DISTRIBUTION BASED MIL POOLING FILTERS ARE SUPERIOR TO POINT ESTIMATE BASED COUNTERPARTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multiple instance learning (MIL) is a machine learning paradigm which learns the mapping between bags of instances and bag labels. There are different MIL tasks which can be solved by different MIL methods. One common component of all MIL methods is the MIL pooling filter, which obtains bag level representations from extracted features of instances. Here, we recommend and discuss a grouping scheme for MIL pooling filters: *point estimate based pooling filters* and *distribution based pooling filters*. The point estimate based pooling filters include the standard pooling filters, such as 'max', 'mean' and 'attention' pooling. The distribution based pooling filters include recently proposed 'distribution' pooling and newly designed 'distribution with attention' pooling. In this paper, we perform the first systematic analysis of different pooling filters. We theoretically show that the distribution based pooling filters are superior to the point estimate based counterparts in terms of amount of information captured while obtaining bag level representations from extracted features. Then, we empirically study the performance of the 5 pooling filters, namely 'max', 'mean', 'attention', 'distribution' and 'distribution with attention', on distinct real world MIL tasks. We showed that the performance of different pooling filters are different for different MIL tasks. Moreover, consistent with our theoretical analysis, models with distribution based pooling filters almost always performed equal or better than that with point estimate based pooling filters.

1 INTRODUCTION

Multiple instance learning (MIL) is a machine learning paradigm which learns the mapping between bags of instances and bag labels. MIL models are handy for tasks where data is in the form of bags of instances and only bag labels are provided. For example, medical image processing tasks are typical MIL tasks since an image can be treated as a bag of pixels (instances) and there usually exists only an image (bag) label without providing the particular region-of-interest (Zhu et al., 2017).

The standard MIL task is the *positive vs negative bag classification* task (Dietterich et al., 1997; Maron & Lozano-Pérez, 1998). This task asks if a bag contains the targeted object or not. Moreover, there are other MIL tasks in the literature as well, like *unique class count prediction* (Oner et al., 2020), *multi-class classification* (Feng & Zhou, 2017), *multi-task classification* (Yang et al., 2016) or *regression* (Zhang et al., 2018). In order to solve these MIL tasks, different methods have been developed. While some methods first classify instances inside bags and then pool the instances' scores by using an MIL pooling filter (Dietterich et al., 1997; Maron & Lozano-Pérez, 1998; Andrews et al., 2003; Zhang & Goldman, 2002), others first extract features of instances inside bags, obtain bag level representations by using an MIL pooling filter and finally classify the bags (Wang et al., 2018; Ilse et al., 2018; Oner et al., 2020). The common component in the two approaches is the MIL pooling filter. Here, we recommend a grouping scheme for MIL pooling filters: *point estimate based pooling filters* and *distribution based pooling filters*. The point estimate based pooling filters include the standard pooling filters, such as 'max' pooling (Wang et al., 2018; Wu et al., 2015; Feng & Zhou, 2017), 'mean' pooling (Wang et al., 2018; 2019) or 'attention' pooling (Ilse et al., 2018) (see Sec. 2.1.1). All the above pooling filters have the same nature; they obtain point estimates (like the mean) of the features of all instances in the bag.

Although point estimate based pooling filters perform well in many applications, they cannot capture full information inside the features of the instances. On the other hand, it is possible to capture

richer information with pooling filters that compute the distribution of the features of the instances. We call this type of pooling filters as *distribution based pooling filters*. Recently, for example, in order to obtain bag level representations, marginal distributions of the features are estimated by using kernel density estimation in Oner et al. (2020). Here, we denote this method as 'distribution' pooling. Furthermore, in this paper, we developed another distribution based pooling filter known as 'distribution with attention' pooling by incorporating an attention mechanism into 'distribution' pooling. (See Sec. 2.1.2 for the details of these two distribution based pooling filters.)

There are some studies empirically comparing point estimate based pooling filters in a certain application like sound event detection (Wang et al., 2019). However, they don't go beyond one specific application and have lack of theoretical analysis. In this paper, we perform a systematic study of both point estimate based and distribution based pooling filters. Precisely, we compare 5 filters, namely, 'max', 'mean', 'attention', 'distribution' and 'distribution with attention'. We theoretically show that distribution based pooling filters are superior to point estimate based counterparts in terms of amount of information captured while obtaining bag level representations from extracted features.

