Highlights

- End-to-end multiple instance learning (MIL) method for hemorrhage detection.
- It combines attention layer, CNN and Gaussian Process for robust training.
- The end-to-end MIL approach outperforms the two-phase MIL training strategy.

An End-to-end Approach to Combine Attention Feature Extraction and Gaussian Process Models for Deep Multiple Instance Learning in CT Hemorrhage Detection

Jose Pérez-Cano[a,*], Yunan Wu[b], Arne Schmidt[a], Miguel López-Pérez[a], Pablo Morales-Álvarez[c], Rafael Molina[a], Aggelos K. Katsaggelos[b]

[a]*Department of Computer Science and Artificial Intelligence, University of Granada, 18010 Granada, Spain.*
[b]*Department of Electrical Computer Engineering, Northwestern University, Evanston, IL, 60208 USA.*
[c]*Department of Statistics and Operations Research, University of Granada, 18010 Granada, Spain.*

Abstract

Intracranial hemorrhage (ICH) is a serious life-threatening emergency caused by blood leakage inside the brain. Radiologists usually confirm the presence of ICH by analyzing computed tomography (CT) scans, so, developing an automated diagnosis system that can process this type of images has become an important research problem. One of the main challenges to apply AI algorithms in this setting is the lack of labelled data. To mitigate the labeling burden, Multiple Instance Learning (MIL) algorithms group instances into bags, relying solely on bag-level labels for model training. Due to their capacity to handle uncertainty and deliver accurate predictions, Gaussian Processes (GPs) stand out as promising classifiers for MIL problems. Recent research has also demonstrated the effectiveness of combining attention mechanisms with GPs for ICH detection. Nonetheless, existing methods have a notable limitation: they train the attention mechanism and the GP separately, resulting in suboptimal feature

*Corresponding author.
Email addresses: joseperez2000@hotmail.es (Jose Pérez-Cano), yunanwu2020@u.northwestern.edu (Yunan Wu), arne@decsai.ugr.es (Arne Schmidt), mlopez@decsai.ugr.es (Miguel López-Pérez), pablomorales@ugr.es (Pablo Morales-Álvarez), rms@decsai.ugr.es (Rafael Molina), a-katsaggelos@northwestern.edu (Aggelos K. Katsaggelos)

extraction for GP-based classification. In this study, we introduce an innovative end-to-end MIL model that concurrently trains the CNN backbone and attention mechanism along with the GP classifier. Our approach enhances the robustness and accuracy of bag predictions by optimizing feature extraction for GP-based classification. We validate our method experimentally by focusing on two ICH detection datasets. Our results reveal a significant performance advantage in terms of accuracy, F1-score, precision, and ROC-AUC score over existing MIL approaches, especially two-stage GP approaches. Additionally, we offer empirical insights into the functionality and effectiveness of our novel model.

Keywords: End-to-end Multiple Instance Learning; Gaussian Process; Attention; CT hemorrhage detection

---

1. Introduction

Intracranial hemorrhage (ICH) is a significant medical emergency, with an annual rate of nearly 20 cases per 100,000 people (Rajashekar and Liang, 2020), accounting for 26% of all global strokes each year (Krishnamurthi et al., 2020). ICH carries a relatively low 1-year survival rate of 43.5% (Huang and Chen, 2021) and only 12% to 39% of survivors achieve full recovery (An et al., 2017). To address this critical issue, Computer-Assisted Diagnosis (CAD) aims to facilitate the triage process, assisting radiologists in swiftly and accurately identifying ICH cases to save more people's lives.

Previous studies have predominantly employed supervised deep learning models for the detection of head hemorrhages in computed tomography (CT) scans, with each scan slice individually labeled. For instance, Chang et al. (Chang et al., 2018) and Chilamkurthy et al. (Chilamkurthy et al., 2018) both leveraged 2D Convolutional Neural Networks (CNN) to make ICH predictions at the slice level. Phong et al. (Phong et al., 2017) conducted a more extensive investigation, exploring various CNN architectures for slice-level ICH predictions. However, a significant limitation of these methodologies is their dependency on

labeled data for each individual slice within a scan. The manual labeling of each slice is not only cost-intensive but also exceedingly time-consuming, resulting in a scarcity of large datasets with comprehensive slice labels.

Since the global label appears in clinical reports, recent approaches aim to only use scan labels for training to avoid additional labeling effort. Two research lines using only scan labels are supervised 3D CNN and MIL methods. In the case of 3D CNN the whole scan is considered as the input to the network. K. Jnawali et al. (Jnawali et al., 2018), and Titano et al. (Titano et al., 2018) used these 3D CNNs for ICH classification with satisfying results. The main problem with 3D CNNs is that they require a large amount of memory. Moreover, they cannot localize the injury in the scan. MIL methods overcome these problems by processing each slice individually and then aggregating either the features or the predictions (Carbonneau et al., 2016). They can infer intermediate labels for the slices, helping locate the injury and need less memory.

There are several recent examples of the successful application of MIL approaches to medical images. Campanella et al. (Campanella et al., 2019) applied a recurrent neural network as the aggregation method to combine the predictions for each image of the bag. They achieved good results for the classification of different cancer types in histopathological images. Since the aggregation determines the prediction of the whole bag, the focus of the literature has been placed on how to perform this aggregation. (Li et al., 2019) employed top-k pooling as their aggregation scheme obtaining satisfying results on several image datasets, and (Bi et al., 2021) proposed a local pyramid perception module that emphasizes the key instances from the local scale, and a global perception module that provides a spatial weight distribution from a global scale to classify retinal disease. Recently, the success of attention layers has brought popularity to its application in the MIL setting (Ilse et al., 2018). The attention mechanism was also used in medical imaging for MIL, for example, to diagnose COVID-19 from chest CT (Han et al., 2020), to classify COVID-19 from normal pneumonia (Qi et al., 2021) and for cancer survival prediction using Whole Slide Images (WSIs) (Yao et al., 2020).

All the methods discussed so far are deterministic which are prone to overconfident predictions and to overfitting when data is scarce (Gawlikowski et al., 2021). In medicine, this can have dramatic consequences because an inaccurate diagnosis can lead to a wrong treatment of the patient. Probabilistic methods like Gaussian Processes (GPs) have been proposed to overcome these limitations and provide more accurate predictions. Furthermore, GPs are robust to overfitting and have good generalization capability. For supervised tasks, Wilson et al. (Wilson et al., 2016), showed that combining GP and CNNs end-to-end, the so-called Deep Kernel Learning (DKL) paradigm, was better than previous works on different benchmarks like ImageNet or CIFAR. In the medical image domain, DKL was used by Wu et al. (Wu et al., 2021b) achieving a good performance in Bone Age Prediction and Lesion Localization. For MIL, the VGPMIL (Variational Gaussian Processes for MIL) model has shown promising results when classifying histological images of Barrett's cancer (Haußmann et al., 2017). Y. Wu et al. (Wu et al., 2021a) combined VGPMIL with CNN to improve the results obtained when using only the CNN. Here, the training was performed in two different steps. First, a CNN with an attention mechanism was trained to extract features and, in a second stage, these features were fed into the probabilistic model VGPMIL (Haußmann et al., 2017). M. López-Pérez et al. (López-Pérez et al., 2022) proposed an improvement of this approach by concatenating several GPs leading to the model of Deep Gaussian Processes for Multiple Instance Learning (DGPMIL). This classifier was more expressive and obtained better results in ICH detection than most published work in the literature. The problem with both approaches was that they trained the model in two phases, therefore not taking full advantage of the combination of CNN, attention mechanism, and GP. This two-stage approach has several major drawbacks: (i) the features are not optimized for the GP classifier because it is not trained end-to-end, (ii) the attention mechanism is discarded in the second training phase and during prediction and (iii) the training procedure becomes more complex because it consists of two stages (both requiring, for example, hyperparameter tuning, model convergence and model saving).

This work proposes a probabilistic end-to-end model for ICH detection with scan labels that overcomes the above mentioned problems. The proposed model combines a 2D CNN feature extractor, attention mechanism (Ilse et al., 2018), and GPs in an end-to-end fashion for the MIL problem. The main contributions of our work are as follows:

- In contrast to other methods that use CNNs and the attention mechanism (Ilse et al., 2018; Bi et al., 2020), we explore and propose how to include the probabilistic GPs, the attention mechanism, and the CNN in an end-to-end manner. To the best of our knowledge, this is the first time that these three modules have been trained end-to-end in a MIL problem.

- We design two different architectures, which differ in the position of the attention layer, before or after the GP. Both architectures are compared theoretically and empirically to analyse the advantages and disadvantages of each one of them. In addition, we compare our strategy to other state-of-the-art methods and find that the end-to-end training outperforms the two-stage training and other previous approaches.

- We also provide insightful ablation studies for the further understanding of the model. Moreover, we find that high attention weights are correlated with positive slice labels, helping to locate the slices affected by hemorrhage, which is of high importance in the diagnostic process.

The rest of the paper is structured as follows. In section 2 we outline the methods and theory, in section 3 we report details about the experiments and the experimental outcomes and we conclude our work in section 4.

