the context of MIL, these slices are called *instances* and a complete scan (consisting of multiple slices) *bags*. The bags are non-overlapping, such that each index i of an instance can be only assigned to one bag b. We denote the instances of one bag as $\mathbf{X}_b = \{x_i | i \in \text{bag } b\}$ and corresponding instance labels as $\mathbf{Y}_b = \{y_i | i \in \text{bag } b\}$. In the MIL assumption, the instance labels remain unobserved and only the bag label T_b is known. When a CT scan is diagnosed as ICH positive, at least one slice must contain the pattern of hemorrhage while a negative scan contains only negative slices, in other words,

$$T_b = \max\{y_i | i \in \text{bag } b\}. \tag{1}$$

2.2. Overview of the model

To solve the MIL problem just defined, our model is trained in two phases, described in Fig. 1. First, we train a convolutional neural network (CNN) that serves as a feature extractor in combination with an attention mechanism (Phase 1). The purpose of this phase is to build a feature extractor that is able to obtain expressive features from the slices. Although this phase 1 model (Att-CNN) is also able to predict ICH on CT scan level we disregard the attention layer after the first phase because we can experimentally prove that our DGPMIL model shows a stronger classification performance using the obtained features (see 3.4). The second phase consists of the classification using the extracted model features. In [17], the second phase was performed using VGPMIL. Notice that this shallow model could be too simple for the extracted features. In this work, we propose for the first time Deep Gaussian Processes for Multiple Instance Learning (DGPMIL). We describe the modeling and the inference with all derivations. The emphasis of this work lies on the training of the DGPMIL model (in phase 2) that provides the final slice and scan level predictions. We prove that DGPs take advantage of the complex patterns of the extracted features.

In the following subsections, we will briefly explain how the feature extraction is performed in our experiments.

2.3. Feature extraction

This subsection provides a brief introduction to the attention mechanism with CNNs to extract brain CT features at slice-level, as shown in Fig. 1. Assume a CNN model $\mathbf{F}_{cnn}$ is used to extract high dimensional features ξ_i for each instance x_i, such that $\xi_i = \mathbf{F}_{cnn}(x_i), \forall i = 1, 2, \ldots, N$. Note that the same network is applied to each instance and the weights are shared. $\mathbf{F}_{cnn}$ consists of six convolutional layers, each followed by a max pooling layer. The convolutional layers aim to extract discriminative features from each instance and the max pooling layers are used to reduce the feature dimensions. Moreover, a flatten layer and a fully connected layer are followed by to control the size of feature vectors $\xi_i \in R^{M\times 1}$ fed to the attention layer and the DGPMIL model in Phase 2.

An attention layer $\mathbf{L}_{att}$ is applied after $\mathbf{F}_{cnn}$ to estimate an attention weight α_i, corresponding to each unique feature vector ξ_i. The attention weights are used to calculate a weighted sum of feature vectors for the final, bag-level classification. Let $\Xi_b = \{\xi_i | i \in \text{bag } b\}$ be the set of all feature vectors in a bag b and $\{\alpha_i | i \in \text{bag } b\}$ be the attention weights for feature vectors Ξ_b, such that $\mathbf{L}_{att}$ is defined as

$$\mathbf{L}_{att}(\Xi_b) = \sum_{i \in b} \alpha_i \xi_i, \tag{2}$$

where

$$\alpha_i = \frac{\exp\{w^\top \tanh(V\xi_i)\}}{\sum_{j \in b} \exp\{w^\top \tanh(V\xi_j)\}}, \tag{3}$$

$w \in R^{L\times 1}$ and $V \in R^{L\times M}$ are trainable parameters that accommodate different instance numbers of a bag. The hyperparameter L is one dimension of weight matrices w and V which defines the number of trainable parameters of the attention mechanism (and is invariant to the bag size). We set $L = 50$ following the existing literature

[13]. M equals the dimension of the feature vectors, and we report the experiments for $M = 8, 32$, and 128, see Section 3.4.1. The sum of all α_i in one bag is 1. The non-linearity $\tanh(\cdot)$ aims to preserve both positive and negative values during the gradient flow.

Next, the weighted sum of the feature vectors $\mathbf{L}_{att}(\Xi_b)$ is fed to a classifier $\mathbf{F}_c$, which is made up of a fully connected layer with a sigmoid activation function, to predict the scan labels, such that

$$p(T_b|X_b) = \mathbf{F}_c(\mathbf{L}_{att}(\Xi_b)) = \mathbf{F}_c(\mathbf{L}_{att}(\mathbf{F}_{cnn}(X_b))). \tag{4}$$

The feature extractor $\mathbf{F}_{cnn}$, attention layers $\mathbf{L}_{att}$ and classifier $\mathbf{F}_c$ are trained end-to-end using the basic binary cross-entropy, CE, until it converges. The loss $\mathcal{L}$ is defined as:

$$\mathcal{L} = \sum_b CE(T_b, p(T_b|X_b)). \tag{5}$$

This whole attention CNN process is denoted as Att-CNN. For more details about the attention mechanism for MIL, we refer to [13]. Previous studies show that the labels at the instance level can be inferred from the attention weights [13,24]. The closer to 1, the more important role that specific instance contributes to the bag prediction. Therefore, in terms of this study, if a scan is predicted as normal, all slices will be considered normal. If a scan is predicted as the ICH, the slices with min-max normalized attention weights above 0.5 will be predicted as the ICH. By doing this, we are able to have weakly predicted labels at slice-level to facilitate radiologists with their diagnosis and localization. In the next section, we describe the DGPMIL model for the given problem. In what follows, to be consistent with the GP literature, we replace Ξ_b and ξ_i by $\mathbf{X}_b$ and $\mathbf{x}_i$ as the extracted feature vectors serve as an input for the final DGP classification.

2.4. Deep gaussian processes for multiple instance learning (DGPMIL)

Here, we introduce the novel DGP model to solve the MIL problem for binary classification. We outline the basic theory of GPs and DGPS in the Appendix A and refer the reader to [19,21,25] for further theoretical background. Note that in contrast to previous DGP-based methods, our proposed model trains with only the bag labels T_b while the instance labels y_b are unknown, as described in Section 2.1. For the observation model, we follow the approach used for *Variational Gaussian Process Multiple Instance Learning* [16]. There, the authors parametrize the bag label likelihood using

$$\mathrm{p}(T_b|\mathbf{Y}_b) = \frac{H^{G_b}}{H+1}, \tag{6}$$

where $G_b := T_b \max(\mathbf{y}_b) + (1 - T_b)(1 - \max(\mathbf{y}_b))$. In this equation, H is a positive constant. Notice that this likelihood is a noisy version of the MIL assumption presented in Section 2.1 and it becomes exact when H approaches infinity. The constant H controls the probability of the bag being positive considering that there is at least one positive instance. Assuming independence across bags produces the factorization

$$\mathrm{p}(\mathbf{T}|\mathbf{Y}) = \prod_{b=1}^{B} \frac{H^{G_b}}{H+1}, \tag{7}$$

where $\mathbf{T}$ refers to the variable which groups together all the bag labels.

We predict the instance label $\mathbf{y}$ by modeling a latent function $\mathbf{F}^L$ using a DGP with L layers.

Combining the Deep Gaussian Process model and the bag observation model we obtain the full probabilistic model

$$\mathrm{p}(\mathbf{Y}, \mathbf{T}, \{\mathbf{F}^l, \mathbf{U}^l\}_{l=1}^L) = \mathrm{p}(\mathbf{T}|\mathbf{y}) \cdot \mathrm{p}(\mathbf{y}|F^L) \prod_{l=1}^{L} \mathrm{p}(\mathbf{F}^l|\mathbf{U}^l; \mathbf{F}^{l-1}, \mathbf{Z}^{l-1}) \mathrm{p}(\mathbf{U}^l; \mathbf{Z}^{l-1}), \tag{8}$$

where the dependency on the observed features $\mathbf{X}$ and the hyperparameters Θ have been omitted for simplicity.

2.5. DGPMIL inference

In this subsection, we describe the inference for our DGPMIL model. Additional details are provided in Appendix B. Our goal is to approximate the intractable posterior distribution $\mathrm{p}(\mathbf{Y}, \{\mathbf{F}^l, \mathbf{U}^l\}_{l=1}^L|\mathbf{T}, \Theta)$ with an approximate distribution $\mathrm{q}(\mathbf{Y}, \{\mathbf{F}^l, \mathbf{U}^l\}_{l=1}^L)$. Specifically, we perform doubly stochastic inference for DGPs [20]. We convert the inference problem into an optimization one by maximizing the Evidence Lower Bound (ELBO), defined by

$$\mathrm{ELBO}(\mathrm{q}) = \int \mathrm{q}(\mathbf{y}, \{\mathbf{F}^l, \mathbf{U}^l\}_{l=1}^L) \times \log \frac{\mathrm{p}(\mathbf{y}, \{\mathbf{F}^l, \mathbf{U}^l\}_{l=1}^L|\mathbf{T}, \Theta)}{\mathrm{q}(\mathbf{y}, \{\mathbf{F}^l, \mathbf{U}^l\}_{l=1}^L)} \quad \mathrm{d}\mathbf{y}\mathrm{d}\{\mathbf{F}^l, \mathbf{U}\}_{l=1}^L. \tag{9}$$

.

In this work, we use the mean-field approximation, i.e., q factorizes across as follows:

$$\mathrm{q}(\mathbf{Z}, \{\mathbf{F}^l, \mathbf{U}^l\}_{l=1}^L) = \mathrm{q}(\mathbf{Y}) \times \mathrm{q}(\{\mathbf{F}^l\}_{l=1}^L|\{\mathbf{U}^l\}_{l=1}^L, \Theta) \times \mathrm{q}(\{\mathbf{U}^l\}_{l=1}^L), \tag{10}$$

with the following parametric form for each factor:

$$\mathrm{q}(\mathbf{Y}) = \prod_{n=1}^{N} \mathrm{q}(y_n) = \prod_{n=1}^{N} q_n^{y_n}(1 - q_n)^{1-y_n}, \tag{11}$$

$$\mathrm{q}(\{\mathbf{F}^l\}_{l=1}^L|\{\mathbf{U}^l\}_{l=1}^L, \Theta) = \mathrm{p}(\{\mathbf{F}^l\}_{l=1}^L|\{\mathbf{U}^l\}_{l=1}^L, \Theta), \tag{12}$$

$$\mathrm{q}(\{\mathbf{U}^l\}_{l=1}^L) = \prod_{l=1}^{L} \mathrm{q}(\mathbf{U}^l) = \prod_{l=1}^{L} \mathcal{N}(\mathbf{U}^l|\mathbf{m}^l, \mathbf{S}^l). \tag{13}$$

The proposed posterior on the instance labels $\mathbf{Y}$ factorizes across the instances and we denote by q_n the probability of the nth instance to belong to the positive class and by $\mathbf{q}_{b-n}$ all other instance probabilities in the same bag. The prior conditional $\mathbf{F}|\mathbf{U}$ does not introduce any new variational parameter. The proposed posterior distribution on $\mathbf{U}^l$ factorizes across the layers and is given by a Gaussian distribution. In summary, the variational parameters $\mathbf{V}$ to be estimated are $\{q_n\}_{n=1}^N$ and $\{\mathbf{m}^l, \mathbf{S}^l\}_{l=1}^L$.