To test the real-life performance of pooling filters, we formulated five different MIL tasks using a lymph node metastases dataset in Oner et al. (2020). The lymph node metastases dataset consists of images cropped from histopathology slides of lymph node sections (Bejnordi et al., 2017) and has corresponding ground truth metastases segmentation masks. There are three types of images in this dataset: *fully normal* - all cells are normal, *fully metastases* - all cells are metastases and *boundary* - mixture of normal and metastases cells. We formulated five different MIL tasks on this dataset: (i) *positive vs negative bag classification*, (ii) *unique class count prediction*, (iii) *multi-class classification*, (iv) *multi-task classification* and (v) *regression* (see Sec. 4.1). For each MIL task, we trained five different models with five different MIL pooling filters and statistically compared their performances. Consistent with our theoretical analysis, distribution based pooling filters mostly outperformed their point estimate based counterparts. Lastly, we tested the performance of 'distribution' pooling filter on classical MIL datasets. It outperformed state-of-the-art MIL methods.

Hence, this paper has three main contributions:

1. We recommend and discuss a grouping scheme for MIL pooling filters: *point estimate based pooling filters* and *distribution based pooling filters*. Moreover, we newly designed a distribution based pooling filter, 'distribution with attention', by incorporating an attention mechanism into 'distribution' pooling filter.
2. We theoretically analyzed five different MIL pooling filters and showed that distribution based pooling filters are superior to point estimate based pooling filters in terms of amount of information captured while obtaining bag level representations from extracted features. Also, we showed that among the two distribution based pooling filters, 'distribution with attention' pooling filter is theoretically the best in this sense.
3. Consistent with our theoretical analysis, we experimentally showed that distribution based pooling filters have equal or better performance than point estimate based pooling filters both in MIL tasks defined on the lymph node metastases dataset and classical MIL datasets. Furthermore, the experiments showed that 'distribution with attention' pooling filter is better than 'distribution' pooling filter.

For the rest of the paper, Sec. 2 introduces our MIL framework and theoretical analysis of MIL pooling filters; Sec. 3 gives a brief literature review; Sec. 4 defines the experiments and presents the results and Sec. 5 concludes the paper.

2 Multiple Instance Learning Framework

In MIL paradigm, the objective is to predict a bag label Y for a given bag of instances $X = \{x_i | x_i \in \mathcal{I}, i = 1, 2, \cdots, N\}$ where $\mathcal{I}$ is the instance space and N is the number of instances inside the bag. Each instance $x_i \in X$ is endowed with an underlying label $y_i \in \mathcal{L}$, where $\mathcal{L}$ is the instance label space; however, these labels are inaccessible during training. The relation between bag label Y and instance labels $\{y_i | y_i \in \mathcal{L}, i = 1, 2, \cdots, N\}$ is imposed by the definition of MIL task. Note that although the number of instances in each bag can be different in real world, it is treated as constant in

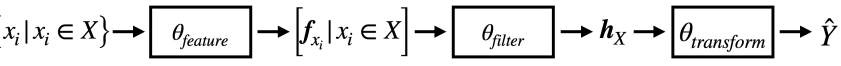


Figure 1: Block diagram of MIL framework. The *feature extractor* module θ_{feature} extracts a feature vector $\boldsymbol{f}_{x_i} \in \mathcal{F}$, $\forall x_i \in X$. Then, the *MIL pooling filter* module θ_{filter} aggregates the extracted feature vectors and obtains a bag level representation $\boldsymbol{h}_X \in \mathcal{H}$. Lastly, the *bag level representation transformation* module $\theta_{\text{transform}}$ transforms bag level representation into predicted bag label $\hat{Y} \in \mathcal{Y}$.

this paper for clarity of notation, yet all the properties stated in here are valid for bags with variable number of instances as well.