2. Methods/Theory

In this section we introduce the proposed approach, along with the necessary background to understand it. Specifically, in Section 2.1 we present the main notation and the problem formulation. In Sections 2.2 and 2.3 we introduce two MIL methods that are at the core of our proposal. Whereas the former uses a

deep learning based attention mechanism, the latter leverages GPs. In Section 2.4 we explain how these two approaches have been already combined to produce a two-stage model. Finally, in Section 2.5 we present our main contribution: a novel model that combines both algorithms in a end-to-end manner.

2.1. Notation and problem formulation

Our notation follows the standard one used in most state-of-the-art MIL approaches (Ilse et al., 2018; Haußmann et al., 2017; Wang and Pinar, 2021). The training data is given by a set of bags $\mathbf{X} = \{\mathbf{X}_b\}_{b\in\mathcal{B}}$ and their corresponding labels $\mathbf{y} = \{y_b\}_{b\in\mathcal{B}}$. We deal with a binary problem, i.e. $y_b \in \{0, 1\}$ for all $b \in \mathcal{B}$. Each bag $\mathbf{X}_b = \{\mathbf{x}_i\}_{i\in b}$ contains $|b|$ instances, i.e. $b = \{i_1, \ldots, i_{|b|}\} \subseteq [N]$ (N is the total amount of instances). In the MIL setting, one assumes that each instance has a label, which is unknown. The MIL labelling assumption dictates that a bag is considered positive (class 1) if at least one of its instances is positive.

In the particular case of CT scans that we will tackle here, each $\mathbf{X}_b$ is a complete CT scan, which is composed by its slices $\{\mathbf{x}_i\}_{i\in b}$. Each slice has an unknown label (0 for non-hemorrhage and 1 for hemorrhage), and we only have access to the bag label y_b (whether the scan is positive or not, i.e. whether it contain at least one slice that is positive). The goal is to train a model based only on bag labels $\{y_b\}_{b\in\mathcal{B}}$, i.e. not having access to slice-level labels.

Notice that each slice $\mathbf{x}_i$ is an array with shape (W, H, C), where W is the weight of the image, H is its height, and C is the number of channels. For computational efficiency, we will assume that all the scans have the same amount of slices. This can be achieved by adding slices full of zeros when necessary. In the experimental section we will see that the attention mechanism is able to correctly handle these artificial slices full of zeros.

2.2. Convolutional neural networks for MIL

Convolutional neural networks (CNNs) are widely used in the field of image processing. A standard CNN needs a target value for each training instance.

However, this information is not available in the MIL scenario, where we only have labels at bag level. Therefore, an aggregation mechanism is required to obtain bag-level predictions. Then, a loss between such predictions and true bag labels must be minimized.

Two important families of aggregation mechanisms have been proposed to tackle this issue. Some approaches are based on *instance-level* aggregations, where the prediction is made at instance level and such values are aggregated to yield the bag prediction. This is the case of Additive MIL (Javed et al., 2022), which obtains patch-wise class contributions, and ProMIL (Łukasz Struski et al., 2023), which aims to identify the optimal percentage of instance-level predictions required for a positive bag prediction. However, here we will focus on *embedding-level* aggregation, which has been advocated as preferable in terms of the bag level classification performance in previous work Wang et al. (2018); Ilse et al. (2018); Schmidt et al. (2023). The idea in this family of methods is to aggregate the embeddings (or features) of the instances, and then obtain a prediction for such aggregated embedding.

In our proposal we will rely on the most popular approach within embedding-level methods, which is given by (Ilse et al., 2018). They propose an attention-based aggregation mechanism, which allows detection of the most relevant instances that trigger the bag label. Specifically, the aggregated embedding is given by a weighted average of the instance embeddings, i.e. $\mathbf{h}_{\text{agg}} = \sum_{i \in b} a_i \cdot \mathbf{h}_i$. The weights a_i are calculated through an attention layer:

$$a_i = \frac{\exp\{\mathbf{w}^T \tanh(\mathbf{V}\mathbf{h}_i)\}}{\sum_{j \in b} \exp\{\mathbf{w}^T \tanh(\mathbf{V}\mathbf{h}_j)\}}, \tag{1}$$

where $\mathbf{w}$ and $\mathbf{V}$ are parameters to be estimated during training. In the sequel, this method will be referred to as Att-MIL.

2.3. Gaussian Processes for MIL

In the machine learning community, Gaussian Processes (GPs) are widely used in supervised problems due to their excellent capability to quantify uncertainty (Williams and Rasmussen, 2006). In the last years, GPs have been

extended to the MIL scenario, where uncertainty is of great importance (recall that instance labels are unknown). Among the different formulations that have been proposed for GP-MIL (Kim and De la Torre, 2010; Kandemir et al., 2016), the most popular approach nowadays is VGPMIL (Haußmann et al., 2017), which has been extended later in several directions, e.g. (Wang and Pinar, 2021; Yousefi et al., 2019).

VGPMIL leverages a GP classification model to describe the (unknown) instance labels. Then, the observed bag labels are modelled from such instance labels through a bag-likelihood that codifies the MIL assumption: a bag will be positive if at least one of its instances is positive (see (Haußmann et al., 2017, eq.(3)) for full details on the formulation). Moreover, since standard GPs are characterized by a restrictive $\mathcal{O}(N^3)$ computational complexity on the number of training instances N, VGPMIL resorts to sparse GPs (Snelson and Ghahramani, 2006).

The idea behind sparse GPs is to summarize the information contained in the N training instances through a set of $M \ll N$ inducing points $\mathbf{u} = \{u_m\}_{m=1}^{M}$. These values $\mathbf{u}$ are GP realizations at M locations $\mathbf{Z} = \{\mathbf{z}_m\}_{m=1}^{M}$, just like $\mathbf{f} = \{f_n\}_{n=1}^{N}$ are GP realizations at the input locations $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^{N}$. In order for the information to flow from $\mathbf{u}$ to $\mathbf{f}$, notice that the joint distribution on $(\mathbf{u}, \mathbf{f})$ is given by:

$$\mathrm{p}(\mathbf{u}) = \mathcal{N}(\mathbf{u}|0, \mathbf{K_{ZZ}}), \tag{2}$$

$$p(\mathbf{f}|\mathbf{u}) = \mathcal{N}(\mathbf{f}|\mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{u}, \mathbf{K_{XX}} - \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}}), \tag{3}$$

where $\mathbf{K}_{AB}$ refers to the GP kernel evaluated on datasets A and B, i.e. $\mathbf{K}_{AB} = \kappa(A, B)$.

2.4. The two-stage strategy

The strengths of deep learning and GPs are complementary. Whereas the former can extract abstract features that lead to accurate predictions, the latter provides uncertainty estimation that guarantees robustness and allows for reliable decision-making (Khan et al., 2019). Consequently, in the last years there

has been a growing effort to combine both approaches in standard supervised problems (i.e. no MIL). For instance, deep kernel learning (Wilson et al., 2016) and deep Gaussian processes (Salimbeni et al., 2019; Svendsen et al., 2020) have obtained promising results, and they are very active research fields nowadays (Ober et al., 2021; Ober and Aitchison, 2021).

In the field of MIL, the first attempt to combine deep learning and GPs was recently presented in (Wu et al., 2021a). In addition to the aforementioned complementarity of both techniques, the motivation for the authors of (Wu et al., 2021a) was to develop a method to apply VGPMIL on images. Indeed, notice that directly feeding VGPMIL with images is challenging, since GPs struggle when dealing with high-dimensional input spaces (as those implied by images) (Blomqvist et al., 2019). Consequently, the work (Wu et al., 2021a) provides a two-stage algorithm to first extract features from each instance with Att-MIL (described in Section 2.2), and then feed those features to an uncertainty-aware model such as VPGMIL (described in Section 2.3). Notice that the features extracted during the first stage are considered fixed/frozen for the second stage.

In this paper, the method introduced in (Wu et al., 2021a) will be referred to as 2SS (two-stage strategy). More specifically, two variants are proposed for 2SS in (Wu et al., 2021a), depending on whether the extracted features are multiplied or not by the attention weights before being fed to VGPMIL. Following their notation, these variants will be called here 2SS-AL-Aw and 2SS-AL-nAw, respectively. The complete details for their proposal can be found in (Wu et al., 2021a, Section 2), in particular see their Figure 1.

Very recently, the two-stage strategy proposed in (Wu et al., 2021a) has been improved by using deep GPs instead of GPs. This method will be referred to as DGPMIL (López-Pérez et al., 2022), and it will be also included as a state-of-the-art baseline in the experiments.

2.5. End-to-end model

Since 2SS is trained in two stages, the features extracted in the first step may not be the optimal ones for the second step. Our goal is to introduce

an algorithm that performs an end-to-end training of CNNs and GPs in the context of MIL. Based on previous experience (think for instance in the general improvement obtained by deep learning over hand-crafted features (Goodfellow et al., 2016)), we hypothesize that such end-to-end process will ultimately lead to a better performance in practice.

Notice that there are three key components in 2SS: the CNN, the attention module, and the GP. The first two elements are trained in the first stage (they are part of Att-MIL), and the third one is trained in the second stage (it is part of VGPMIL). In our method we leverage the same three components. Naturally, the CNN is applied in the first place to perform the feature extraction. Depending on the order of the other elements, we propose two different end-to-end (E2E) approaches, which are described next.