Finally, we obtain $\mathbf{V}$, Θ, and $\{\mathbf{Z}^l\}_{l=1}^L$ by maximizing the ELBO. The ELBO can be written explicitly as

$$\begin{aligned}
&\mathrm{ELBO}(\mathbf{V}, \Theta, \{Z^{l-1}\}_{l=1}^L) = \\
&\mathbb{E}_{\mathrm{q}(\mathbf{Y})\mathrm{p}(\{\mathbf{f}^l\}_{l=1}^L|\{\mathbf{U}^l\}_{l=1}^L)\mathrm{q}(\{\mathbf{U}\}_{l=1}^L)} \left[\log \frac{\mathrm{p}(\{\mathbf{U}\}_{l=1}^L)\mathrm{p}(\{\mathbf{F}^l\}_{l=1}^L|\{\mathbf{U}^l\}_{l=1}^L)\mathrm{p}(\mathbf{Y}|\mathbf{f}^L)\mathrm{p}(\mathbf{T}|\mathbf{Y})}{\mathrm{q}(\{\mathbf{U}\}_{l=1}^L)\mathrm{p}(\{\mathbf{F}^l\}_{l=1}^L|\{\mathbf{U}^l\}_{l=1}^L)\mathrm{q}(\mathbf{Y})}\right] \\
&= \mathbb{E}_{\mathrm{q}(\mathbf{Y})\mathrm{p}(\mathbf{F}^L|\mathbf{U}^L)\mathrm{q}(\mathbf{U}^L)}\left[\log \mathrm{p}(\mathbf{Y}|\mathbf{f}^L)\right] + \mathbb{E}_{\mathrm{q}(\mathbf{Y})}[\log \mathrm{p}(\mathbf{T}|\mathbf{Y})] \\
&- \mathbb{E}_{\mathrm{q}(\mathbf{Y})}[\log \mathrm{q}(\mathbf{Y})] \\
&+ \mathbb{E}_{\mathrm{q}(\{\mathbf{U}^l\}_{l=1}^L)}\left[\log \frac{\mathrm{p}(\{\mathbf{U}^l\}_{l=1}^L)}{\mathrm{q}(\{\mathbf{U}^l\}_{l=1}^L)}\right].
\end{aligned} \tag{14}$$

Notice that the term $\mathbb{E}_{\mathrm{q}(\mathbf{Y})}[\log \mathrm{p}(\mathbf{T}|\mathbf{Y})]$ is not differentiable since it involves the max function. This fact prevents us from optimizing the ELBO using gradient descent. To overcome this limitation, we iteratively update first $\mathrm{q}(\mathbf{Y})$ and then the DGP parameters. Since we are using the mean-field approximation, following the approach of [16], we can compute the optimal distribution of $\mathrm{q}(\mathbf{Y})$

with the other distributions fixed [26]. The optimal update for $q(\mathbf{y})$ is given by (see Appendix B.1),

$$q_n \leftarrow \sigma\left(\mathbb{E}_{q(f_n^L)}\left[f_n^L\right] + \log H \cdot (2T_b + \max \mathbf{q}_{b-n} - 2T_b \max \mathbf{q}_{b-n} - 1)\right). \tag{15}$$

Using the approximation $\mathbb{E}[\max\{y_i\}] \approx \max\{\mathbb{E}[y_i]\}$ as in [16], the ELBO can be approximated by (see Appendix B.2)

$$\begin{aligned}\text{ELBO} \approx & \sum_{n=1}^{N} q_n \mathbb{E}_{q(f_n^L)}\left[\log p(y_n = 1|f_n^L)\right] \\ & + (1-q_n)\mathbb{E}_{q(f_n^L)}\left[\log p(y_n = 0|f_n^L)\right] \\ & + \log H \sum_{b=1}^{B} (2T_b \max \mathbf{q}_b - \max \mathbf{q}_b) \\ & - \sum_{n=1}^{N} q_n \log q_n + (1-q_n)\log(1-q_n) \\ & - \sum_{l=1}^{L} \mathrm{KL}\left(q(\mathbf{U}^l)||p(\mathbf{U}^l)\right) + \text{const.}\end{aligned} \tag{16}$$

Now, with q_n fixed, we can optimize the ELBO in Eq. (16) to obtain the optimal distribution for $q(\{\mathbf{U}^l\}_{l=1}^L)$, the kernel hyperparameters Θ and the inducing locations $\{\mathbf{Z}^l\}_{l=1}^L$ by using gradient descent (see Appendix B.3). Then, we can compute the variational parameters q_n with the update in Eq. (15) where the other parameters are fixed. As we commented before, this optimization is performed iteratively.

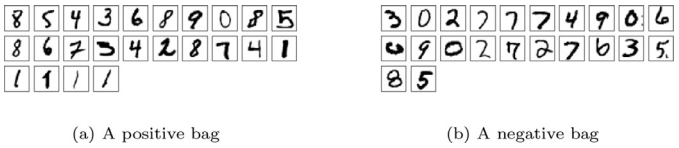
3. Experiments

This section provides an empirical validation of the proposed DGPMIL model. We carry out two different experiments. First, we create a synthetic toy example based on the popular MNIST dataset to show the behavior of DGPMIL against VGPMIL [16] in a controlled environment. Then, we use the features extracted by the attention-based CNN presented in Section 2.3 with both VGPMIL and DGPMIL for clinical ICH detection. We show the capacity of DGPMIL against the previous VGPMIL [17] and other state-of-the art methods in this problem.

3.1. Toy example: MNIST

To see the behavior of the novel DGPMIL, we analyze a synthetic MIL problem using the MNIST dataset. MNIST has 60,000 training samples and 10,000 test samples and each instance is composed of a 784-dimensional feature vector. We want to compare a shallow GP model with deep GP models to evaluate their capacity to handle high-dimensional, complex feature distributions. Since it is a controlled experiment, we carry out a comprehensive analysis to highlight its main properties. The availability of instance labels allows us to assess the model at both instance and bag levels.

In our MNIST synthetic problem, bags contain images of numbers between 0 and 9. The goal is to decide whether the bag contains at least one image of a one and, if possible, to localize it (them) in the bag. Each positive bag contains 1 to 10 positive (images of ones), and 10 to 30 negative (other numbers) instances. Negative bags contain only negative, specifically 10 to 30 negative instances. In total, we obtain 1416 negative and 1333 positive bags for training. Figure 2 shows two examples of bags in the training set. The 10,000 samples of the test set are distributed in 229 negative and 231 positive bags. We compare DGPMIL and VGPMIL models in our experiment. For the Deep Gaussian Process model, we compare the performance with 2, 3, and 4 GP layers. The dimension of the latent space of the hidden layers is set to 7 for every layer, 200 inducing points are used for each model per layer. We compute the accuracy in the test set at both instance and bag level. To assess the confidence of the methods, we also compute the log loss over the test set.



(a) A positive bag (b) A negative bag

Fig. 2. Examples of bags in the training set for the MNIST experiment.

Table 1
Results in the MNIST of Multiple Instance Methods based on Gaussian Processes using the first 30 principal components after using PCA. VGPMIL is the shallow Variational GP while DGPMIL is the deep version with 2, 3, and 4 GP layers. We assess the classification performance both at the instance and bag levels.

	Instance level		Bag level	
	Accuracy	Log Loss	Accuracy	Log Loss
VGPMIL	0.9767	0.6006	0.8496	0.4016
DGPMIL2	0.9896	0.2672	0.9586	0.2859
DGPMIL3	**0.9913**	0.2638	**0.9760**	0.2531
DGPMIL4	0.9909	**0.2602**	**0.9760**	**0.2517**

Table 2
Results in the MNIST of Multiple Instance Methods based on Gaussian Processes using the 784-dimensional feature vector. VGPMIL is the shallow Variational GP while DGPMIL is the deep version with 2, 3, and 4 GP layers. We assess the classification performance both at the instance and bag level.

	Instance level		Bag level	
	Accuracy	Log Loss	Accuracy	Log Loss
VGPMIL	0.1135	0.6931	0.4978	0.6931
DGPMIL2	0.9857	0.2729	0.9304	0.3113
DGPMIL3	0.9930	0.2655	**0.9717**	**0.2519**
DGPMIL4	**0.9932**	**0.2533**	0.9652	0.2598

3.1.1. Dimensionality reduction with PCA

Shallow methods are not good at dealing with high-dimensional complex data. This is one of the main reasons for the advent of hierarchical methods. For a fair comparison, we first reduce the dimensionality of data with Principal Component Analysis (PCA) and keep the first 30 principal components for each digit image. In the next experiment, we apply VGPMIL and DGPMIL to the raw MNIST. By doing this, we can analyze and discern the relevance of deep methods in both low and high-dimensional contexts.

Table 1 shows the comparison between VGPMIL and DGPMIL for this experiment. VGPMIL achieves a good instance classification with a value of 0.9767 in accuracy but lower for bag classification with 0.8496. In contrast, DGPMIL shows a good performance for both, instance and bag classification. For example, DGPMIL3 obtains 0.9913 at the instance level and 0.9760 at the bag level. In general, DGPMIL outperforms VGPMIL at the bag level. Regarding the log loss, VGPMIL performs poorly at the instance level which indicates that the high uncertainty lowers the overall bag classification. Although we reduced the complexity of this problem by the PCA preprocessing, we observe that the deeper GP models achieve significantly better performance on the bag level.

3.1.2. Raw MNIST data

Table 2 shows the comparison between VGPMIL and DGPMIL on the raw MNIST data. Due to the high-dimensionality of this dataset and the simplicity of the classifier, VGPMIL performs poorly. This

table shows that it predicts always the positive class at the instance and bag level. That is the reason why it reaches a value of 0.11 in accuracy for instance evaluation, while reaches a value of 0.49 in accuracy for bag evaluation. In contrast, deep models are able to process this high-dimensional data and provide accurate predictions. We can see that the best instance classifier is the deepest model DGPMIL4 with an accuracy of 0.9932 and log loss of 0.2533, followed by DGPMIL3, which achieves the best result at bag level with an 0.9717 accuracy and of 0.2519 log loss.

3.2. CT scan

So far, we have seen the behavior of DGPMIL in a controlled experiment. It shows a satisfying performance against its shallow version, i.e., VGPMIL. Now, we study the performance of an attention-based CNN combined with GP-based methods in a real-world problem. We tackle the problem of detecting ICH on brain CTs in a MIL setting. We analyze the advantages produced by using a DGP classifier on the top of the CNN instead of a shallow GP, which was presented in [17]. We consider a full scan as a bag and each slice in a scan as an instance. Generally, different scans contain a different number of slices. So in this case, the number of instances in bags varies.

3.2.1. Data preprocessing

The used dataset was published by the Radiological Society of North America (RSNA)[1] in 2019. This study includes a total of 39750 slices acquired from 1150 patients, which are further split into 1000 subjects for training and validation, and the rest 150 subjects for testing. Specifically, the training dataset includes 589 normal scans (i.e., negative cases) and 411 scans with ICH (i.e., positive cases) and the testing dataset includes 78 normal scans and 72 ICH scans. The number of slices in each scan ranges from 24 to 57 in size of 512 $\times$ 512. At slice-level, the training dataset includes 29,520 negative slices and 4976 positive slices and the testing dataset includes 806 positive slices and 4448 negative slices.

The CQ500 dataset provided by various centers in New Delhi, India [27] is used as an external test set in this study to show the generalization of our proposed model trained on RSNA. It includes the ground truth only at scan-level, including 285 normal scans and 205 ICH scans. The number of slices in each scan varies from 16 to 128.

In both datasets, in order to mimic the way radiologists often adjust different window centers (C) and widths (W) when diagnosing a brain scan, each slice is passed through three window settings to enhance the different display of the brain [W:80, C:40], blood [W:200, C:80] and soft tissue [W:380, C:40]. The windowing images from each slice are stacked together as three image channels and the intensities are normalized to [0,1] before being fed into the CNNs.