Let $\mathcal{D}$ be an MIL dataset such that $\mathcal{D} = \{(X, Y) \mid X \in \mathcal{X} \text{ and } Y \in \mathcal{Y}\}$, where $\mathcal{X} = \mathcal{I}^N$ is the bag space and $\mathcal{Y}$ is the bag label space. Given any pair $(X, Y) \in \mathcal{D}$, our objective is to predict bag label Y for a given bag of instances $X = \{x_i | x_i \in \mathcal{I}, i = 1, 2, \cdots, N\}$. Let $\hat{Y}$ be predicted bag label of X. We obtain $\hat{Y}$ by using a three stage framework, block diagram of which is given in Figure 1. The first stage is a *feature extractor* module $\theta_{\text{feature}} : \mathcal{I} \to \mathcal{F}$, where $\mathcal{F}$ is the feature space. For each $x_i \in X$, the *feature extractor* module takes x_i as input, extracts J features and outputs a feature vector: $\boldsymbol{f}_{x_i} = \theta_{\text{feature}}(x_i) = [f^j_{x_i} | f^j_{x_i} \in \mathbb{R}, j = 1, 2, \cdots, J] \in \mathcal{F}$ where $\mathcal{F} = \mathbb{R}^J$. Let $\boldsymbol{F}_X = [\boldsymbol{f}_{x_i} | \boldsymbol{f}_{x_i} \in \mathbb{R}^J, i = 1, \cdots N] \in \mathbb{R}^{JN}$ be feature matrix constructed from extracted feature vectors such that i^{th} column corresponds to $\boldsymbol{f}_{x_i}$. The second stage is an *MIL pooling filter* module $\theta_{\text{filter}} : \mathbb{R}^{JN} \to \mathcal{H}$, where $\mathcal{H}$ is the bag level representation space. The *MIL pooling filter* module takes the feature matrix $\boldsymbol{F}_X$ and aggregates the extracted feature vectors to obtain a bag level representation: $\boldsymbol{h}_X = \theta_{\text{filter}}(\boldsymbol{F}_X) \in \mathcal{H}$ where $\mathcal{H}$ depends on θ_{filter}. For example, 'max' pooling gets maximum value of each feature over the feature vectors of all instances ($\mathcal{H} = \mathbb{R}^J$) or 'distribution' pooling estimates the marginal distribution of each feature over the feature vectors of all instances ($\mathcal{H}$ is a distribution space). In this paper, we analyze the performance of 5 different MIL pooling filters (see Sec. 2.1) on 5 different MIL tasks. The last stage is a *bag level representation transformation* module $\theta_{\text{transform}} : \mathcal{H} \to \mathcal{Y}$. The *bag level representation transformation* module transforms bag level representation into predicted bag label: $\hat{Y} = \theta_{\text{transform}}(\boldsymbol{h}_X)$.

We use neural networks to implement θ_{feature} and $\theta_{\text{transform}}$ so that we can fully parameterize the learning process. For θ_{filter}, we use different filters, some of which also incorporate trainable components parameterized by neural networks. This system of neural networks is end-to-end trainable.

2.1 MIL POOLING FILTERS

An MIL pooling filter obtains a bag level representation by aggregating instance level representations. As a key component of an MIL model, the pooling filter provides the MIL model with two essential properties of MIL paradigm necessitated by set definition (note that a bag is nothing but a set). First, output of the model must be permutation-invariant, i.e. the model must produce the same output regardless of ordering of elements in the set. Second, the model must accept sets with variable sizes as input. These two properties are incorporated in our neural network models with the proper choice of the MIL pooling filters. Visual summary of pooling filters used in this paper is given in Figure 2. A simple controlled toy example is also given in Supp. D to emphasize the advantages of distribution based pooling filters over point estimate based ones.

2.1.1 POINT ESTIMATE BASED POOLING FILTERS

Given a feature matrix $\boldsymbol{F}_X = [f^j_{x_i} | f^j_{x_i} \in \mathbb{R},\ i = 1, 2, \cdots, N \text{ and } j = 1, 2, \cdots, J]$ obtained from a bag $X = \{x_1, x_2, \cdots, x_N\}$. The pooling filter aggregates the feature vectors of the N instances in the bag X and obtains a bag level representation $\boldsymbol{h}_X = [h^j_X \mid h^j_X \in \mathbb{R},\ j = 1, 2, \cdots, J] \in \mathcal{H}$ where $\mathcal{H} = \mathbb{R}^J$. For 'max', 'mean' and 'attention' pooling, the values h^j_X are:

- **Max pooling:** $h^j_X = \max_{i=1}^{N} f^j_{x_i}\ \forall_{j=1,2,\cdots,J}$.
- **Mean pooling:** $h^j_X = \frac{1}{N} \sum_{i=1}^{N} f^j_{x_i}\ \forall_{j=1,2,\cdots,J}$.