E2E-Att-GP. In this case, the attention mechanism is used before the GP. Specifically, the attention module receives the features extracted by the CNN from each instance, and outputs a feature vector for each bag, following eq. (1). Notice that this is a deterministic feature vector. Then, a probabilistic transformation is performed through a sparse GP layer, to introduce uncertainty in the model. Each sparse GP is described in eqs. (2)–(3), where the GP is evaluated at the location of the extracted features of the bag, i.e., the weighted average of the instance features using the attention mechanism. Finally, we use a dense layer with sigmoid activation to obtain the probability that a bag is class one, which is denoted p_b. Figure 1 illustrates the described architecture.

E2E-GP-Att. In this case, the GP is applied before the attention module. Namely, a sparse GP layer transforms the features extracted by the CNN from each instance, yielding stochastic features at instance level. Each sparse GP is described in eqs. (2)–(3). To propagate this stochasticity, we leverage sampling from the GP output. Therefore, the attention module receives the GP realizations and outputs a new feature vector for each bag, using eq. (1). Finally, we use a dense layer with sigmoid activation to obtain the probability that the bag is class one, which is denoted p_b. Figure 2 illustrates the described architecture.

In order to train both models, we use the cross-entropy (CE) loss between

the observed bag labels y_b and the predicted probabilities at bag level p_b:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{B}\sum_{b\in\mathcal{B}}\left[y_b \log(p_b) + (1-y_b)\log(1-p_b)\right]. \tag{4}$$

However, it is well-known that using this loss alone usually leads to over-fitting. As a consequence, different regularizers have been proposed in deep learning (Kukacka et al., 2017). Here we leverage a regularizer based on the Kullback-Leibler (KL) divergence, which is common in the GP literature (Ruiz et al., 2019). Specifically, our loss is:

$$\mathcal{L} = (1-\alpha)\cdot\mathcal{L}_{\text{CE}} + \alpha\cdot \text{KL}(\text{q}(\mathbf{U})||\text{p}(\mathbf{U})), \tag{5}$$

where $\alpha \in (0, 1)$ regulates the weight of each term, $\mathcal{L}_{\text{CE}}$ is given by eq. (4), $\text{p}(\mathbf{U})$ is the prior distribution over the sparse GPs in the sparse GP layer, and $\text{q}(\mathbf{U})$ is the posterior distribution (which is a Gaussian with parameters to be estimated). The KL divergence between two distributions is always greater than zero, and equals zero if and only if both distributions coincide. Therefore, the KL regularizer encourages the posterior distribution over $\mathbf{U}$ to stay close to the prior one. The loss in eq. (5) is minimized *end-to-end* with respect to all the model parameters, including the weights of the CNN, the attention parameters, and the posterior GP parameters (i.e. the mean vectors and covariance matrices of $\text{q}(\mathbf{U})$). In the experiments we will evaluate the behavior of E2E-Att-GP and E2E-GP-Att for different values of α. In particular, the case $\alpha = 0.5$ corresponds to the minimization of the negative lower bound of the log-likelihood of the observations.

For the inference process, the GP in our end-to-end architecture acts as an additional layer that receives the input features of the previous layer and outputs realizations of a probability distribution to the subsequent layer. We follow standard variational inference for sparse Gaussian Processes as in Hensman et al. (2015). Namely, the procedure for inference is as follows:

1. The GP layer builds the prior distribution $\text{p}(\mathbf{u})$ of the inducing points with eq. (2). This distribution is computed using the kernel function and the inducing point locations.

2. The GP layer approximates the posterior distribution $p(\mathbf{f}|\mathbf{u},\mathbf{y})$ of the GP using the conditional prior in eq. (3), the approximated posterior q($\mathbf{U}$) and the input features. Notice that this can be done in closed form, since both $p(\mathbf{f}|\mathbf{u})$ and q($\mathbf{U}$) are Gaussian distributions.
3. By minimizing eq. (5), the GP layer estimates an approximation q($\mathbf{U}$).
4. Finally, the output of the GP layer is a sample of the approximated posterior distribution $p(\mathbf{f}|\mathbf{u},\mathbf{y})$, which is forwarded to the subsequent layer.

Finally, in order to make predictions on a new bag, we propagate it through the three different layers (CNN, Attention, and GP). Whereas the former two are deterministic, the latter is probabilistic, so samples are obtained (from the GP output distribution) and propagated all the way to the model output. The final prediction is the mean over the different samples. Also, note that the dropout included in the CNN layers is disabled during test time. The complete experimental details are provided in Section 3.2, as well as in the Appendix.

3. Results and Discussion

In this section we validate empirically the proposed method. In Section 3.1 we introduce the used data and its processing, and in section 3.2 we provide details about the experimental setup. Then, in section 3.3 we show ablation studies for the two proposed methods, and in section 3.4 we compare our methods to other state-of-the-art approaches. Finally, in section 3.5 we provide insights on the importance of the attention layer, and in section 3.6 we show some relevant visualizations to illustrate the behavior of our methods. The code with our implementation will be publicly available upon acceptance of the paper.

3.1. Dataset and preprocessing

Our model is trained against the dataset of head CT images from the 2019 Radiological Society of North America (RSNA) challenge (https://kaggle.com/competitions/rsna-intracranial-hemorrhage-detection). Each tomography contains between 24 and 57 slice images. Due to the implementation constraints imposed by the Python

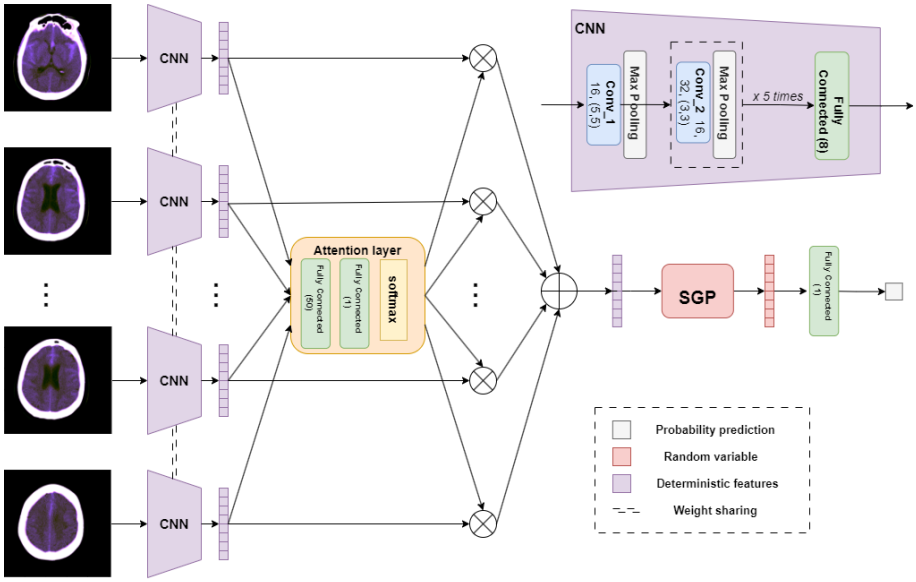


Figure 1: The proposed model architecture of E2E-Att-GP. Each slice of the CT scan is first processed by a feature extractor which consists of a Convolutional Neural Network (CNN) that is applied equally to each input slice in parallel. The resulting feature vectors enter an attention mechanism consisting of two Fully Connected (FC) layers and a Softmax (SM) layer, calculating one attention weight for each of them. The (attention-) weighted sum of the feature vectors then enters a Sparse Gaussian Process (SGP) and final FC layer for classification. The novel combination of attention mechanism and SGP can be trained end-to-end. The distinction between deterministic and stochastic features is highlighted by using purple and red colors, respectively. The digits in brackets $(\cdot)$ refer to the size and the number of kernels in the CNN layers.

libraries we use (i.e., GPflow and GPFlux), all bags need to have the same number of slices. To achieve this, we add black images to the bags whose number of slices is smaller than the maximum slice number to make sure all bags are in the same size of 57. Apart from that, each slice is also preprocessed (i.e., the windowing strategy) by using the same approach as described in (Wu et al., 2021a) to change the image brightness and apparent contrast to enhance the appearance of different types of tissues.