3.3. Implementation details

The model is first trained with an attention CNN with the ground truth at scan-level where the estimated attention weights will indicate the probability of that slice being positive. Then, the features at slice level can be extracted from the fully connected layers. Finally, these extracted features are fed into VGPMIL and DGPMIL.

The attention CNN is trained from scratch (without pre-trained weights) and the whole training procedure costs an average of 4.5 h. The number of training epochs is 100 and the batch size is 16 per step. The Adam optimizer [28] is used with an initial learning rate of 5×10^{-4}. The experiment is run five times independently and both the training and testing processes are performed on a single GPU (Nvidia GeForce RTX 2070 Super) using Tensorflow 2.0 and Python 3.7.

In this experiment, we compare the performance of GP-based methods and deep neural networks. We use the shallow VGPMIL and three different values of depth for DGPs: 2-layer (DGPMIL2), 3-layer (DGPMIL3) and 4-layer (DGPMIL4) models. The training of DGP models is performed with Adam optimizer, 512 mini-batch size and 30 epochs. Furthermore, the dimension of the latent space has been set to 3 for hidden layers. After several tries, we see empirically that a small latent space benefit and accelerate convergence. The learning rate is set to 0.001. While for VGPMIL we use the published implementation in NumPy of [16], DGPMIL is implemented using GPyTorch 1.3.1, which is a software for GPs based on PyTorch. The used version of PyTorch is 1.7.1.

3.4. Results

In this section, we report the results for ICH detection. First, we study the impact of the hyperparameters to the model's performance. Then, we test the model in the RSNA and the external CQ500 databases. Finally, we compare the performance of the DGPMIL to other state-of-the-art classifiers in ICH.

To measure the performance of the different variants of DGPMIL and compare to other state-of-the-art methods, we mainly use three important metrics: F1 score, Area Under the Curve of the receiver operating characteristic (ROC-AUC) and the precision-recall (PR-AUC) curve. The F1 score measures the performance based on precision and recall and is a common machine learning metric that is also suitable for class-imbalanced scenarios. The ROC plots the true positive rate against the false positive rate for different confidence thresholds of the model. Here, a good model can obtain a high true positive rate while maintaining a low false positive rate. The precision-recall curve plots precision against recall for different confidence thresholds. All three metrics have a range between 0 and 1 and the higher the value, the better.

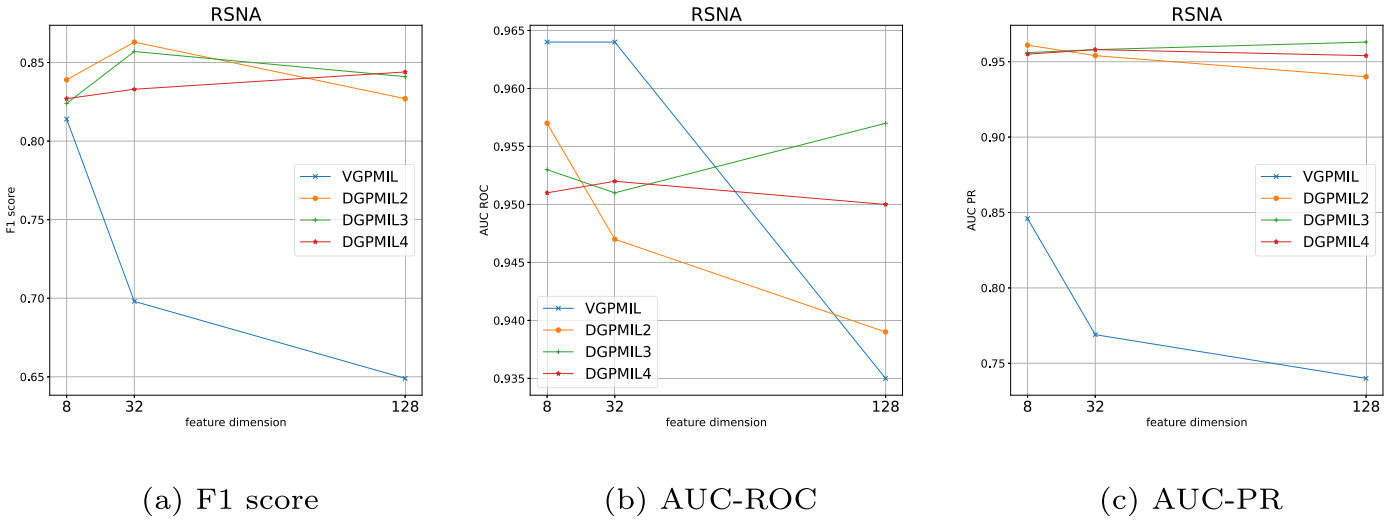
3.4.1. Ablation studies

This subsection studies the characteristics of the DGPMIL model and its hyperparameters. We conducted an ablation study. Specifically, we report the impact of the number of feature dimensions, the number of DGPMIL layers, the number of inducing points, and the dimensionality of the latent space on GPs' performance.

We start by analyzing the effect of different feature space dimensions M of the vectors ξ_i that enter the DGPMIL model and the number of GP layers, i.e., the depth of the proposed model. We compare the shallow VGPMIL to the DGPMIL models with 2, 3, and 4 layers for 8, 32, and 128-dimensional input features. We measured the performance at the scan (bag) level. During these experiments, we fixed the number of inducing points to 200 and the latent space dimensions to 3. See below for an analysis of these hyperparameters.

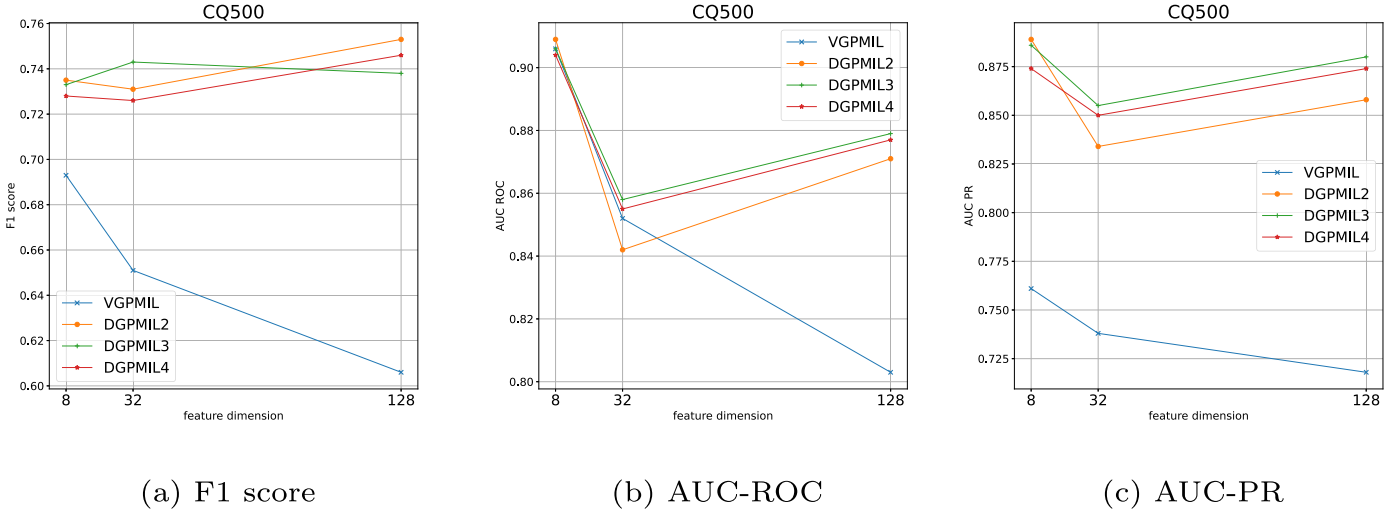
Figure 3 shows the results for the RSNA dataset, while Fig. 4 shows the results for the CQ500 dataset. Both figures report F1 score, AUC-ROC, and AUC-PR metrics. As we can observe in all figures, the shallow VGPMIL model could not achieve satisfying results for higher feature dimensions. We measured some significant performance drops, e.g., the AUC-ROC for the CQ500 dataset (Fig. 4b) drops by 5% for 32 feature dimensions and 10% for 128 feature dimensions. The DGPMIL models show more robust performance in all three metrics, and even with 128-dimensional feature vectors, they achieve satisfying results. Within the different DGPMIL models, higher feature dimensions seem to harm the DGPMIL2 model the most, as the performance drops are larger than for

[1] https://www.kaggle.com/c/rsna-intracranial-hemorrhage-detection/



(a) F1 score (b) AUC-ROC (c) AUC-PR

Fig. 3. RSNA dataset: F1 score, AUC-ROC and AUC-PR for GP and DGP models using different input feature dimensions.



(a) F1 score (b) AUC-ROC (c) AUC-PR

Fig. 4. CQ500 dataset: F1 score, AUC-ROC and AUC-PR for GP and DGP models using different input feature dimensions.

the DGPMIL3 and DGPMIL4 models for all AUC metrics (Figs. 3b,c, 4c,d). Regarding the F1 score, we can even see improved performance when using more feature dimensions. The DGPMIL3 and DGPMIL4 models both show a better F1 score when using 128 dimensions in comparison to 8 on both datasets (see Figs. 3a and 4a). Overall we observed that DPGMIL can learn useful models from feature vectors of higher dimensions while the shallow VGPMIL can not. In Section 4, we further discuss this interesting relationship between feature dimensions and GP layers.

In the final experiments, we stick to DGPMIL2 with 8 feature dimensions because this setting still achieves the best results on both datasets in terms of AUC-ROC and AUC-PR.

Next, we studied the effect of varying the number of inducing points while leaving the feature dimensions fixed at 8 and latent space dimensions at 3. As reported in Table C.1, we observed a robust performance across different numbers of inducing points. 200 inducing points show the best F1 scores for both datasets and the best AUC ROC for the RSNA dataset, we use this setting for the following experiments. Further increasing the number of inducing points did not provide any significant improvement and led to higher computational costs. Similarly, we conducted experiments to prove that the relatively small number of GP's latent space dimensions of $D = 3$ is enough. Table C.2 shows that the performance of the model with 3 and 10 latent dimensions is comparable, while 50 dimensions lead to a model that can not converge anymore.