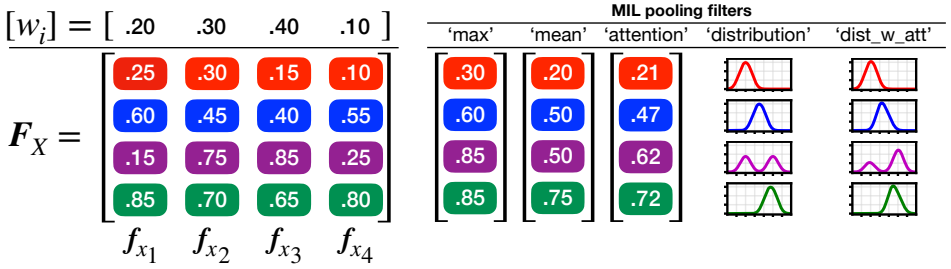


Figure 2: MIL pooling filters. The feature matrix $\boldsymbol{F}_X$ obtained from the bag $X = \{x_1, x_2, x_3, x_4\}$ contains 4 feature vectors $\boldsymbol{f}_{x_1}, \boldsymbol{f}_{x_2}, \boldsymbol{f}_{x_3}$ and $\boldsymbol{f}_{x_4}$. Each feature vector consists of 4 features highlighted with different colors. Moreover, attention weights of the instances are given in $[w_i]$. Outputs of different MIL pooling filters are shown. Note that the difference between 'distribution' and 'distribution with attention' ('dist_w_att') pooling is more visible for the 3^{rd} feature in purple.

- **Attention pooling (Ilse et al., 2018):** $h_X^j = \sum_{i=1}^{N} w_i f_{x_i}^j \ \forall_{j=1,2,\cdots,J}$. Each instance has an attention weight w_i obtained from a neural network module $\mathcal{W}$ such that $w_i = \frac{\exp\{\mathcal{W}(\boldsymbol{f}_{x_i})\}}{\sum_{t=1}^{N}\exp\{\mathcal{W}(\boldsymbol{f}_{x_t})\}} \ \forall_{i=1,2,\cdots,N}$. Note that instances' weights sum up to 1.

2.1.2 Distribution based pooling filters

While point estimate based pooling filters only summarize the extracted features by calculating point statistics, full information in the extracted features theoretically can be captured by estimating the joint distribution. However, it is computationally intractable. On the other hand, in order to capture as much information as possible, marginal distributions of extracted features can be estimated. Indeed, distribution based pooling filters obtain a bag level representation by estimating the marginal distribution of each extracted feature. Distribution based pooling filters have a notable advantage that they enable $\theta_{\text{transform}}$ module to fully utilize the information in shape of the distributions rather than looking at the point estimates as in 'max', 'mean' and 'attention' pooling. Moreover, the point estimates obtained by point estimate based pooling filters, in principle, can be fully recovered from the estimated marginal distributions obtained by distribution based pooling filters (see Sec. 2.2).

Distribution pooling (Oner et al., 2020): Given a feature matrix $\boldsymbol{F}_X = [f_{x_i}^j | f_{x_i}^j \in \mathbb{R},\ i = 1, 2, \cdots, N \text{ and } j = 1, 2, \cdots, J]$ obtained from a bag $X = \{x_1, x_2, \cdots, x_N\}$, its bag level representation is obtained by estimating the marginal distribution of each extracted feature. Let $\tilde{p}_X^j : \mathbb{R} \rightarrow \mathbb{R}^+ \cup \{0\}$ be the estimated marginal distribution of j^{th} extracted feature and $\tilde{p}_X^j \in \mathbb{P}$ where $\mathbb{P}$ is the set of all possible marginal distributions. $\tilde{p}_X^j$ is calculated by using kernel density estimation (Parzen, 1962), which employs a Gaussian kernel with standard deviation σ, as shown in the Eq. 1. Hence, the bag level representation $\boldsymbol{h}_X = [\tilde{p}_X^j \ | \tilde{p}_X^j \in \mathbb{P}, j = 1, 2, \cdots, J] \in \mathcal{H}$ where $\mathcal{H} = \mathbb{P}^J$. Note that the estimated marginal distributions are uniformly binned during training neural network models for computational purposes.

$$\tilde{p}_X^j(v) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\left(v - f_{x_i}^j\right)^2} \ \forall_{j=1,2,\cdots,J} \tag{1}$$

Distribution with attention pooling: We have extended 'distribution' pooling as in Eq. 2 by incorporating an attention mechanism in θ_{filter} module. While all of the instances contribute equally to estimate the distribution (with a weight of $1/N$, see Eq. 1) in 'distribution' pooling, each instance $x_i \in X$, in this method, has an attention based weight w_i obtained from a neural network module $\mathcal{W}$ such that $w_i = \frac{\exp\{\mathcal{W}(\boldsymbol{f}_{x_i})\}}{\sum_{t=1}^{N}\exp\{\mathcal{W}(\boldsymbol{f}_{x_t})\}} \ \forall_{i=1,2,\cdots,N}$. Note that instances' weights sum up to 1. Our motivation in associating an attention weight with each instance in a bag is that all instances may not contribute equally to the bag label. In such cases, it is important to prioritize some of the instances inside the bag while obtaining the bag level representation. For example, in positive vs negative bag classification task (see Sec. 2.3), even one 'positive' instance makes a bag 'positive' and it is crucial to capture this instance while obtaining bag level representation. This can easily be achieved

in our 'distribution with attention' pooling filter by assigning high attention weight to that instance. Similarly, bag level representation $\boldsymbol{h}_X = [\tilde{p}_X^j \,|\tilde{p}_X^j \in \mathbb{P}, j = 1, 2, \cdots, J] \in \mathcal{H}$ where $\mathcal{H} = \mathbb{P}^J$.