The model is trained on 1000 bags, with 411 positive values (i.e., ICH scans) and 589 negative values (i.e., normal scans). In addition, the testing is done

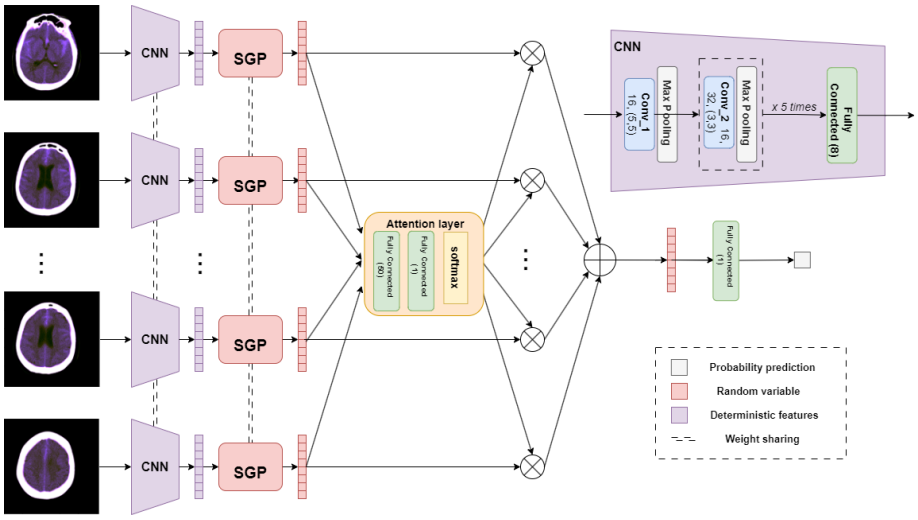
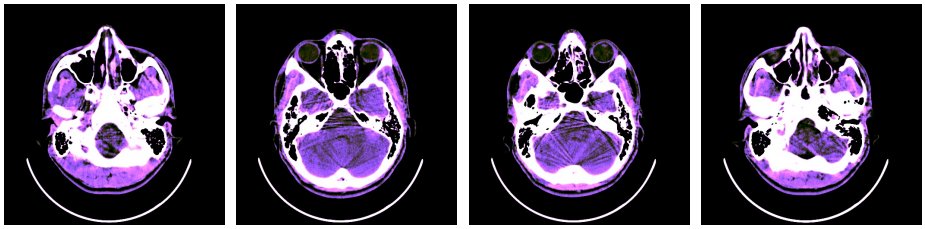


Figure 2: The proposed model architecture of E2E-GP-Att. In this variant, the Sparse Gaussian Process (SGP) follows directly after the Convolutional Neural Network (CNN). Both are applied equally to each input slice in parallel. The output of the SGP enters the attention layer which is analogous to the one used for E2E-Att-GP, see Figure 1. After combining the feature vectors with the attention weights and adding them up, a simple Fully Connected (FC) layer performs the final classification. The distinction between deterministic and stochastic features is highlighted by using purple and red colors, respectively. The digits in brackets $(\cdot)$ refer to the size and the number of kernels in the CNN layers.

using a separate dataset of 150 bags, with 72 positive values and 78 negative values. We use the same dataset as in (Wu et al., 2021a) and (López-Pérez et al., 2022). To stress the inherent difficulty of the problem, Figure 3 shows some examples of positive and negative slices as labelled by the clinicians. Notice that the distinction is not straightforward at all for a non-expert evaluator.

To assess the generalization capability of our model, we conduct evaluations on the external dataset CQ500 (Chilamkurthy et al., 2018), which comprises a total of 490 CT scans. These scans consist of 285 normal CT scans and 205 scans with annotations provided only at scan level. The number of slices within each scan varies, ranging from 16 to 128 slices. While maintaining consistent bag sizes during the training phase, we allow for variable bag sizes during testing. This approach ensures that the test set adheres to the same windowing strategy

Negative slices



Positive slices

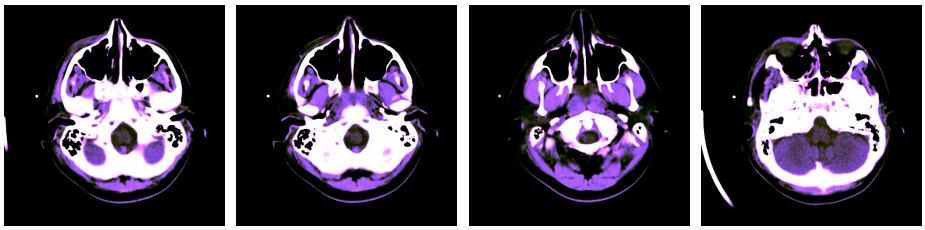


Figure 3: Some examples of positive and negative slices in the RSNA test set as labelled by expert clinicians. Notice that the task of ICH detection is a challenging one, since differences between positive and negative instances are not straightforward for a non-expert.

Table 1: The results for E2E-Att-GP. The values for each metric denote the mean and standard deviation from five experimental trials. The first column indicates the combination of hyperparameters used. The first number is the scaling factor α from eq. (5), and the second is the number of inducing points used by the sparse GP. The best results are highlighted in bold. The asterisk '*' means that the comparison with the performance of model pairs is statistically significant ($p < 0.05$).

Configuration	Accuracy	F1	Precision	Recall	ROC-AUC
0.1, 50	0.849 ± 0.014	0.875 ± 0.009	0.823 ± 0.020	0.936 ± 0.022	0.937 ± 0.006
0.5, 50	0.873 ± 0.020	0.872 ± 0.015	**0.844 ± 0.031***	0.902 ± 0.034	0.938 ± 0.005
0.9, 50	0.836 ± 0.031	0.878 ± 0.018	0.818 ± 0.033	0.947 ± 0.036	0.937 ± 0.009
0.1, 100	0.852 ± 0.010	0.875 ± 0.008	0.818 ± 0.027	0.943 ± 0.026	0.964 ± 0.003
0.5, 100	**0.876 ± 0.023***	**0.886 ± 0.011***	0.825 ± 0.032	0.959 ± 0.025	**0.965 ± 0.007***
0.9, 100	0.861 ± 0.018	0.879 ± 0.013	0.820 ± 0.029	0.947 ± 0.025	0.961 ± 0.006
0.1, 150	0.807 ± 0.009	0.830 ± 0.007	0.725 ± 0.010	**0.972 ± 0.012***	0.924 ± 0.003
0.5, 150	0.850 ± 0.010	0.869 ± 0.011	0.833 ± 0.023	0.909 ± 0.022	0.949 ± 0.005
0.9, 150	0.820 ± 0.021	0.875 ± 0.016	0.812 ± 0.049	0.949 ± 0.026	0.945 ± 0.010

preprocessing, without the inclusion of any black images.

3.2. Experimental design

Since the proposed model is stochastic, we need to run several trials for each experiment in order to obtain a robust value of each metric associated with the model. Hence, we run each experiment five times to obtain the mean and standard deviation for each metric.

Moreover, to have a deeper insight into how the model is performing, we use five different metrics in this classification task. The accuracy is the one with the most intuitive explanation. The Area Under the Receiver Operating Characteristic Curve (ROC-AUC) tells how well the model is separating the classes without giving any preference to any threshold. In addition, precision, recall, and F1 score, i.e., the harmonic mean between precision and recall, are used together to avoid the effects of the imbalanced dataset on the evaluation.

Furthermore, our approach involves optimizing two key hyperparameters: the number of the inducing points and the scaling factor α (as per Eq. (5)). In our ablation studies, we systematically explore various combinations of these hyperparameters to assess their impact on model performance. Furthermore, to

Table 2: The results for E2E-GP-Att. The values for each metric denote the mean and standard deviation from five experimental trials. The first column indicates the combination of hyperparameters used. The first number is the scaling factor α from eq. (5), and the second is the number of inducing points used by the sparse GP. The best results are highlighted in bold. The asterisk '*' means that the comparison with the performance of model pairs is statistically significant ($p < 0.05$).

Configuration	Accuracy	F1	Precision	Recall	ROC-AUC
0.1, 50	0.804 ± 0.053	0.849 ± 0.049	0.785 ± 0.089	0.937 ± 0.050	0.947 ± 0.007
0.5, 50	0.812 ± 0.041	0.825 ± 0.028	0.722 ± 0.056	**0.963 ± 0.021***	0.945 ± 0.006
0.9, 50	0.773 ± 0.059	0.800 ± 0.047	0.694 ± 0.087	0.944 ± 0.049	0.925 ± 0.014
0.1, 100	0.835 ± 0.028	0.843 ± 0.020	0.752 ± 0.075	0.959 ± 0.030	0.953 ± 0.011
0.5, 100	**0.856 ± 0.021***	**0.879 ± 0.015***	**0.823 ± 0.041***	0.944 ± 0.022	**0.964 ± 0.005***
0.9, 100	0.822 ± 0.030	0.839 ± 0.040	0.745 ± 0.092	0.961 ± 0.027	0.949 ± 0.012
0.1, 150	0.787 ± 0.055	0.804 ± 0.044	0.717 ± 0.079	0.917 ± 0.044	0.915 ± 0.023
0.5, 150	0.847 ± 0.029	0.872 ± 0.039	0.799 ± 0.062	0.960 ± 0.020	0.940 ± 0.017
0.9, 150	0.822 ± 0.040	0.877 ± 0.020	0.809 ± 0.047	0.958 ± 0.020	0.938 ± 0.014

rigorously compare the results obtained with different models, we employ the Wilcoxon signed-rank test (Rosner et al., 2006), a non-parametric statistical method. This test is used to determine whether the medians of two paired groups, based on their evaluation metric results, exhibit statistically significant differences. We consider the comparison to be statistically significant if the calculated p-value is less than 0.05.