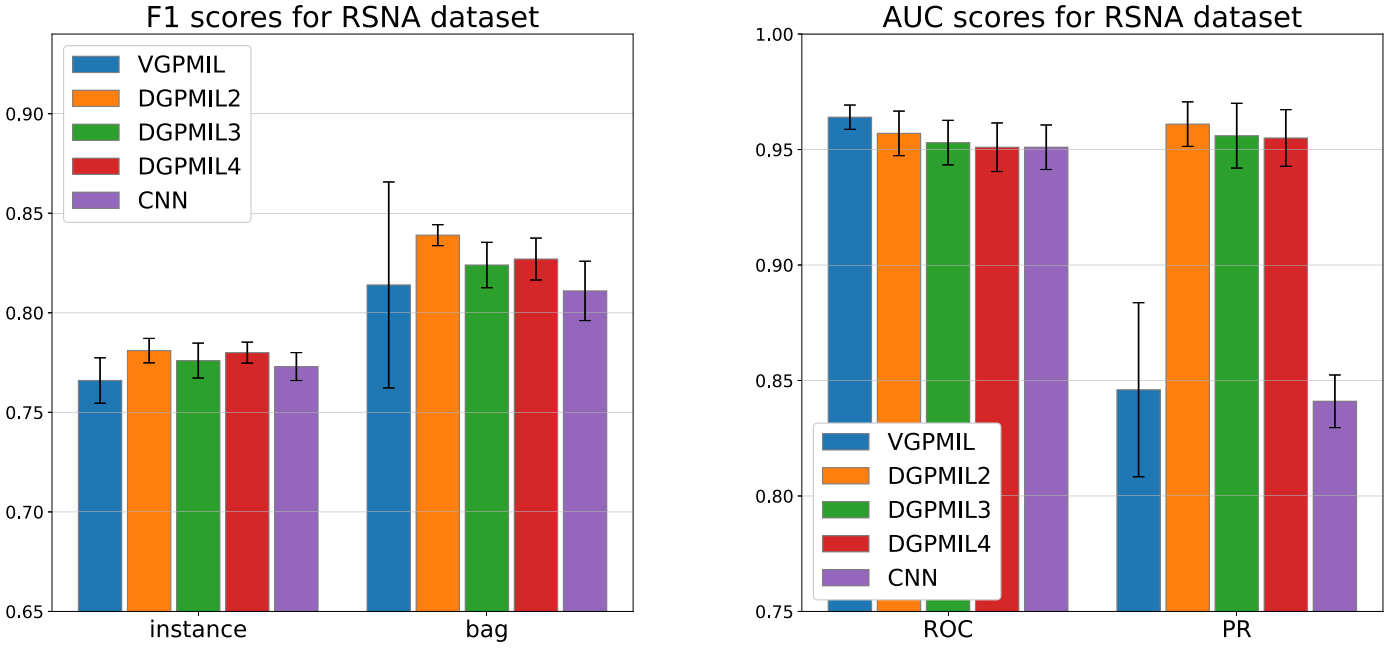
In summary, we observed that the DGPMIL model is not very sensitive to the analyzed hyperparameters. In these experiments we made an interesting observation: for higher-dimensional feature vectors, more GP layers should be used because the shallow VGPMIL model is not able to obtain good results. This finding is further discussed in Section 5. For the final results, we used 8 dimensional feature vectors, 200 inducing points, and 3 dimensions in the GP latent space. For the number of GP layers, all model variants are included in the experiments under the names DGPMIL2, DGPMIL3, and DGPMIL4.

3.4.2. Results for the RSNA dataset

Table 3 shows the results of testing with the RSNA dataset for 8-dimensional features. For this test set, although models are trained with only the scan labels, we have both slice and scan labels to evaluate the model performance. We reported the performances of the Attention-CNN model, VGPMIL, and DGPMIL with a different number of layers. Mean-aggregation of the feature vectors was previously analyzed for this problem [17] and can be considered a simple baseline with a bag-level ROC-AUC of 0.768. Regard-

Table 3
Mean results testing with the RSNA dataset for 8-dimensional features in five different runs at both slice and scan level. VGPMIL is the shallow Variational GP while DGPMIL is the deep version with 2, 3, and 4 GP layers. The CNN stands for the attention-based CNN.

Slice level metrics	VGPMIL	DGPMIL2	DGPMIL3	DGPMIL4	CNN
Accuracy	**0.938±0.003**	0.929±0.003	0.927±0.005	0.928±0.002	0.923±0.005
F1 score	0.766±0.013	**0.781±0.007**	0.776±0.01	0.780±0.006	0.773±0.008
Cohen's kappa	0.731±0.015	**0.739±0.009**	0.732±0.013	0.737±0.007	0.727±0.011
Scan level metrics	**VGPMIL**	**DGPMIL2**	**DGPMIL3**	**DGPMIL4**	**CNN**
Accuracy	0.780±0.089	**0.825±0.006**	0.805±0.014	0.809±0.018	0.781±0.023
F1 score	0.814±0.059	**0.839±0.006**	0.824±0.013	0.827±0.012	0.811±0.017
Cohen's kappa	0.567±0.172	**0.654±0.011**	0.614±0.029	0.622±0.035	0.569±0.045
AUC-ROC	**0.964±0.006**	0.957±0.011	0.9530±0.011	0.951±0.012	0.951±0.011
AUC-PR	0.846±0.043	**0.961±0.011**	0.956±0.016	0.955±0.014	0.841±0.013



(a) F1 score for RSNA dataset (b) AUC values for RSNA dataset

Fig. 5. RSNA dataset with 8-dimensional features: F1 score and AUC values with 0.95 confidence interval.

ing our analyzed models, the CNN model obtains the worst results and coupling the feature vectors to GPs improves the performance considerably. For most of the metrics at slice and scan levels, we see that DGPMIL2 shows the best performance.

Figure 5 shows F1 score and AUC values with 0.95 confidence interval. We can see that VGPMIL has a high variance for the F1 score and AUC-PR at the bag label while DGPMIL obtains good results with tight confidence intervals. This shows that DGPMIL is more robust. Furthermore, the non-overlapping intervals of DGPMIL against its competitors at the AUC-PR show visually the statistically significant improvement of DGPMIL thanks to the better precision.

Some examples of DGPMIL predictions for the RSNA dataset can be found in Fig. 6. Furthermore, we include some misclassified slices in Fig. 7. Figure 7a and b are false negatives with prediction probabilities of 0.23 and 0.16. We found that they are both the only positive slice in their own scans, so the model is more difficult to detect those small and mild types of hemorrhage. Figure 7c is a false positive slice predicted from an ICH scan with probability of 0.60. It is adjacent to a positive slice, so it might be predicted as positive because some bleeding can still be found in this slice. Figure 7d is a false positive slice predicted from a normal scan with probability of 0.59. In this case, although the probability is low and close to 0.5, a false positive slice will lead to an overall positive scan prediction. Therefore, in order to handle all these challenges, for future work we propose to not treat the instances independently but focus more on the correlations among the instances, i.e., the sequence of the slices in a scan.

3.4.3. Results for the external database CQ500

Table 4 shows the results of our trained model (on RSNA) tested with the CQ500 dataset for 8-dimensional features. For this test set, we only have scan labels. DGPMIL2 outperforms all other models in all metrics. Especially in the Cohen's Kappa value and AUC-PR we can see huge improvements in comparison to the CNN and VGPMIL model. Figure 8b shows F1 score and AUC values with 0.95 confidence interval. We can see that VGPMIL has a large variance both for the F1 score and AUC-PR metrics. Again, DGPMIL, specially DGPMIL2 and DGPMIL3, obtains a tight confidence interval even when generalizing to an external database. The non-overlapping confidence intervals show the statistical superiority of the proposed DGPMIL in AUC-PR.

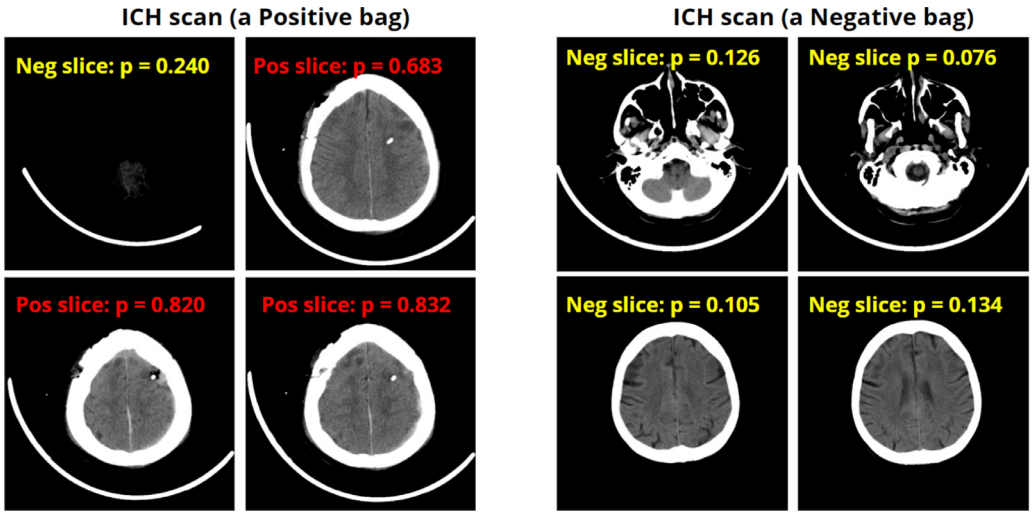


Fig. 6. Examples of two bags with DGPMIL predictions at bag-level and at instance-level. Left: an ICH scan with a bag prediction of 0.834; Right: a normal scan with a bag prediction of 0.217. Probability $p \geq 0.5$ denotes an ICH prediction is positive and $p < 0.5$ denotes a negative ICH prediction. The model is trained at bag-level but it is able to provide individual instance label correctly as the p values indicate.

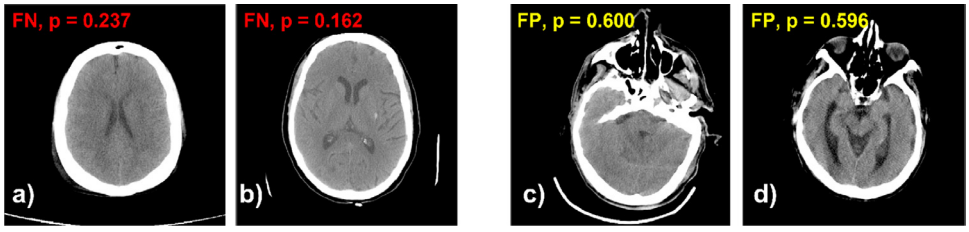
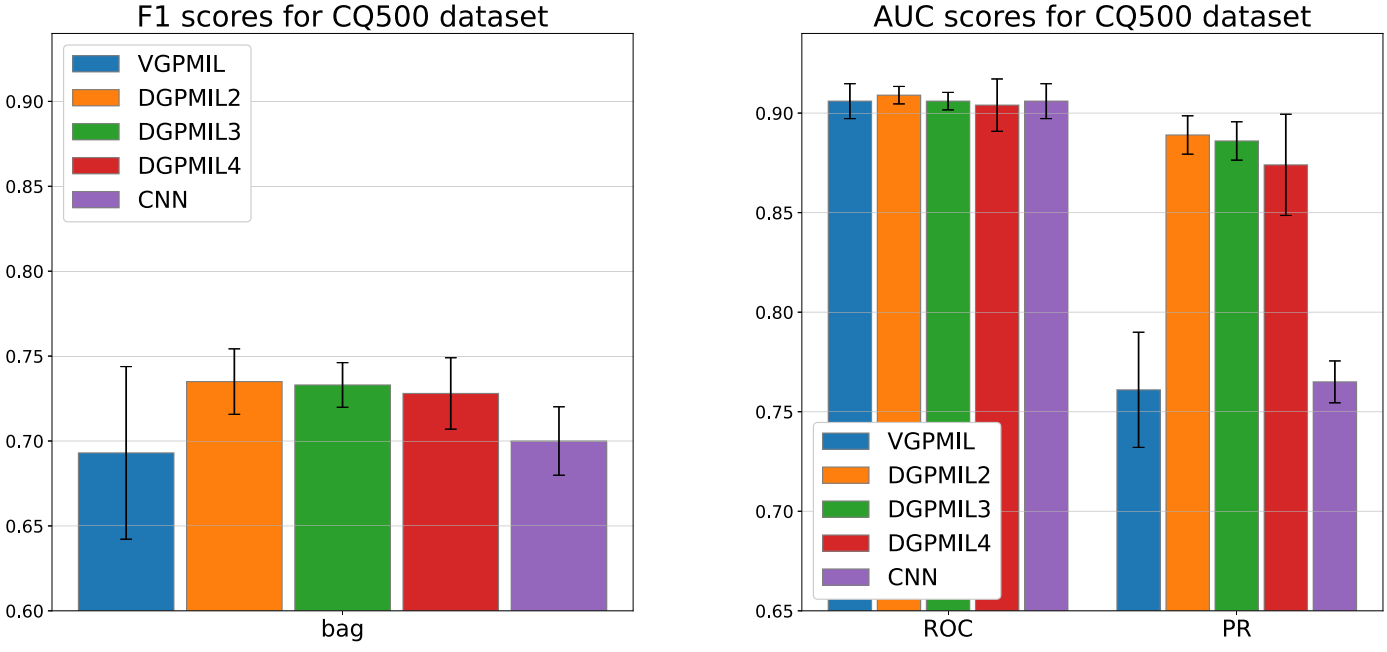


Fig. 7. Examples of False Negatives (FN) and False Positives (FP) with DGPMIL predictions at the instance level.(a,b) False Negatives; (c) a False Positive from a positive bag; (d) a False Positive from a negative bag. Probability $p \geq 0.5$ denotes an ICH prediction is positive and p <0.5 denotes a negative ICH prediction.