$$\tilde{p}_X^j(v) = \sum_{i=1}^{N} w_i \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\left(v - f_{x_i}^j\right)^2} \; \forall_{j=1,2,\cdots,J} \tag{2}$$

2.2 THEORETICAL ANALYSIS OF MIL POOLING FILTERS

This section is reserved to theoretically show that distribution based pooling filters are superior to their point estimate based counterparts in terms of amount of information captured while obtaining bag level representations from extracted features.

Proposition 1 *Given two feature matrices obtained from bags $X_a = \{x_1^{(a)}, x_2^{(a)}, \cdots, x_N^{(a)}\}$ and $X_b = \{x_1^{(b)}, x_2^{(b)}, \cdots, x_N^{(b)}\}$;*

- $\boldsymbol{F}_{X_a} = [f^j_{x_i^{(a)}} | f^j_{x_i^{(a)}} \in \mathbb{R},\ f^j_{x_i^{(a)}} \neq f^j_{x_u^{(a)}}\ \forall_{i \neq u},\ i, u = 1, 2, \cdots, N \text{ and } j = 1, 2, \cdots, J]$
- $\boldsymbol{F}_{X_b} = [f^j_{x_i^{(b)}} | f^j_{x_i^{(b)}} \in \mathbb{R},\ f^j_{x_i^{(b)}} \neq f^j_{x_u^{(b)}}\ \forall_{i \neq u},\ i, u = 1, 2, \cdots, N \text{ and } j = 1, 2, \cdots, J]$

and two pooling filters; 'max' pooling filter θ_{filter}^{max} and 'distribution' pooling filter θ_{filter}^{dist}. Let ${}_{max}\boldsymbol{h}_{X_a}$ and ${}_{max}\boldsymbol{h}_{X_b}$ be bag level representations obtained by θ_{filter}^{max} from $\boldsymbol{F}_{X_a}$ and $\boldsymbol{F}_{X_b}$, respectively. Similarly, let ${}_{dist}\boldsymbol{h}_{X_a}$ and ${}_{dist}\boldsymbol{h}_{X_b}$ be bag level representations obtained by θ_{filter}^{dist} from $\boldsymbol{F}_{X_a}$ and $\boldsymbol{F}_{X_b}$, respectively. If ${}_{max}\boldsymbol{h}_{X_a} \neq {}_{max}\boldsymbol{h}_{X_b}$, then ${}_{dist}\boldsymbol{h}_{X_a} \neq {}_{dist}\boldsymbol{h}_{X_b}$.

In Proposition 1, we showed that given two feature matrices, whenever 'max' pooling filter produces two different representations, so does 'distribution' pooling filter. This proposition states that, for example, given two bags from different classes and the corresponding feature matrices satisfying the conditions of Proposition 1, if 'max' pooling filter produces two different bag level representations, then 'distribution' pooling filter also produces two different bag level representations. In other words, whenever two bags from different classes are discriminated by 'max' pooling, they are also discriminated by 'distribution' pooling. However, converse of the Proposition 1 is not true. Given two different bag level representations obtained by 'distribution' pooling filter does not guarantee that bag level representations obtained by 'max' pooling filter are different since the feature matrices may still have the same maximum feature values for both of the bags. That is, there are cases that two bags from different classes are discriminated by 'distribution' pooling, but they are not discriminated by 'max' pooling. Hence, we conclude that 'distribution' pooling filter is superior to 'max' pooling filter in terms of amount of information captured in bag level representations.