Other experimental details include the Adam optimizer choice with an initial learning rate of 10^{-4}, and the early stopping criterion to decide when to stop the training process with a patience of 8. We split the training dataset into training (75%) and validation (25%) according to the number of bags. The evaluation is based on the ROC-AUC score, so whenever the score does not improve for 8 consecutive epochs in the validation dataset, the training process is completed. The training and testing processes are performed on three GPUs (Nvidia Quadro RTX 8000) using Tensorflow 2.7 and Python 3.7.

3.3. Results

The results are shown in Table 1 for E2E-Att-GP and in Table 2 for E2E-GP-Att. In both tables, the first column shows the combination of different

Table 3: Comparison between the best performing E2E-Att-GP, the best performing E2E-GP-Att and other state-of-the-art results on the same RSNA dataset. We can clearly see that both of our architectures outperform current state-of-the-art models, while E2E-Att-GP performs slightly better than E2E-GP-Att. The asterisk '*' means that the comparison with the performance of model pairs is statistically significant ($p < 0.05$).

	Methods with the same dataset				
Methods	Accuracy	F1	Precision	Recall	ROC-AUC
E2E-Att-GP	**0.876 ± 0.023***	**0.886 ± 0.011***	**0.825 ± 0.032**	0.959 ± 0.025	**0.965 ± 0.007**
E2E-GP-Att	0.856 ± 0.021	0.879 ± 0.015	0.823 ± 0.041	0.944 ± 0.022	0.964 ± 0.005
DGPMIL(López-Pérez et al., 2022)	0.825 ± 0.006	0.839 ± 0.006	N/A	N/A	0.957 ± 0.011
2SS-AL-nAw(Wu et al., 2021a)	0.780 ± 0.089	0.814 ± 0.059	0.705 ± 0.099	0.975 ± 0.025	0.964 ± 0.006
2SS-AL-Aw(Wu et al., 2021a)	0.743 ± 0.176	0.794 ± 0.104	0.705 ± 0.172	0.944 ± 0.043	0.951 ± 0.010
Att-MIL	0.781 ± 0.023	0.811 ± 0.017	0.694 ± 0.023	**0.975 ± 0.021**	0.951 ± 0.011

	Other methods with different dataset		
Methods	Model	Labeling Dimension	ROC-AUC
Sato et al.(Sato et al., 2018)	Convolutional auto-encoder	3D scans	0.87
Arbabshirani et al.(Arbabshirani et al., 2018)	starightforward CNNs	3D scans	0.85
Titano et al. (Titano et al., 2018)	ResNet 50	3D scans	0.73
Saab et al. (Saab et al., 2019)	Multiple instance learning	3D scans	0.91
Patel et al.(Patel et al., 2019)	VGG-like + LSTM	2D slices	0.96
Chang et al.(Chang et al., 2018)	Mask R-CNN like	2D slices	0.98

values of the scaling factor α and number of inducing points. For example, "$0.1, 50$" means that the scaling factor is 0.1 and the number of inducing points is 50. Other columns show the mean and deviation for each metric for five experimental trials, with the best performing model highlighted in bold for each metric.

In both tables, the results show that the scaling factor of 0.5 for a given number of inducing points achieves the best result for most metrics. In other words, the model achieves the best performance when the KL divergence is of equal importance as the cross-entropy. Another interesting finding is that the number of inducing points affects the performance of the model in a non-linear way, which indicates that more inducing points do not necessarily result in better performance. This is positive because using more inducing points requires more computational power. In this study, we found that 150 was the maximum number of inducing points we could use without getting out of memory, while the best performing model only required 100 inducing points for both tables.

Table 4: Comparison between the best performing E2E-Att-GP, the best performing E2E-GP-Att and other state-of-the-art method on the same external testing CQ500 dataset. The asterisk '*' means that the comparison with the performance of model pairs is statistically significant ($p < 0.05$).

Metrics	2SS-AL-nAw (Wu et al., 2021a)	Att-MIL (Ilse et al., 2018)	DGPMIL (López-Pérez et al., 2022)	E2E-Att-GP	E2E-GP-Att
Accuracy	0.639±0.106	0.655±0.043	0.717±0.035	**0.796±0.018***	0.756±0.023
F1	0.693±0.058	0.700±0.023	0.735±0.022	**0.785±0.017***	0.736±0.031
ROC-AUC	0.906±0.010	0.906±0.010	0.909±0.005	**0.918±0.003***	0.912±0.003
Cohen's kappa	0.335±0.171	0.359±0.069	0.461±0.059	**0.594±0.035***	0.509±0.048

Table 5: Metrics of the models substituting the attention layer by the maximum aggregation. The variability is reduced by sampling 100 different measures from the GP and averaging the results at the end.

Metrics	Accuracy	F1	Precision	Recall	ROC-AUC
E2E-Att-GP 100 samples	0.480 ± 0	0.649 ± 0	0.480 ± 0	1 ± 0	0.625 ± 0.003
E2E-GP-Att 100 samples	0.505 ± 0.003	0.659 ± 0.001	0.492 ± 0.002	1 ± 0	0.759 ± 0.002

3.4. Comparisons with the state of the art

First, we compare our model with others using exactly the same training and testing data cohorts, including DGPMIL (López-Pérez et al., 2022), the model with two separate training steps (2SS-AL-nAw, 2SS-AL-Aw) (Wu et al., 2021a) and Att-MIL (i.e., only the CNN model with Att layer), as shown in Table 3. We can observe that E2E-Att-GP and E2E-GP-Att outperform DGPMIL, 2SS-AL-nAw and 2SS-AL-Aw in most metrics with an accuracy of 0.876, F1 score of 0.886, precision of 0.825 and ROC-AUC of 0.965 while the difference between E2E-Att-GP and E2E-GP-Att is significant in accuracy and F1 score but not in precision and ROC-AUC. Although the recall is worse than Att-MIL and 2SS-AL-nAw, our method significantly outperforms them in all other metrics, which shows that our approach is less prone to false positives so it produces better predictions in general. The comparison with Att-MIL shows that GP plays a significant role in improving model performance. In addition, the direct comparison with 2SS-AL-nAw and 2SS-AL-Aw shows that the model with an end-to-end training strategy performs better than the same model architecture

but with two separate training steps. The reason for this is that, the end-to-end training optimizes the parameters of the CNNs, attention layer and GP together while if they are trained separately, the features extracted from CNNs may not be optimal for GP.

Regarding the model complexity, it is in the same order of magnitude as the baselines. The number of parameters is 152,487 for E2E-Att-GP and 95,359 for E2E-GP-Att. As a comparison, the number of parameters for DGPMIL is 122,346 parameters. Furthermore, the proposed models are efficient and able to be run on consumer hardware. For instance, inference for one sample takes around 200ms on a MacBook Pro M1 Max for E2E-GP-Att and E2E-Att-GP.

To demonstrate the generalization capabilities of our approach, we resort to the external test dataset, CQ500 (Chilamkurthy et al., 2018). Notice that we are not training on this new dataset, but using the methods previously trained on the RSNA dataset. The results are included in Table 4. Notably, both E2E-Att-GP and E2E-GP-Att demonstrate outstanding performance, significantly better than other models across all evaluation metrics. Furthermore, E2E-Att-GP performs better than E2E-GP-Att, which aligns with the findings presented in Table 3. The results provided in Table 4 show the efficacy of our model in adapting to test CT scans acquired from diverse scanner settings, confirming the robust generalization capabilities inherent to the end-to-end training scheme.

We further compare our approach with other methods using different datasets in ICH diagnosis. As multiple different metrics are evaluated in these methods, we choose ROC-AUC score to compare the model performances. Since no available codes are recently public for these methods, it is challenging to train the other methods on our dataset to have a fair comparison. However, although different dataset are utilized, we can see that E2E-Att-GP and E2E-GP-Att outperform methods training with 3D scan labels (Sato et al., 2018; Arbabshirani et al., 2018; Titano et al., 2018; Saab et al., 2019) and have a comparable (Chang et al., 2018) or even a higher ROC-AUC score (Patel et al., 2019) compared to models using 2D slice labels. The results are promising, because manually labeling each slice is tedious and time-consuming for radiologists, while the scan

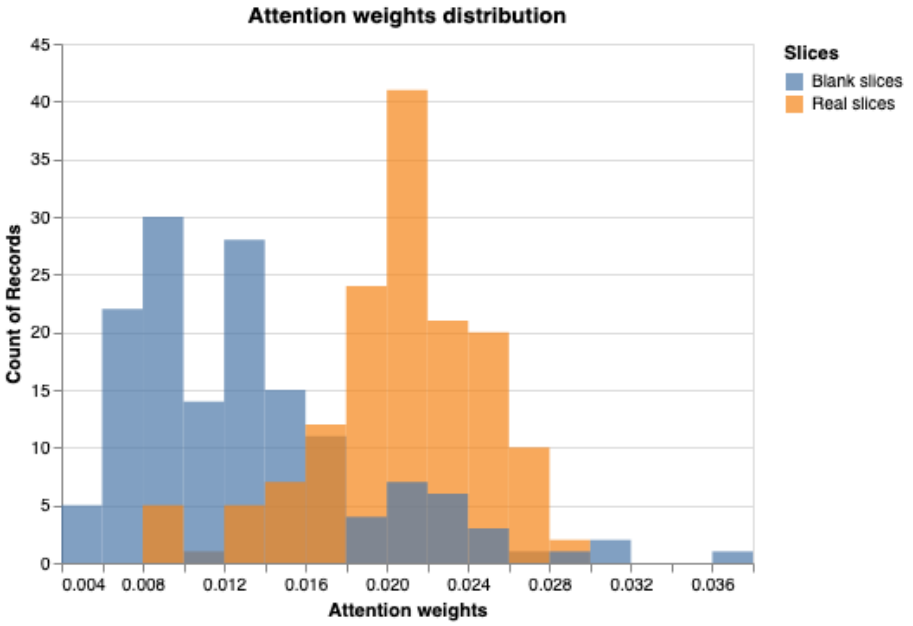


Figure 4: Histogram representing the distributions of the attention weights separated by the type of slice. The values were computed by taking the average of the attention weights for real and artificial images separately in each bag.