(a) F1 score for CQ500 dataset (b) AUC values for CQ500 dataset

Fig. 8. CQ500 dataset with 8-dimensional features: F1 score and AUC values with 0.95 confidence interval.

Table 4
Mean results testing with the CQ500 dataset for 8-dimensional features in five different runs at scan level. VGPMIL is the shallow Variational GP while DGPMIL is the deep version with 2, 3, and 4 GP layers. The CNN stands for the attention-based CNN.

Scan level metrics	VGPMIL	DGPMIL2	DGPMIL3	DGPMIL4	CNN
Accuracy	0.639±0.106	**0.717±0.035**	0.713±0.023	0.701±0.041	0.655±0.043
F1 score	0.693±0.058	**0.735±0.022**	0.733±0.015	0.728±0.024	0.700±0.023
Cohen's kappa	0.335±0.171	**0.461±0.059**	0.455±0.039	0.436±0.068	0.359±0.069
AUC-ROC	0.906±0.010	**0.909±0.005**	0.906±0.005	0.904±0.015	0.906±0.010
AUC-PR	0.761±0.033	**0.889±0.011**	0.886±0.011	0.874±0.029	0.765±0.012

Table 5
Comparison of different approaches for binary ICH detection. Our results are reported as the mean of 5 independent runs.

ICH detection at scan-level with different dataset				
Source	Dataset size	Labeling type	Method	ROC AUC
Saab et al. [24]	4340 scans	Scan	MIL	0.91
Jnawali et al. [9]	40357 scans	Scan	3D CNNs	0.87
Titano et al. [8]	37236 scans	Scan	3D CNNs	0.88
Sato et al. [30]	126 scans	Scan	3D Autoencoder	0.87
Arbabshirani et al. [29]	45583 scans	Scan	3D CNNs	0.85
VGPMIL (Wu et al. [17])	1150 scans	Scan	MIL	0.964
DGPMIL2	1150 scans	Scan	MIL	0.957
Evaluation on CQ500				
Source	Dataset size	Labeling type	Method	ROC AUC
Chilamkurthy et al. [27]		Slice	2D CNNs	0.94
Nguyen et al. [32]	490	Slice	2D CNN + LSTM	0.96
Monteiro et al. [31]	scans	Scan	voxel-based CNN	0.83
VGPMIL (Wu et al. [17])		Scan	MIL	0.906
DGPMIL2		Scan	MIL	0.909

3.4.4. State-of-the-art comparison

The performance of DGPMIL is compared with those state-of-the-art studies, as shown in Table 5. It shows that our model outperforms other models trained at scan-level with an AUC-ROC of 0.957, including basic MIL [24], 3D CNNs [8,9,29], and 3D autoencoder [30]. In addition, it is comparable to VGPMIL [17] with an AUC-ROC of 0.964. Note, that in this case, different scan-level approaches for ICH detection are compared that are using different datasets. Therefore we add a comparison of different models for the CQ500 dataset, where all models are tested on the same set. At the same time, this dataset serves as an external test set (as described above) because the model is trained on the RSNA dataset. DGPMIL achieves an AUC-ROC of 0.909, which performs better than the methods that are trained at the same scan-level with an AUC-ROC of 0.906 [17] and 0.83 [31]. Furthermore, the performance of DGPMIL is comparable to those trained at slice-level [27,32], where the AUC-ROC scores ranged from 0.94-0.96.

4. Discussion

In MIL problems, having a good instance classifier does not necessarily lead to a good bag classification. For the MIL setting, one misclassification of one instance leads to the wrong classification of a full bag. For this reason, well-calibrated models are desirable in MIL. The introduction of DGPMIL overcomes this problem and reaches much better classification performance at the bag level. Furthermore, it still retains a good instance performance, making it suitable for classifying new unseen or unlabeled instances.

DGPMIL achieves State-of-the-art results and generalizes better. Table 5 compares the ICH prediction results with other methods at scan-level. DGPMIL outperforms other methods based on AUC-ROC score except for VGPMIL [17], but DGPMIL performs significantly better than [17] in AUC-PR score and F1 score as previously discussed. Furthermore, we include an external database (CQ500) to check the generalization capability of our proposed models. In this real-world scenario, we are more interested in training a model on a dataset from a center and using it to predict correctly on the dataset from another center. The external evaluations on CQ500 dataset show that DGPMIL outperforms other models in Table 4, which proves the good generalization of our model. We further compare the performance of DGPMIL on CQ500 with those state-of-the-art studies in Table 5. It shows that DGPMIL outperforms other methods train with the same labeling type on the scan [17,31] and it is comparable to other studies that training with precise slice labels [27,32]. It is remarkable that DGPMIL2 performs well across all different feature spaces. In addition, by selecting the number of layers, we can adjust the model to extract features with different dimensions. Since DGPMIL achieves good predictions at scan level, it is the most suitable for diagnosis on unseen scans from different centers.

DGPMIL is able to achieve good results with complex high-dimensional data. We have seen in the MNIST experiment (Section 3.1) as well as in the ablation studies of the hemorrhage classification problem (Section 3.4.1) that the DGPMIL model can handle complex, high-dimensional feature distributions while the shallow VGPMIL model shows significant performance drops. This can be explained by the better ability to approximate complex functions due to multiple stacked GP layers. It enables the model to transform the feature distribution in the latent space, as depicted in the explanatory example of Fig. A.12, and leads to higher expressiveness. This property makes the DGPMIL especially interesting for other problems with a fixed, high number of feature dimensions where the DGPMIL model can be expected to outperform shallow models like VGPMIL by even a larger margin than in our final results with 8-dimensional features.

DGPMIL outperforms VGPMIL in a synthetic example. The first experiment is compared DGPMIL and VGPMIL models on a synthetic example using the MNIST dataset. Regarding the instance classification, the overall performance of DGPMIL is only slightly

better than VGPMIL when PCA is implemented. This indicates that for a problem with low-dimensional extracted features, both shallow and deep models perform well when classifying instances. However, this is not the case for bag classifications where DGPMIL outperforms VGPMIL and it corroborates the premise of a good instance classification is not enough. The proposed DGPMIL overcomes this limitation and is more suitable for MIL problems than the previous VGPMIL. As shown in Table 2, without a previous feature extraction on MNIST dataset, VGPMIL is not able to learn a good model.

Coupling an attention-based CNN with GPs produces better results. Although CNNs are widely applied in different areas of medical images, using only a standard CNN in MIL problems is not good enough because many details in bags are hidden. For the ICH detection task, we show that the CNN predictions can be substantially improved by further utilizing the extracted features with GP models (i.e., both VGPMIL and DGPMIL), leading to better instance and bag classification results. As shown in Fig. 6, with the features extracted by an attention-based CNN, DGPMIL is able to train images at scan-level and accurately predict images at slice-level. This fact encourages the use of GP models for ICH detection without radiologists' manual annotations on each slice. Since probabilistic models quantify better the uncertainty and are therefore even more adequate for this medical diagnosis scenario than deterministic model such as standard CNNs.

DGPMIL retains a good precision. The F1 score achieved by DGPMIL is better than that obtained by the CNN and VGPMIL. Considering the AUC of the ROC and PR curves, we observe that although VGPMIL and the CNN show good AUC-ROC results, their AUC-PR results are worse, meaning that the precision scores of these models are poor compared with DGPMIL. In other words, both VGPMIL and the CNN produce many false positives, which overload the doctors with a lot of false ICH detections. DGPMIL is capable of detecting suspicious cases with a high precision, as shown in Table 3, that the AUC-PR of DGPMIL2 for RSNA dataset reaches that of 0.961.

DGPMIL performs much better at the bag level. This fact has been already reflected in the synthetic example of MNIST and has been further confirmed on a real-world CT scan experiment. Although sometimes VGPMIL achieves a good classification on CT slices, DGPMIL outperforms VGPMIL at scan level. In terms of MIL problem, misclassifying only one instance in a negative bag will ruin the classification of the full bag. This is the reason why both VGPMIL and the CNN misclassify many negative bags with false positives because they cannot handle the uncertainty quantification while DGPMIL achieves the best precision and as a consequence reaches a better diagnosis at the bag level.

Advantages and drawbacks of DGPMIL: Our approach is an attractive alternative to attention CNNs for MIL that achieves good performance by integrating a probabilistic model, Gaussian Processes. In addition, compared to other weakly supervised learning methods [8,9], DGPMIL is easy to train as it does not have many hyperparameters or model parameters and can be used even with limited computing power. This work exploits its formulation to achieve a satisfying performance compared to previous methods for ICH detection, as shown in Table 5, at both scan-level and slice-level. Furthermore, the AUC-PR results are remarkable in comparison to other models in Tables 3 and 4. This metric indicates that it is not prone to have many false positives, which is important for medical applications to not distract from the really severe cases. Furthermore, it is robust to overfitting and generalizes better than other methods on external testing dataset [17,31]. However, as DGPMIL can not deal with images directly, it relies on a first step based on a CNN for feature extraction. Although this adds on extra training and parameter tuning procedures, it shows that our method can generalize well to other MIL problems [33] by just exchanging the feature extractors. Future work will focus on building an end-to-end training CNNs and GPMIL model. Another drawback of DGPMIL is that it does not take the order of the instances into account. Instances are trained independently in a bag, but the correlations existing in nearby instances may boost the performance of the model. Future work will try to implement some sequential models [32] into DGPMIL to extract the features among the order of instances.

5. Conclusions

In this work, we propose a novel model, DGPMIL, for MIL classification based on DGPs. DGPs are a hierarchical extension of the widely used GPs. Furthermore, we use DGPMIL for ICH detection on CT scans combined with the features extracted by an attention-based CNN using only scan labels. To the best of our knowledge, this is the first time DGPs have been proposed for the MIL problem and specifically for ICH detection.

The experiments show that DGPMIL can obtain good results with high-dimensional data by extracting more complex patterns in contrast to the shallow VGPMIL. For instance, DGPMIL outperforms VGPMIL in a synthetic MIL problem of classifying digits using the MNIST database. When using data with dimensionality reduction, VGPMIL performs slightly worse at the instance level compared to deep versions. However, when raw MNIST is used, VGPMIL can not learn a good model. Furthermore, DGPMIL performs notably better at the bag level, which is the final objective of the MIL problem.

We empirically validate the model in a real-world application. We detect ICH on CT scans using only scan labels. The experiment results demonstrate that combining a CNN with a GP leads to an improvement in the results. DGPMIL achieves the best performance compared to VGPMIL, the attention CNN and other state-of-the-art methods. Furthermore, it achieves a great precision value in contrast to VGPMIL and the attention CNN.

Additionally, we use a different database for assessing the generalization capability of the methods. This evaluation proves that DGPMIL generalizes better when predicting at scan level. All of these facts make DGPMIL with an attention-based CNN suitable for ICH diagnosis. Also, it can potentially be applied to many other medical-imaging problems.