Proposition 2 *Given a feature matrix $\boldsymbol{F}_X = [f^j_{x_i} | f^j_{x_i} \in \mathbb{R},\ i = 1, 2, \cdots, N \text{ and } j = 1, 2, \cdots, J]$ obtained from a bag $X = \{x_1, x_2, \cdots, x_N\}$, 'mean' pooling filter θ_{filter}^{mean} and 'distribution' pooling filter θ_{filter}^{dist}. Let ${}_{mean}\boldsymbol{h}_X = [{}_{mean}h_X^j \mid {}_{mean}h_X^j \in \mathbb{R},\ j = 1, 2, \cdots, J]$ and ${}_{dist}\boldsymbol{h}_X = [\tilde{p}_X^j \,|\tilde{p}_X^j \in \mathbb{P}, j = 1, 2, \cdots, J]$ be bag level representations obtained from $\boldsymbol{F}_X$ by θ_{filter}^{mean} and θ_{filter}^{dist}, respectively. Then, ${}_{mean}h_X^j = \mathbb{E}[V^j]\ \forall_{j=1,2,\cdots,J}$ where $V^j \sim \tilde{p}_X^j$. Note that $\mathbb{P}$ is the set of all possible marginal distributions.*

Proposition 3 *Given a feature matrix $\boldsymbol{F}_X = [f^j_{x_i} | f^j_{x_i} \in \mathbb{R},\ i = 1, 2, \cdots, N \text{ and } j = 1, 2, \cdots, J]$ obtained from a bag $X = \{x_1, x_2, \cdots, x_N\}$, 'attention' pooling filter θ_{filter}^{att} with attention weights $w_i > 0\ \forall_i$, $\sum_{i=1}^N w_i = 1$ and 'distribution' pooling filter θ_{filter}^{dist}. Let ${}_{att}\boldsymbol{h}_X = [{}_{att}h_X^j \mid {}_{att}h_X^j \in \mathbb{R},\ j = 1, 2, \cdots, J]$ be bag level representation obtained from $\boldsymbol{F}_X$ by θ_{filter}^{att}. If attention weights are accessible, then ${}_{att}h_X^j = N \times \mathbb{E}[V^j]\ \forall_{j=1,2,\cdots,J}$ where $V^j \sim \tilde{p}_X^j$ and ${}_{dist}\boldsymbol{h}_X = [\tilde{p}_X^j \,|\tilde{p}_X^j \in \mathbb{P}, j = 1, 2, \cdots, J]$ is the bag level representation obtained by θ_{filter}^{dist} from $\boldsymbol{G}_X = [g^j_{x_i} | g^j_{x_i} = w_i f^j_{x_i} \in \mathbb{R},\ i = 1, 2, \cdots, N \text{ and } j = 1, 2, \cdots, J]$. Note that $\mathbb{P}$ is the set of all possible marginal distributions.*

Similarly, we proved that given a feature matrix, representations obtained by 'mean' and 'attention' pooling filters (given attention weights are accessible) can be fully recovered from representation obtained by 'distribution' pooling filter in Proposition 2 and 3, respectively. Moreover, we know that the converse of these propositions are not true since it is possible to obtain the same point estimates from two different distributions. For example, two different feature distributions obtained by 'distribution' pooling filter may have the same mean and 'mean' pooling filter cannot discriminate them. Hence, we conclude that 'distribution' pooling filters are superior to point estimate based pooling filters in terms of amount of information contained inside bag level representations. Furthermore, from Propositions A.1 and A.2 in Supp. A, we inferred that 'distribution with attention' pooling is superior to 'distribution' pooling and 'attention' pooling is superior to 'mean' pooling, respectively. Note that bag level representations containing more information helps to improve the performance on the task at hand.

2.3 MIL TASKS

Positive vs negative bag classification: This is the classical MIL task most commonly referred in the literature (Dietterich et al., 1997; Maron & Lozano-Pérez, 1998). In this task, each bag $X = \{x_i | x_i \in \mathcal{I}, i = 1, 2, \cdots, N\}$ has a bag label $Y = \max_{i=1}^{N}\{y_i\} \in \{0 : negative, 1 : positive\}$. A bag is called '0:negative' if and only if all instances inside the bag are '0:negative', otherwise the bag is called '1:positive'. Note that each instance x_i has a label $y_i \in \{0 : negative, 1 : positive\}$.

Unique class count classification: This is one of the recently introduced tasks in MIL (Oner et al., 2020). In this task, each bag $X = \{x_i | x_i \in \mathcal{I}, i = 1, 2, \cdots, N\}$ has a unique class count (ucc) label $Y = |\{y_i | y_i \in \{1, 2, \cdots, L\}, i = 1, 2, \cdots, N\}| \in \{1 : ucc1, 2 : ucc2, \cdots, L : uccL\}$. *Unique class count* is defined as the number of unique classes that all instances in the bag X belong to.

MIL multi-class classification: Each bag $X = \{x_i | x_i \in \mathcal{I}, i = 1, 2, \cdots, N\}$ has a bag label $Y \in \{0, 1\}^K$. Label vector Y is a one-hot vector consisting of K classes.