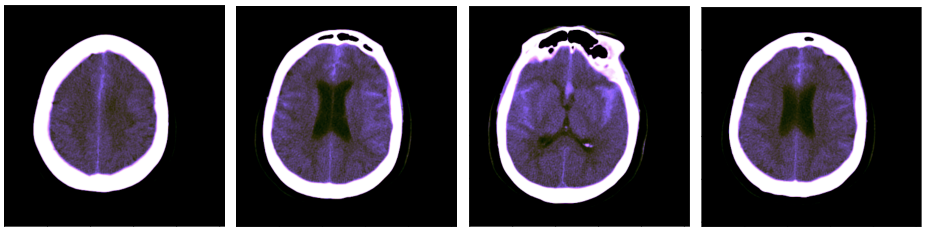
labels can be easily accessed from patients clinical reports. Therefore, since our method achieves a comparable performance on ICH diagnosis, they can greatly reduce the workload of radiologists and potentially, improve the clinical triage system.

3.5. The explainabilty of "attention"

In this section, we analyze the relevance of the attention layer as one of the three components of our end-to-end methodology. Overall, we have three findings from our experiments: 1. substituting the attention layer by the maximum aggregation worsens the result; 2. the attention weights are correlated with the slice labels; 3. the attention layer is correctly identifying the added black slices that fill the bag, which proves that they have no harmful effect on the training process.

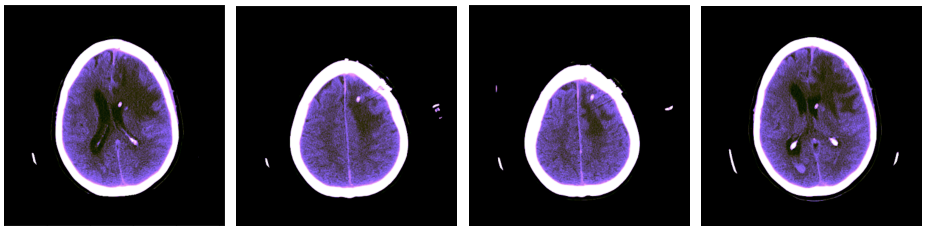
First, to show the importance of the attention layer, we utilize a way of

Scan 4



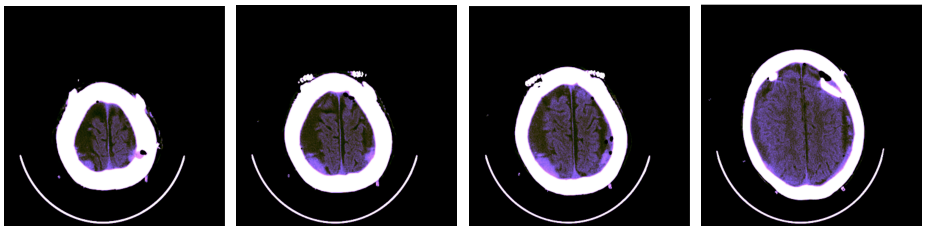
(a) 0.0582 (b) 0.0497 (c) 0.0475 (d) 0.0471

Scan 9



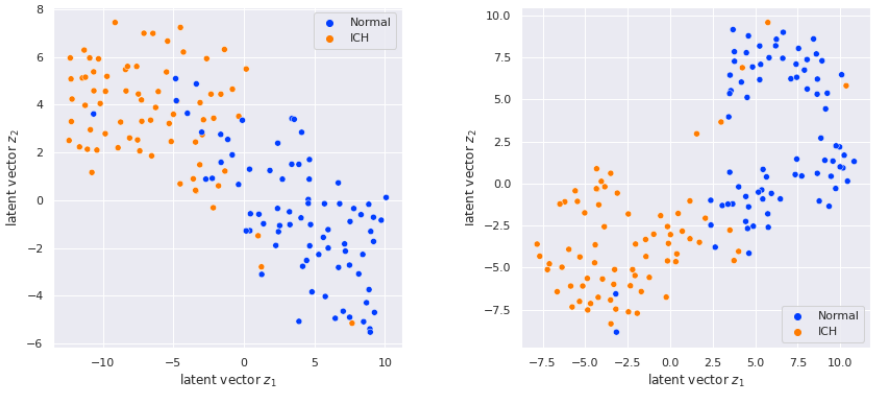
(e) 0.0649 (f) 0.0369 (g) 0.0366 (h) 0.0306

Scan 10



(i) 0.0835 (j) 0.0818 (k) 0.0644 (l) 0.0551

Figure 5: Each row corresponds to a different scan, and shows the four slices that were assigned the highest attention values in that scan. Importantly, all these slices are labelled as positive by the experts in the database, which stresses that there exists a correlation between high attention weight and the presence of ICH. The images are ordered from left to right by the estimated attention weight, which is shown below each image.



(a) E2E-GP-Att

(b) E2E-Att-GP

Figure 6: t-SNE of the latent vectors of scans as produced by the two models previous to applying the last layer to generate the probabilities.

substituting the attention layer by a maximum aggregation, which has been used in several studies as a way to work in MIL (Campanella et al., 2018, 2019). The modifications for the experiment consist of applying the layer after the attention layer (i.e., GP in E2E-Att-GP and a dense layer in E2E-GP-Att) equally to all slices, and then, applying the maximum operation to generate one outcome prediction. This modification is reasonable because the attention layer is designed to carefully select the slices, so the latent spaces of the image will remain the same without the attention layer. Therefore, the layers after the attention layer can be applied to the feature vectors before the attention layer, and we only add the maximum operation at the end to generate one final probability.

The results for the modified E2E-Att-GP and E2E-GP-Att model are shown in Table 5. We see a significant drop in performances for all metrics. A possible explanation for this is that hemorrhages normally do not just occur on one slice of the head but on several different regions of the brain. Slices are different sections of the same brain separated by a small spacing distance, so they are not independent from each other. For that reason, detecting hemorrhages is a matter of looking into several slices like a radiologist. However, by removing

the attention layer, the GP needs to detect the hemorrhage at each individual slice without taking into account any possible relationships between the slices, while the attention layer is the one that can weight on all the slices to find that relationship. For that reason, working at each slice individually without further looking at other slices makes the modified model lack this spatial information, leading to much worse performance compared to using the attention layer.

For our second finding, we analyze the relationship between the estimated attention values and the ground truth label (which is known for test instances). Similarly to previous MIL approaches using attention (Ilse et al., 2018; Schmidt et al., 2023; Wu et al., 2023), we expect that slices with ICH are assigned higher values of attention than those not presenting ICH. Indeed, we find that the mean of the attention weights for positive slices is 0.023, whereas for negatives it is 0.016. Notice that the average is 43% higher for positive slices. This quantitative result, which takes into account all the data, is complemented with some qualitative examples showing that the highest attention values in positive bags are usually assigned to positive slices, see Figure 5l.

Finally, the third interesting finding is that the attention layer can correctly identify which slices are artificially generated (i.e., the black slices). To prove this, we generate the attention weight for each slice of the testing dataset and take two mean values: one for the artificial slices and one for the real slices. Figure 4 illustrates their respective distributions. It is straightforward to find that the mean is higher for the real slices, demonstrating that the attention layer is assigning more attention to real slices. Moreover, in 81% of the bags, the mean values of the weights from the real slices are higher than those from the artificial slices. There is some variance shown in both distributions, but overall, the attention layer is able to correctly ignore the images that are added to the bags, which indicates that the attention layer is trained to learn important features.

3.6. Visualizations

To depict how well the classes are separated in their corresponding latent spaces of E2E-GP-Att and E2E-Att-GP, we show two t-distributed stochastic embedding (t-SNE) plots of the latent vectors by extracting features from the last fully connected layer. t-SNE is used to visualize high-dimensional feature vectors in a two or three dimensional map, so after t-SNE, the dimensionality of each scan reduces from 8 to 2. The result for E2E-GP-Att and for E2E-Att-GP is shown in Figure 6. Except for some outliers, each figure shows two well-separated clusters representing each class, meaning that the internal representations of bags the models have learned are discriminative enough for this classification task.

Apart from classifying bags, the model can also be used to detect which slices are more important for the model to be predicted as positive. The attention layer provides an score for each slice that can be interpreted as the probability of such slice to be positive, as shown in Figure 5. Therefore, the attention weights can potentially be used to indicate instance predictions by only training on bag labels.