Declaration of Competing Interest

Deep Gaussian Processes for Multiple Instance Learning: Application to CT Intracranial Hemorrhage Detection Miguel Lez-Pez (corresponding author), Universidad de Granada, Spain. This statement is to certify that all Authors have seen and approved the manuscript being submitted. We warrant that the article is the Authors' original work. We warrant that the article has not received prior publication and is not under consideration for publication elsewhere. On behalf of all Co-Authors, the corresponding Author shall bear full responsibility for the submission. This research has not been submitted for publication nor has it been published in whole or in part elsewhere. We attest to the fact that all Authors listed on the title page have contributed significantly to the work, have read the manuscript, attest to the validity and legitimacy of the data and its interpretation, and agree to its submission to the Journal of Computers Methods and Programs in Biomedicine

Appendix A. Revisiting Gaussian processes

This appendix provides a brief introduction to GPs for binary classification. Let us assume a dataset $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$ composed of N instances with $y_n \in \{0, 1\}$.

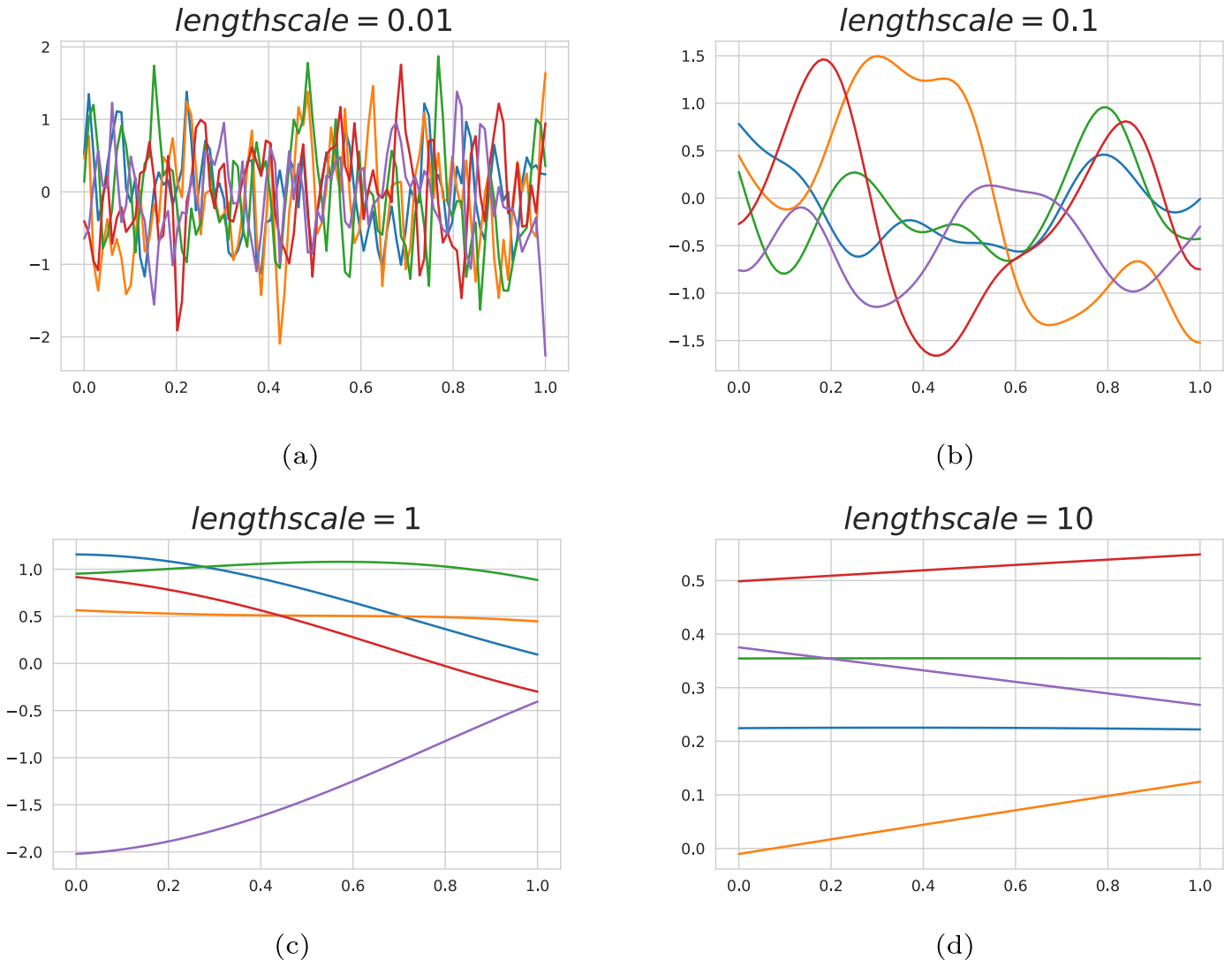


Fig. A1. Example of different sampled functions from a 1-dimensional GP with an SE kernel. Y-axis represents the value of the sampled function and X-axis the input feature of the GP. We use different values of the lengthscale hyperparameter to show how it affects the resulting functions. Shorter values of the lengthscale l produce wriggly curves while larger values produce flat functions.

A Gaussian process prior assumes a multivariate normal distribution in the latent variable $\mathbf{f} = (f_1, \ldots, f_N)^\intercal$ given $\mathbf{X}$. This prior distribution is defined by a mean function $\mu(\mathbf{x})$ and a kernel (covariance function) $k(\mathbf{x}, \mathbf{x}')$. The mean function is usually set to $\mathbf{0}$, without losing generality. The kernel encodes the prior belief about the data. In this paper we use the Squared Exponential (SE) kernel. It is a common choice in Gaussian Processes due to its flexibility and expressiveness. Also, it encodes smoothness in the latent function, which is a desirable property in many different scenarios. The SE kernel is defined as $k_{\text{SE}}(\mathbf{x}_i, \mathbf{x}_j) = C \exp\left(\frac{-||\mathbf{x}_i - \mathbf{x}_j||^2}{2l^2}\right)$, where the parameters C and l are estimated through the learning task. Figure A.9 shows samples of a GP prior with a SE kernel with different values of l. We can see that the level of smoothness relies on the value of l. Large values of l produce flat functions while small values produce less smooth functions. It is worthy noticing that these functions do not have varying levels of smoothness across the data points. This is one of the motivation to use DGPs, e.g., functions with flat areas and abrupt jumps.

Once we have modelled the latent function $\mathbf{f}$ using a GP prior, we have to define the observation model. Our likelihood for binary classification is the Bernoulli distribution, i.e., $\mathrm{p}(y_i|f_i) = \mathrm{Ber}(y_i; \sigma(f_i))$. Here, σ is the sigmoid and $f_i = f(\mathbf{x}_i)$ refers to the value of the latent variable f at the point $\mathbf{x}_i$. The joint density of $\mathbf{y}$ and $\mathbf{f}$ becomes,

$$\mathrm{p}(\mathbf{y}, \mathbf{f}) = \underbrace{\prod_{n=1}^{N} \mathrm{p}(y_n|f_n)}_{\text{likelihood}} \underbrace{\mathrm{p}(\mathbf{f})}_{\text{GP prior}}, \tag{A.1}$$

where we assume independence across the instance labels given the latent variables. The goal becomes the estimation of the model parameters, in this case C and l, and the calculation of $\mathrm{p}(\mathbf{f}|\mathbf{y})$.

One main drawback of Gaussian Processes is their scalability. They have computational cost $\mathcal{O}(N^3)$ because their use involves the inversion of an $N \times N$ matrix. To overcome this limitation, sparse GPs have been proposed [34]. The idea behind them is to define $\tilde{M} \ll N$ inducing points u_m which are GP realizations at inducing locations $\mathbf{z}_m$. We can see this as $f(\mathbf{z}) = u$. The inducing points encode the information of the observations in a few points. Their locations $\{\mathbf{z}_m\}_{m=1}^{M}$ are estimated while learning. This approach lightens the computational cost to $\mathcal{O}(n\tilde{M}^2)$. However, the posterior distribution is intractable and approximate inference must be used. The Scalable Variational Gaussian Process (SVGP) in-

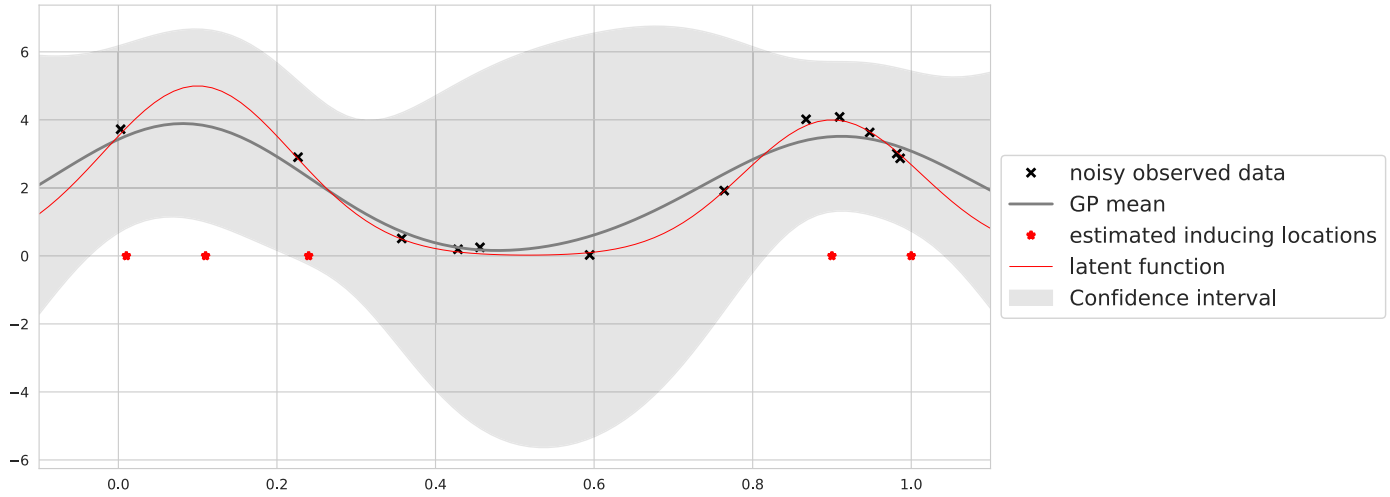


Fig. A2. Example of a Sparse Gaussian Process on a 1-dimensional regression problem. We draw the latent function that generates the noisy observed data, the mean of the estimated GP, the uncertainty and also the estimated inducing locations. The GP has more uncertainty where there are less inducing points.