MIL multi-task classification: Each bag $X = \{x_i | x_i \in \mathcal{I}, i = 1, 2, \cdots, N\}$ has a bag label $Y \in \{0, 1\}^K$. Label vector Y is a binary vector consisting of K classes, which may not be mutually exclusive, $Y = [Y^k | Y^k \in \{0, 1\}, k = 1, 2, \cdots, K]$.

Regression: Each bag $X = \{x_i | x_i \in \mathcal{I}, i = 1, 2, \cdots, N\}$ has a bag label $Y \in \mathbb{R}$.

3 RELATED WORK

MIL was first introduced in (Dietterich et al., 1997; Maron & Lozano-Pérez, 1998) as positive vs negative bag classification task for drug activity prediction. After that different versions of MIL tasks emerged: unique class count prediction (Oner et al., 2020), multi-class classification (Feng & Zhou, 2017), multi-task classification (Yang et al., 2016) or regression (Zhang et al., 2018). In order to solve these tasks, different MIL methods were derived with different assumptions (Gärtner et al., 2002; Zhang & Goldman, 2002; Chen et al., 2006; Foulds, 2008; Zhang & Zhou, 2009; Zhou et al., 2009; Ramon & De Raedt, 2000; Zhou & Zhang, 2002; Zhang & Zhou, 2004), which are reviewed in detail in (Foulds & Frank, 2010). However, recently there has been a massive shift towards to use neural networks in MIL setup to exploit the power of deep learning (Wu et al., 2015; Wang et al., 2018).

MIL methods were used for many different applications such as, image annotation / categorization / retrieval (Chen & Wang, 2004; Zhang et al., 2002; Tang et al., 2010), text categorization (Andrews et al., 2003; Settles et al., 2008), spam detection (Jorgensen et al., 2008), medical image processing (Dundar et al., 2007; Quellec et al., 2017), face/object detection (Zhang et al., 2006; Felzenszwalb et al., 2010), object tracking (Babenko et al., 2011), defenses against adversarial attacks (Kou et al., 2020). Another related area is set classification, which is the same thing with bag classification in MIL since a bag is a set of instances. Recently, there are a few deep learning based studies in this area (Zaheer et al., 2017; Rezatofighi et al., 2017; Kuncheva et al., 2017). In 'Deep Sets' (Zaheer et al., 2017), for example, the general form of *permutation invariant* set functions on a set is defined.

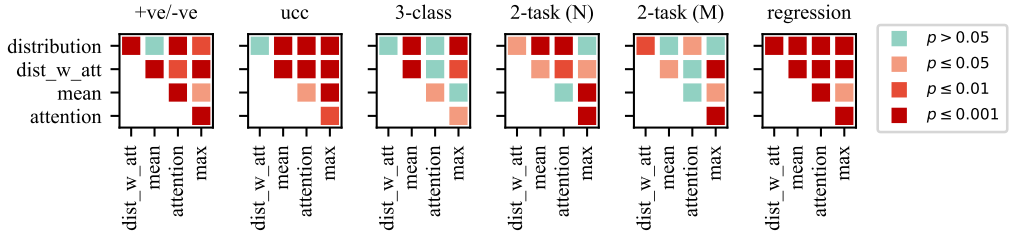
Lastly, different types of MIL pooling filters are used to combine extracted features of instances inside the bags, such as 'max' pooling (Wang et al., 2018; Wu et al., 2015; Feng & Zhou, 2017), 'mean' pooling (Wang et al., 2018; 2019) or 'log-sum-exp' pooling (Ramon & De Raedt, 2000). Although these filters have been widely used in the literature, recently there are new kinds of MIL

Table 1: MIL tasks are summarized by their bag labels for each kind of image and loss functions used during training. For example, in +ve/-ve bag classification task, bag labels are 2-bit one-hot vectors such that '10': -ve bag (fully normal image) and '01': +ve bag (fully metastases or boundary image).

	bag label (Y)				
	+ve/-ve (*one-hot*)	ucc (*one-hot*)	3-class (*one-hot*)	2-task (*binary*)	% metastases (*real-valued*)
Fully normal	10	10	100	1,0	0.0
Fully metastases	01	10	010	0,1	1.0
Boundary	01	01	001	1,1	$0.0 < Y < 1.0$
Loss	CCE	CCE	CCE	BCE	L1
CCE: Categorical Cross Entropy (PyTorch, b), BCE: Binary Cross Entropy (PyTorch, a)					

Table 2: *Top:* Test set performances of MIL models in 5 different MIL tasks formulated on the lymph node metastases dataset. *Bottom:* Pairwise statistical test results are presented as color coded maps obtained by thresholding p-values at different significance levels. Best models are highlighted in bold. *dist_w_att: 'distribution with attention' pooling*