4. Conclusions and future work

This work proposes two architectures that combine convolutional neural networks, an attention layer, and Gaussian processes with end-to-end training. We show that the end-to-end model performs better than previous works, trained in two separate phases. To the best of our knowledge, this is the first work that combines CNN, attention, and GP into one architecture in an end-to-end manner for multiple instance learning.

When evaluating the model with the RSNA dataset, we obtain a slightly better ROC-AUC score, and considerably higher accuracy and F1 than previous state-of-the-art models. We find that applying the attention module before the GP leads to better results than the other way around. However, both architectures score higher than previous approaches, showing that different com-

binations of CNN, attention layer, and GP achieve state-of-the-art predictions as long as they are trained end-to-end. These positive empirical findings also apply to the generalization capability of our approach, as shown in the external validation with the CQ500 dataset.

In the future, a more complex classifier with Deep Gaussian Processes could even achieve a higher performance when trained end-to-end with the feature extractor and attention mechanism. Furthermore, the use of more sophisticated attention mechanisms, as well as alternative DL models, is an interesting line of research to further improve the accuracy. Other challenging MIL scenarios, such as utilizing 3D patches, should be addressed in future work too.

Acknowledgements

This work was supported by FEDER/Junta de Andalucía-Consejería de Transformación Económica, Industria, Conocimiento y Universidades through project P20_00286, Spanish Ministry of Science and Innovation through project PID2022-140189OB-C22, and the European Union's Horizon 2020 research and innovation programme under the Marie Skodowska Curie grant agreement No 860627 (CLARIFY Project). The work of Miguel López Pérez has been supported by the University of Granada postdoctoral program "Contrato Puente".

References

An, S. J., Kim, T. J., and Yoon, B.-W. (2017). Epidemiology, risk factors, and clinical features of intracerebral hemorrhage: An update. *J Stroke*, 19(1):3–10.

Arbabshirani, M. R., Fornwalt, B. K., Mongelluzzo, G. J., Suever, J. D., Geise, B. D., Patel, A. A., and Moore, G. J. (2018). Advanced machine learning in action: identification of intracranial hemorrhage on computed tomography scans of the head with clinical workflow integration. *NPJ digital medicine*, 1(1):1–7.

Bi, Q., Qin, K., Li, Z., Zhang, H., Xu, K., and Xia, G.-S. (2020). A multiple-instance densely-connected convnet for aerial scene classification. *IEEE Transactions on Image Processing*, 29:4911–4926.

Bi, Q., Yu, S., Ji, W., Bian, C., Gong, L., Liu, H., Ma, K., and Zheng, Y. (2021). Local-global dual perception based deep multiple instance learning for retinal disease classification. In de Bruijne, M., Cattin, P. C., Cotin, S., Padoy, N., Speidel, S., Zheng, Y., and Essert, C., editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, pages 55–64, Cham. Springer International Publishing.

Blomqvist, K., Kaski, S., and Heinonen, M. (2019). Deep convolutional gaussian processes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 582–597. Springer.

Campanella, G., Hanna, M. G., Geneslaw, L., Miraflor, A., Werneck Krauss Silva, V., Busam, K. J., Brogi, E., Reuter, V. E., Klimstra, D. S., and Fuchs, T. J. (2019). Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. *Nat Med*, 25(8):1301–1309.

Campanella, G., Silva, V. W. K., and Fuchs, T. J. (2018). Terabyte-scale deep multiple instance learning for classification and localization in pathology. *CoRR*, abs/1805.06983.

Carbonneau, M., Cheplygina, V., Granger, E., and Gagnon, G. (2016). Multiple instance learning: A survey of problem characteristics and applications. *CoRR*, abs/1612.03365.

Chang, P. D., Kuoy, E., Grinband, J., Weinberg, B. D., Thompson, M., Homo, R., Chen, J., Abcede, H., Shafie, M., Sugrue, L., Filippi, C. G., Su, M.-Y., Yu, W., Hess, C., and Chow, D. (2018). Hybrid 3D/2D convolutional neural network for hemorrhage evaluation on head CT. *AJNR Am J Neuroradiol*, 39(9):1609–1616.

Chilamkurthy, S., Ghosh, R., Tanamala, S., Biviji, M., Campeau, N. G., Venugopal, V. K., Mahajan, V., Rao, P., and Warier, P. (2018). Development and validation of deep learning algorithms for detection of critical findings in head CT scans. *CoRR*, abs/1803.05854.

Gawlikowski, J., Tassi, C. R. N., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A. M., Triebel, R., Jung, P., Roscher, R., Shahzad, M., Yang, W., Bamler, R., and Zhu, X. X. (2021). A survey of uncertainty in deep neural networks. *CoRR*, abs/2107.03342.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Han, Z., Wei, B., Hong, Y., Li, T., Cong, J., Zhu, X., Wei, H., and Zhang, W. (2020). Accurate screening of COVID-19 using Attention-Based deep 3D multiple instance learning. *IEEE Trans Med Imaging*, 39(8):2584–2594.

Haußmann, M., Hamprecht, F. A., and Kandemir, M. (2017). Variational bayesian multiple instance learning with gaussian processes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6570–6579.

Hensman, J., De G. Matthews, A., and Ghahramani, Z. (2015). Scalable variational Gaussian process classification. In *International conference on artificial intelligence and statistics*, pages 351–360.

Huang, C. and Chen, J.-C. (2021). The Long-Term survival of intracranial hemorrhage patients successfully weaned from prolonged mechanical ventilation. *Int J Gen Med*, 14:1197–1203.

Ilse, M., Tomczak, J., and Welling, M. (2018). Attention-based deep multiple instance learning. In *International Conference on Machine Learning - ICML*, pages 2127–2136.

Javed, S. A., Juyal, D., Padigela, H., Taylor-Weiner, A., Yu, L., and Prakash, A. (2022). Additive mil: Intrinsically interpretable multiple instance learning for pathology. In *Neural Information Processing Systems*.

Jnawali, K., Arbabshirani, M. R., Rao, N., and Patel, A. A. (2018). Deep 3d convolution neural network for ct brain hemorrhage classification. In *Medical Imaging*.

Kandemir, M., Haußmann, M., Diego, F., Rajamani, K. T., Van Der Laak, J., and Hamprecht, F. A. (2016). Variational weakly supervised gaussian processes. In Richard C. Wilson, E. R. H. and Smith, W. A. P., editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 71.1–71.12. BMVA Press.

Khan, M. E. E., Immer, A., Abedi, E., and Korzepa, M. (2019). Approximate inference turns deep networks into gaussian processes. *Advances in neural information processing systems*, 32.

Kim, M. and De la Torre, F. (2010). Gaussian processes multiple instance learning. In *ICML*.

Krishnamurthi, R. V., Ikeda, T., and Feigin, V. L. (2020). Global, regional and Country-Specific burden of ischaemic stroke, intracerebral haemorrhage and subarachnoid haemorrhage: A systematic analysis of the global burden of disease study 2017. *Neuroepidemiology*, 54(2):171–179.

Kukacka, J., Golkov, V., and Cremers, D. (2017). Regularization for deep learning: A taxonomy. *CoRR*, abs/1710.10686.

Li, S., Liu, Y., Sui, X., Chen, C., Tjio, G., Ting, D. S. W., and Goh, R. S. M. (2019). Multi-instance multi-scale CNN for medical image classification. In Shen, D., Liu, T., Peters, T. M., Staib, L. H., Essert, C., Zhou, S., Yap, P., and Khan, A. R., editors, *Medical Image Computing and Computer Assisted Intervention - MICCAI 2019 - 22nd International Conference, Shenzhen, China, October 13-17, 2019, Proceedings, Part IV*, volume 11767 of *Lecture Notes in Computer Science*, pages 531–539. Springer.

López-Pérez, M., Schmidt, A., Wu, Y., Molina, R., and Katsaggelos, A. K. (2022). Deep gaussian processes for multiple instance learning: Application

to ct intracranial hemorrhage detection. *Computer Methods and Programs in Biomedicine*, 219:106783.

Ober, S. W. and Aitchison, L. (2021). Global inducing point variational posteriors for bayesian neural networks and deep gaussian processes. In *International Conference on Machine Learning*, pages 8248–8259. PMLR.

Ober, S. W., Rasmussen, C. E., and van der Wilk, M. (2021). The promises and pitfalls of deep kernel learning. In *Uncertainty in Artificial Intelligence*, pages 1206–1216. PMLR.

Patel, A., Van De Leemput, S. C., Prokop, M., Van Ginneken, B., and Manniesing, R. (2019). Image level training and prediction: Intracranial hemorrhage identification in 3d non-contrast ct. *IEEE Access*, 7:92355–92364.

Phong, T. D., Duong, H. N., Nguyen, H. T., Trong, N. T., Nguyen, V. H., Van Hoa, T., and Snasel, V. (2017). Brain hemorrhage diagnosis by using deep learning. In *Proceedings of the 2017 International Conference on Machine Learning and Soft Computing*, ICMLSC '17, page 34–39, New York, NY, USA. Association for Computing Machinery.