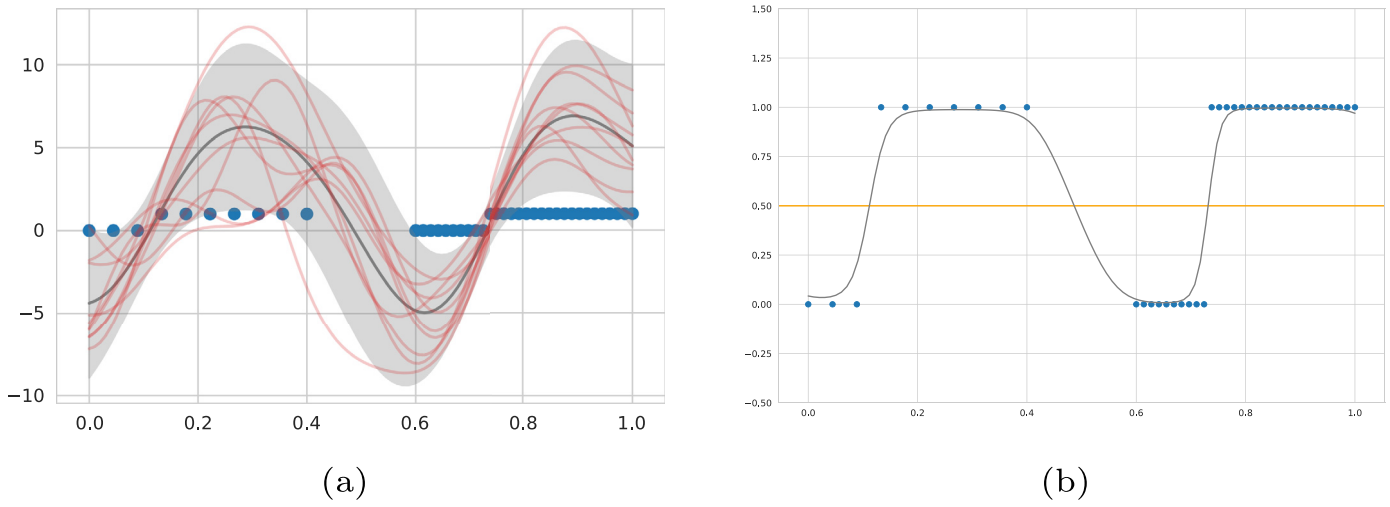


Fig. A3. 1-dimensional binary classification problem with the input dimension on the x-axis and output dimension on the y-axis. The blue points represent the noisy observed data. In (a) we draw the distribution of the latent function $p(f_*)$: the gray line is the mean and the gray shadow the 0.95 confidence interval on the predictions. The classifier has more uncertainty in the region where there are no observations. In (b) we squash the latent function to the [0,1] interval, the black line is $p(y_* = 1)$.

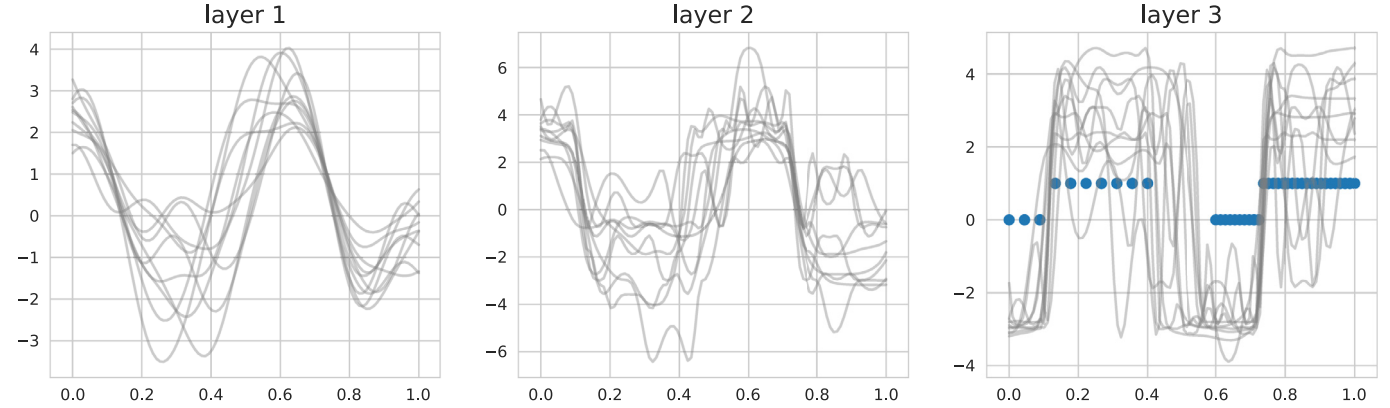


Fig. A4. Samples at every layer of a three-layer DGP trained on a binary toy example. The first two layers are latent spaces where the features are projected onto. The third layer is the output for the final classification. The y-axis represents the values of the latent function before it goes through the sigmoid function. Positive values will be classified as in the positive class and negative values as in the negative one.

ference is the state of the art for sparse GPs [18]. Furthermore, it allows to train in mini-batches. The joint density in this case is given by

$$\mathrm{p}(\mathbf{y},\mathbf{f},\mathbf{u}) = \underbrace{\prod_{n=1}^{N}\mathrm{p}(y_n|f_n)}_{\text{likelihood}}\underbrace{\mathrm{p}(\mathbf{f}|\mathbf{u};\mathbf{Z})\mathrm{p}(\mathbf{u};\mathbf{z})}_{\text{sparse GP prior}}, \tag{A.2}$$

the semicolon notation indicates which are the inputs of each function. The goal here is to calculate $\mathrm{p}(\mathbf{u},\mathbf{f}|\mathbf{y})$ and estimate the model parameters.

Figure A.10 shows a Sparse Gaussian Process for a 1-dimensional regression problem. We see that the GP mean approaches the latent function that generates the noisy observed data. The latent function is inside the confidence interval, and the uncertainty is larger in areas with less inducing points. Also notice, that the optimal location for the inducing points is where the function has more variations. Figure A.11 shows a GP for binary classification in a 1-dimensional toy problem. In (a), we draw samples for the posterior distribution of $\mathrm{p}(\mathbf{f}|\mathbf{y})$. We can notice that all the samples have the same level of smoothness. Then, in (b) we show the probabilities estimated for the positive class after the sigmoid function.

A1. Revisiting deep gaussian processes

A DGP is a hierarchical model which consists of several stacked SVGPs, i.e., the output of a SVGP is the input for the next SVGP [21]. We define $\{\mathbf{F}^l\}_{l=1}^L$ latent variables where each $\mathbf{F}^l$ follows a GP prior with input locations given by $\mathbf{F}^{l-1}$. We consider $\mathbf{F}^0 = \mathbf{X}$. We denote $f^l_{n,d}$ as the latent variable value for the nth instance in the dimension d (being $1 \le d \le D^l$) for the layer l. Notice that in this problem $D^L = 1$. The vector f^l_n contains all the dimensions for the nth instance in the lth later. The likelihood of the unobserved instance labels is defined by a Bernoulli distribution,

$$\mathrm{p}(y_n|f^L_n) = \sigma(f^L_n)^{y_n}\big(1-\sigma(f^L_n)\big)^{1-y_n}, \tag{A.3}$$

Assuming independence across the instance labels given the latent variables, we obtain,

$$\mathrm{p}(\mathbf{Y}|\mathbf{f}^L) = \prod_{n=1}^{N}\mathrm{p}(y_n|f^L_n). \tag{A.4}$$

Because of the computational cost, we have to introduce again the so called sparsity. We have M^{l-1} inducing locations $\mathbf{Z}^{l-1}$ at each layer l with inducing values $\mathbf{U}^l$ for each dimension. So we can write the joint density function,

$$\begin{aligned}\mathrm{p}(\mathbf{Y},\{\mathbf{F}^l,\mathbf{U}^l\}_{l=1}^L) = &\underbrace{\prod_{n=1}^{N}\mathrm{p}(y_n|f^L_n)}_{\text{likelihood}}\\ &\times\underbrace{\prod_{l=1}^{L}\mathrm{p}(\mathbf{F}^l|\mathbf{U}^l;\mathbf{F}^{l-1},\mathbf{Z}^{l-1})\mathrm{p}(\mathbf{U}^l;\mathbf{Z}^{l-1})}_{\text{DGP prior}}.\end{aligned} \tag{A.5}$$

The Doubly Stochastic Variational Inference is the state of the art for DGPs [20]. Furthermore, it allows to perform approximate inference and to train in mini-batches.

Figure A.12 shows samples of the DGP latent function. We show samples from the first and second layer, which are the middle latent representation features before the final classification is done. Then, the third (final) layer is the one that makes the final classification. We can see that the first layer produces smooth functions similar to the ones of the shallow GP. When we apply a GP to these features we can obtain more complex patterns as shown in the second layer. The flat regions are smooth while the jumps in the decision boundaries are abrupter. Although it is a very simple problem, we actually can see the greater expressiveness of DGPs against shallow GPs. This fact encourages their use for complex tasks, as it is in the ICH detection problem.

Appendix B. Detailed DGPMIL inference

This appendix contains all the details for inference in DGPMIL. We follow the doubly stochastic inference to estimate the variational parameters corresponding to the DGP [20]. Together with $\mathbf{Y}_b = \{y_i | i \in \text{bag } b\}$, as defined in Section 2.1, we introduce $\mathbf{Y}_{b-n} = \{y_i | y_i \in \text{bag } b \text{ and } i \neq n\}$.

B1. Update of $\mathrm{q}(\mathbf{y})$

The optimal $\mathrm{q}(y_n)$ distribution fixing the other distributions is given by

$$\begin{aligned}\log \mathrm{q}(y_n) &= \mathbb{E}_{\mathrm{q}(\mathbf{Y}_{b-n})}[\log \mathrm{p}(T_b|\mathbf{y}_b)] + \mathbb{E}_{\mathrm{q}(f^L_n)}\big[\log \mathrm{p}(y_n|f^L_n)\big] + \text{const}\\ &= \log H \cdot \mathbb{E}_{\mathrm{q}(\mathbf{Y}_{b-n})}[G_b] + \mathbb{E}_{\mathrm{q}(f^L_n)}\big[\log \mathrm{p}(y_n|f^L_n)\big] + \text{const}.\end{aligned} \tag{B.1}$$

Now we rewrite the max function as

$$\max \mathbf{Y}_b = y_n + \max \mathbf{Y}_{b-n} - y_n \max \mathbf{Y}_{b-n}, \tag{B.2}$$

and substituting in Eq. (B.1) (using also the Jakkola bound [26]) arises

$$\begin{aligned}\log \mathrm{q}(y_n) = &\, y_n \mathbb{E}_{\mathrm{q}(f^L_n)}\big[f^L_n\big]\\ &+ y_n \log H(2T_b - 2T_b\mathbb{E}_{\mathrm{q}(\mathbf{Y}_{b-n})}[\max\{\mathbf{Y}_{b-n}\}]\\ &+ \mathbb{E}_{\mathrm{q}(\mathbf{Y}_{b-n})}[\max\{\mathbf{Y}_{b-n}\}] - 1) + \text{const}.\end{aligned} \tag{B.3}$$

We use the following approximation as in [16],

$$\mathbb{E}[\max\{y_i\}] \approx \max\{\mathbb{E}[y_i]\}, \tag{B.4}$$

to finally obtain the optimal update for $\mathrm{q}(\mathbf{y})$,

$$\mathrm{q}_n \leftarrow \sigma\Big(\mathbb{E}_{\mathrm{q}(f^L_n)}\big[f^L_n\big] + \log H \cdot (2T_b + \max \mathbf{q}_{b-n} - 2T_b \max \mathbf{q}_{b-n} - 1)\Big). \tag{B.5}$$