Qi, S., Xu, C., Li, C., Tian, B., Xia, S., Ren, J., Yang, L., Wang, H., and Yu, H. (2021). DR-MIL: deep represented multiple instance learning distinguishes COVID-19 from community-acquired pneumonia in CT images. *Comput Methods Programs Biomed*, 211:106406.

Rajashekar, D. and Liang, J. W. (2020). Intracerebral hemorrhage.

Rosner, B., Glynn, R. J., and Lee, M.-L. T. (2006). The wilcoxon signed rank test for paired comparisons of clustered data. *Biometrics*, 62(1):185–192.

Ruiz, P., Morales-Álvarez, P., Molina, R., and Katsaggelos, A. K. (2019). Learning from crowds with variational gaussian processes. *Pattern Recognition*, 88:298–311.

Saab, K., Dunnmon, J., Goldman, R., Ratner, A., Sagreiya, H., Ré, C., and Rubin, D. (2019). Doubly weak supervision of deep learning models for head ct. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 811–819. Springer.

Salimbeni, H., Dutordoir, V., Hensman, J., and Deisenroth, M. (2019). Deep gaussian processes with importance-weighted variational inference. In *International Conference on Machine Learning*, pages 5589–5598. PMLR.

Sato, D., Hanaoka, S., Nomura, Y., Takenaga, T., Miki, S., Yoshikawa, T., Hayashi, N., and Abe, O. (2018). A primitive study on unsupervised anomaly detection with an autoencoder in emergency head ct volumes. In *Medical Imaging 2018: Computer-Aided Diagnosis*, volume 10575, page 105751P. International Society for Optics and Photonics.

Schmidt, A., Morales-Álvarez, P., and Molina, R. (2023). Probabilistic attention based on gaussian processes for deep multiple instance learning. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–14.

Snelson, E. and Ghahramani, Z. (2006). Sparse gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, volume 18.

Svendsen, D. H., Morales-Álvarez, P., Ruescas, A. B., Molina, R., and Camps-Valls, G. (2020). Deep gaussian processes for biogeophysical parameter retrieval and model inversion. *ISPRS Journal of Photogrammetry and Remote Sensing*, 166:68–81.

Titano, J. J., Badgeley, M., Schefflein, J., Pain, M., Su, A., Cai, M., Swinburne, N., Zech, J., Kim, J., Bederson, J., Mocco, J., Drayer, B., Lehar, J., Cho, S., Costa, A., and Oermann, E. K. (2018). Automated deep-neural-network surveillance of cranial images for acute neurologic events. *Nat Med*, 24(9):1337–1341.

Wang, F. and Pinar, A. (2021). The multiple instance learning gaussian process

probit model. In *International Conference on Artificial Intelligence and Statistics*, pages 3034–3042. PMLR.

Wang, X., Yan, Y., Tang, P., Bai, X., and Liu, W. (2018). Revisiting multiple instance neural networks. *Pattern Recognition*, 74:15–24.

Williams, C. and Rasmussen, C. (2006). *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA.

Wilson, A. G., Hu, Z., Salakhutdinov, R. R., and Xing, E. P. (2016). Stochastic variational deep kernel learning. *Advances in Neural Information Processing Systems*, 29.

Wu, Y., Castro-Macías, F. M., Morales-Álvarez, P., Molina, R., and Katsaggelos, A. K. (2023). Smooth attention for deep multiple instance learning: Application to ct intracranial hemorrhage detection. *arXiv preprint arXiv:2307.09457*.

Wu, Y., Schmidt, A., Hernández-Sánchez, E., Molina, R., and Katsaggelos, A. K. (2021a). Combining attention-based multiple instance learning and gaussian processes for CT hemorrhage detection. In *Medical Image Computing and Computer Assisted Intervention – MICCAI*, pages 582–591.

Wu, Z., Yang, Y., Gu, J., and Tresp, V. (2021b). Quantifying predictive uncertainty in medical image analysis with deep kernel learning. In *2021 IEEE 9th International Conference on Healthcare Informatics (ICHI)*, pages 63–72.

Yao, J., Zhu, X., Jonnagaddala, J., Hawkins, N., and Huang, J. (2020). Whole slide images based cancer survival prediction using attention guided deep multiple instance learning networks. *Medical Image Analysis*, 65:101789.

Yousefi, F., Smith, M. T., and Alvarez, M. (2019). Multi-task learning for aggregated data using gaussian processes. *Advances in Neural Information Processing Systems*, 32.

Łukasz Struski, Rymarczyk, D., Lewicki, A., Sabiniewicz, R., Tabor, J., and Zieliński, B. (2023). Promil: Probabilistic multiple instance learning for medical imaging.

5. Appendix

This section is devoted to provide all the technical details about the CNN, the attention layer, the GP and the fully connected layer at the end. It also provides important implementation details.

5.1. Reshaping

Since we are dealing with bags of instances, our tensors contain an additional dimension that has to be dealt with. The input tensor has a shape of $(16, 512, 512, 57, 3)$ where the first dimension is the batch, the next two are the height and width of the image, the fourth dimension is representing the number of slices per bag, which is set to 57 for all bags, and the last one is the number of channels. In order to apply the CNN to each instance separately, two transformations are made. First, we transpose the tensor to move the bag dimension to the beginning. And then, the batch and bag dimensions are joined. The code for that is

```
tf.reshape(tf.transpose(inp, perm=(0,3,1,2,4)), shape=(-1,dim[0],
    dim[1], 3))
```

This transformation is later undone in the attention layer.

5.2. CNN

Our CNN is composed of 6 convolutional layers, together with 6 batch normalization layers and 6 max pooling layers. It also has some dropout layers in between. All the max pool layers are the same, they use a 2 by 2 kernel with a stride of 2 in every direction to reduce the width and height by 2. In tensorflow they are written as

```
layers.MaxPool2D((2, 2),strides=(2, 2),data_format="channels_last")
```

The convolutional layers are all the same except for the first one. The first one uses a 5 by 5 kernel, generates 16 channels, returns an image of the same size (`padding='same'`), and is initialized with the method `glorot_uniform`. In tensorflow:

```
Conv1 = layers.Conv2D(16, (5, 5), data_format="channels_last",
    activation='relu', kernel_initializer='glorot_uniform', padding
    ='same')
```

The rest of the convolutional layers use a 3 by 3 kernel, generate 32 output channels with the default padding ('filter'), and the default initialization, which is also `glorot_uniform`. In tensorflow:

```
layers.Conv2D(32, (3,3),  data_format="channels_last", activation='
    relu')
```

After every convolutional layer, a batch normalization layer is immediately applied, and then a max pooling layer. However, the dropout is only applied in certain layers. Between layers 2 and 3, 5 and 6, and at the end. The dropout layer is added between the max pooling of one layer and the convolutional layer of the next. The dropout rate is set to 0.3 for all the layers.

5.3. Attention layer

The attention layer is simply implemented as two feed forward neural networks. One to account for the transformation of the latent vectors by the matrix $\mathbf{V}$, and another to account for the scalar product with the vector $\mathbf{w}$. After that, a softmax is applied. The code for that is

```
out = layers.Dense(D, activation='tanh')(inp)
out = layers.Dense(1, use_bias = False)(out)
out = tf.reshape(out, shape=(-1,dim[2]))
```

Where D is 50, the number of hidden neurons of that hidden layer between the CNN and the attention layer. Not to be confused with the D of section 2.1 representing the depth of the bag. The output above are the weights, but they need to be multiplied to each instance and averaged. Everything is done as a

tensor product with appropriate reshaping. That way we can take advantage of parallelization. The code is:

```
inp = keras.Input(shape=dim_)
H = layers.Flatten()(inp)
A = attention(H)
H = tf.reshape(H, shape=(-1,dim[2], H.shape[1]))
A = tf.expand_dims(A, axis=1)
intermediate = tf.linalg.matmul(A,H)
intermediate = tf.squeeze(intermediate, axis=1)
out = layers.Dense(8)(intermediate)
```

5.4. GP

Since we wanted to train the GPs with gradient descent it made sense to use two libraries that are an extension of tensorflow-probability: GPflow and GPFlux. The first one is an implementation of sparse GPs using tensorflow as the backend to train them. The second one is an implementation of Deep GPs using GPFlow that makes it possible to create GPs as a normal tensorflow layer, therefore making it easier to include them in any architecture. The only change we needed to do was to modify the class `GPLayerSeq` to add the scaling factor. The main changes in the code of the class are presented below:

```
log_prior = tf.add_n([p.log_prior_density() for p in self.kernel.
    trainable_parameters])
loss = self.prior_kl() # - log_prior
loss_per_datapoint = self.scale_factor * loss # / self.num_data
```

We commented out the `log_prior` because we are already using the binary cross-entropy as the likelihood term. And we also commented out `self.num_data` so that the KL loss was exactly what we described in the formulas in section 2.5.

5.5. Feed-forward neural network at the end

In the two models we have, at the end there is a neural network to pass from an 8 dimensional vector to a 1 dimensional vector (a scalar) representing the

probability of the bag being positive. The final layer is just a dense layer with one neuron and sigmoid as the activation function.