B2. ELBO derivation

Using Eq. (B.4), the $\mathrm{ELBO}(\mathbf{V},\Theta,\{\mathbf{Z}^{l-1}\}_{l=1}^L)$ is finally approximated by

$$\begin{aligned}\mathrm{ELBO} = &\sum_{n=1}^{N}\mathbb{E}_{\mathrm{q}(y_n)\mathrm{q}(f^L_n)}\big[\log \mathrm{p}(y_n|\mathbf{f}^L_n)\big] + \sum_{b=1}^{B}\sum_{n\in b}\mathbb{E}_{\mathrm{q}(y_n)}\left[\log\frac{H^{G_b}}{H+1}\right]\\ &- \sum_{n=1}^{N}\mathbb{E}_{q(y_n)}[\log \mathrm{q}(y_n)] - \sum_{l=1}^{L}\mathbb{E}_{\mathrm{q}(\mathbf{U}^l)}\left[\log\frac{\mathrm{q}(\mathbf{U}^l)}{\mathrm{p}(\mathbf{U}^l)}\right]\\ &\approx \sum_{n=1}^{N} q_n \mathbb{E}_{\mathrm{q}(f^L_n)}\big[\log \mathrm{p}(y_n = 1|f^L_n)\big]\\ &+ (1-q_n)\mathbb{E}_{\mathrm{q}(f^L_n)}\big[\log \mathrm{p}(y_n = 0|f^L_n)\big]\\ &+ \log H \sum_{b=1}^{B}(2T_b \max \mathbf{q}_b - \max \mathbf{q}_b)\\ &- \sum_{n=1}^{N} q_n \log q_n + (1-q_n)\log(1-q_n) - \sum_{l=1}^{L}\mathrm{KL}\big(\mathrm{q}(\mathbf{U}^l)||\mathrm{p}(\mathbf{U}^l)\big)\\ &+ \text{const}.\end{aligned} \tag{B.6}$$

B3. Deep Gaussian process estimation

We can compute analytically the posterior for $\{\mathbf{F}^l\}_{l=1}^L$ by marginalizing the inducing variables from each layer:

$$\mathrm{q}(\{\mathbf{F}^l\}_{l=1}^L) = \prod_{l=1}^{L} \mathrm{q}(\mathbf{F}^l|\mathbf{m}^l, \mathbf{S}^l; \mathbf{F}^{l-1}, \mathbf{Z}^{l-1}) = \prod_{l=1}^{L} \mathcal{N}(\mathbf{F}^l|\tilde{\boldsymbol{\mu}}^l, \tilde{\boldsymbol{\Sigma}}^l), \tag{B.7}$$

where $[\tilde{\boldsymbol{\mu}}^l]_n = \mu_{\mathbf{m}^l,\mathbf{Z}^{l-1}}(\mathbf{f}_n^{l-1})$ and $[\tilde{\boldsymbol{\Sigma}}^l]_{ij} = \Sigma_{\mathbf{S}^l,\mathbf{Z}^{l-1}}(\mathbf{f}_i^{l-1}, \mathbf{f}_j^{l-1})$. The explicit expression for the mean vector $\tilde{\boldsymbol{\mu}}^l$ and the covariance matrix $\tilde{\boldsymbol{\Sigma}}^l$ can be found in [20, Eqs. (7–8)]. We are able to compute the ith marginal at each layer because it only depends on the corresponding ith input of the previous layer. This allows to sample from the last layer $\mathbf{f}_i^L$ by recursively sampling from all the previous layers $\hat{\mathbf{f}}_i^1 \to \hat{\mathbf{f}}_i^2 \to \cdots \to \hat{\mathbf{f}}_i^L$. This can be easily performed by means of univariate Gaussians. We first sample a $\varepsilon_i^l \sim \mathcal{N}(0,1)$ and then for $l = 1, \ldots, L$:

$$\hat{\mathbf{f}}_i^l = \mu_{\mathbf{m}^l,\mathbf{Z}^{l-1}}(\hat{\mathbf{f}}_i^{l-1}) + \varepsilon_i^l \cdot \sqrt{\Sigma_{\mathbf{S}^l,\mathbf{Z}^{l-1}}(\hat{\mathbf{f}}_i^{l-1}, \hat{\mathbf{f}}_i^{l-1})}. \tag{B.8}$$

Since we can sample from the posterior distribution in the last layer, the expectation $\mathbb{E}_{\mathrm{q}(f_n^L)}[\log \mathrm{p}(y_n|f_n^L)]$ in the ELBO (see Eq. (B.6)) can be approximated with a Monte Carlo sample generated with Eq. (B.8). Similarly, we can compute the expectation $\mathbb{E}_{\mathrm{q}(f_n^L)}[f_n^L]$ in the update of the $\mathrm{q}(\mathbf{Y})$, see Eq. (B.5). For scalability, we can use mini-batches in the optimization since the ELBO factorizes across data points.

Once the model is trained and the ELBO optimized, we can make a prediction for new test point $\mathbf{X}_*$. For this, we sample S times from the posterior using Eq. (B.8). In this case, we use the test location as initial input. This yields a set $\{\mathbf{f}_*^{L-1}(s)\}_{s=1}^S$ with S samples. Then, the density over f_*^L is given by the Gaussian mixture (recall that all the terms in Eq. (B.7) are Gaussian):

$$\mathrm{q}(f_*^L) = \frac{1}{S}\sum_{s=1}^{S} \mathrm{q}(f_*^L|\mathbf{m}^L, \mathbf{S}^L; \mathbf{f}_*^{L-1}(s), \mathbf{Z}^{L-1}). \tag{B.9}$$

Appendix C. Additional results

Here, we report additional tables with results. These tables are commented in the main text but we included them here for better readability.

Table C.1
Mean results for 5 different runs of DGPMIL2 with 8-dimensional input features. The results are for both RSNA and CQ500 datasets. We study the metrics for a varying number of inducing points $\tilde{M}$.

	$\tilde{M}$	F1 score	AUC-ROC	AUC-PR
RSNA	10	0.829±0.018	0.953±0.012	0.954±0.014
	50	0.834±0.016	0.954±0.01	0.96±0.008
	200	0.839±0.006	0.957±0.011	0.961±0.011
	500	0.835±0.006	0.956±0.012	0.962±0.009
CQ 500	10	0.714±0.02	0.899±0.01	0.853±0.026
	50	0.734±0.024	0.911±0.012	0.887±0.024
	200	0.735±0.022	0.909±0.005	0.889±0.011
	500	0.731±0.026	0.913±0.01	0.893±0.009

Table C.2
Mean results for 5 different runs of DGPMIL2 with 8-dimensional input features. The results are for both RSNA and CQ500 datasets. We study the metrics for a varying number of dimensions D in the latent space.

	D	F1 score	AUC-ROC	AUC-PR
RSNA	3	0.839±0.006	0.957±0.011	0.961±0.011
	10	0.837±0.008	0.957±0.09	0.964±0.006
	50	0	0.5±0	0.48±0
CQ 500	3	0.735±0.022	0.909±0.005	0.889±0.011
	10	0.733±0.022	0.914±0.013	0.902±0.0279
	50	0	0.5±0	0.418±0

References

[1] D. Kushner, Mild traumatic brain injury: toward understanding manifestations and treatment, Arch. Intern. Med. 158 (15) (1998) 1617.
[2] J.A. Caceres, J.N. Goldstein, Intracranial hemorrhage, Emerg. Med. Clin. North Am. 30 (3) (2012) 771–794.
[3] C.A. Taylor, Traumatic brain injury-related emergency department visits, hospitalizations, and deaths – United States, 2007 and 2013, Surveill. Summ. 66 (2017). Morbidity and Mortality Weekly Report (MMWR)
[4] W.M. Strub, J.L. Leach, T. Tomsick, A. Vagal, Overnight preliminary head CT interpretations provided by residents: locations of misidentified intracranial hemorrhage, Am. J. Neuroradiol. 28 (9) (2007) 1679–1682.
[5] W.K. Erly, W.G. Berger, E. Krupinski, J.F. Seeger, J.A. Guisto, Radiology resident evaluation of head CT scan orders in the emergency department, Am. J. Neuroradiol. 23 (1) (2002) 103–107.
[6] 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: International Conference on Machine Learning and Soft Computing, 2017, pp. 34–39.
[7] J. Cho, K.-S. Park, M. Karki, E. Lee, S. Ko, J.K. Kim, D. Lee, J. Choe, J. Son, M. Kim, S. Lee, J. Lee, C. Yoon, S. Park, Improving sensitivity on identification and delineation of intracranial hemorrhage lesion using cascaded deep learning models, J. Digit. Imaging 32 (3) (2019) 450–461.
[8] 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.
[9] K. Jnawali, M.R. Arbabshirani, N. Rao, A.A.P. M.d, Deep 3D convolution neural network for CT brain hemorrhage classification, in: Medical Imaging 2018: Computer-Aided Diagnosis, vol. 10575, International Society for Optics and Photonics, 2018, p. 105751C.
[10] M.-A. Carbonneau, V. Cheplygina, E. Granger, G. Gagnon, Multiple instance learning: a survey of problem characteristics and applications, Pattern Recognit. 77 (2018) 329–353.
[11] G. Campanella, M.G. Hanna, L. Geneslaw, A. Miraflor, V. Werneck Krauss Silva, K.J. Busam, E. Brogi, V.E. Reuter, D.S. Klimstra, T.J. Fuchs, Clinical-grade computational pathology using weakly supervised deep learning on whole slide images, Nat. Med. 25 (8) (2019) 1301–1309.
[12] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7132–7141.
[13] M. Ilse, J. Tomczak, M. Welling, Attention-based deep multiple instance learning, in: International Conference on Machine Learning - ICML, 2018, pp. 2127–2136.
[14] Z. Han, B. Wei, Y. Hong, T. Li, J. Cong, X. Zhu, H. Wei, W. Zhang, Accurate screening of COVID-19 using attention-based deep 3D multiple instance learning, IEEE Trans. Med. Imaging 39 (8) (2020).
[15] 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.
[16] 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.
[17] Y. Wu, A. Schmidt, E. Hernndez-Snchez, 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 MICCAI, 2021, pp. 582–591.
[18] J. Hensman, A.G. de G Matthews, Z. Ghahramani, Scalable variational gaussian process classification, Artificial Intelligence and Statistics (AISTATS), vol. 38, 2015.
[19] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning), The MIT Press, 2006.
[20] H. Salimbeni, M. Deisenroth, Doubly stochastic variational inference for deep gaussian processes, in: Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 4588–4599.
[21] A. Damianou, N. Lawrence, Deep Gaussian processes, in: International Conference on Artificial Intelligence and Statistics, vol. 31, 2013, pp. 207–215.
[22] ngel E. Esteban, M. Lpez-PĘrez, A. Colomer, M.A. Sales, R. Molina, V. Naranjo, A new optical density granulometry-based descriptor for the classification of prostate histological images using shallow and deep Gaussian processes, Comput. Methods Programs Biomed. 178 (2019) 303–317.

[23] S. Sun, W. Dong, Q. Liu, Multi-view representation learning with deep gaussian processes, IEEE Trans. Pattern Anal. Mach. Intell. 43 (12) (2021) 4453–4468.
[24] K. Saab, J. Dunnmon, R. Goldman, A. Ratner, H. Sagreiya, C. RĘ, D. Rubin, Doubly weak supervision of deep learning models for head CT, in: Medical Image Computing and Computer Assisted Intervention MICCAI 2019, vol. 11766, 2019, pp. 811–819.
[25] E. Snelson, Z. Ghahramani, Sparse Gaussian processes using pseudo-inputs, Advances in Neural Information Processing Systems, vol. 18, MIT Press, 2006.
[26] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
[27] 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.
[28] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: International Conference on Learning Representations - ICLR, 2015.
[29] 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) 1–7.
[30] 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.
[31] 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.
[32] N.T. Nguyen, D.Q. Tran, N.T. Nguyen, H.Q. Nguyen, A CNN-LSTM architecture for detection of intracranial hemorrhage on CT scans, Med. Imaging Deep Learn. (MIDL) (2020).
[33] Y. Zhu, L. Tong, S.R. Deshpande, M.D. Wang, Improved prediction on heart transplant rejection using convolutional autoencoder and multiple instance learning on whole-slide imaging, in: International Conference on Biomedical Health Informatics (BHI), 2019, pp. 1–4.
[34] M.K. Titsias, Variational learning of inducing variables in sparse gaussian processes, in: Artificial Intelligence and Statistics (AISTATS), 5, 2009, pp. 567–574.