	+ve/-ve	ucc	3-class	2-task (*accuracy*)		% metastases
	(*accuracy*)	(*accuracy*)	(*accuracy*)	normal (N)	metastases (M)	(*absolute error*)
distribution	0.8193	**0.8628**	**0.7473**	0.8533	0.8383	0.1906
dist_w_att	**0.9117**	**0.8709**	**0.7405**	0.8696	**0.8723**	**0.1671**
mean	0.8139	0.6413	0.6780	**0.8913**	0.8438	0.2426
attention	0.8804	0.6957	**0.7188**	**0.8927**	**0.8614**	0.3264
max	0.7636	0.7582	0.6712	0.8356	0.8111	0.2223



pooling filters as well, such as 'distribution' pooling (Oner et al., 2020), 'attention' pooling (Ilse et al., 2018; Lee et al., 2019; Wang et al., 2019) or 'sort' pooling (Lu et al., 2015; Zhang et al., 2020). We have also extended 'distribution' pooling to 'distribution with attention' pooling by incorporating an attention mechanism into it in this paper.

4 Experiments

4.1 Analysis of different MIL pooling filters in different MIL tasks

The objective of this experiment is to investigate the effect of MIL pooling filters on the performance of an MIL model in a real world MIL task. We designed a neural network based MIL framework with the same structure in Sec. 2. We used ResNet18 (He et al., 2016) architecture without batch normalization as feature extractor module, θ_{feature}, and a three layer multi-layer-perceptron as bag level representation transformation module, $\theta_{\text{transform}}$. We tested this framework with 5 different MIL pooling filters in θ_{filter} module on 5 different MIL tasks formulated on the lymph node metastases dataset. Hence, we had 25 different models sharing the same architecture in θ_{feature} module. Note that the code was made publicly available at: `https://bit.ly/mil_pooling_filters`

The lymph node metastases dataset is adapted from Oner et al. (2020) and has training, validation and test sets (*Supp. B.1* for details). The dataset consists of images cropped from histopathology slides of lymph node sections (Bejnordi et al., 2017) and has corresponding ground truth metastases

segmentation masks. There are three types of images in this dataset: *fully normal* - all cells are normal, *fully metastases* - all cells are metastases and *boundary* - mixture of normal and metastases cells. Similar to Sec. 2.3, we formulated five different MIL tasks on this dataset: positive vs negative bag classification *(+ve/-ve: predict whether an image contains metastases cells or not)*; unique class count prediction *(ucc: predict how many types of cells exist in an image)*, multi-class classification *(3-class: predict whether an image is fully normal, fully metastases or boundary)*, multi-task classification *(2-task:* 1^{st} *task - predict whether an image contains normal cells or not,* 2^{nd} *task - predict whether an image contains metastases cells or not)* and regression *(% metastases: predict percentage of metastases pixels inside an image)*. The tasks are summarized by their bag labels for each kind of image and loss functions used during training in Table 1.

On each task, we trained five different models with MIL pooling filters defined in Sec. 2.1, namely 'max', 'mean', 'attention', 'distribution' and 'distribution with attention' pooling. Each model was randomly initialized and trained end-to-end with early-stopping criteria on validation set performance. Once we obtained the best models, we checked performances of the models on hold-out test set (*Supp. B.2* for details). The results are summarized in Table 2 together with used performance metrics for each task (*Supp. B.3* for details). Note that we presented the performances of two tasks separately for multi-task setup. Furthermore, we have conducted statistical tests for comparing the performance of different models in each task. For classification tasks, we used McNemar's test (Everitt, 1977) since all the models were trained on the same training set and tested on the same hold-out test set as suggested in Dietterich (1998). On the other hand, we used paired t-test (Hsu & Lachenbruch, 2005) on the absolute error values obtained for each sample in the test set to compare models in regression task. Results of the pairwise statistical tests for each task are also presented at the bottom of Table 2 with color coded maps obtained by thresholding p-values at different significance levels.

We observed that while the performance of our framework in a particular MIL task is different for different MIL pooling filters, the performance of our framework with a specific MIL pooling filter also changes from task to task. Furthermore, there is no one single MIL pooling filter performing best for all of the MIL tasks; however, 'distribution with attention' pooling filter, consistent with our theoretical analyses, is among the best performing pooling filters in all of the tasks, except 'normal' task in *multi-task* setup, in which 'mean' and 'attention' pooling perform better. Hence, 'distribution with attention' pooling filter seems to be a good candidate for all of the tasks.

4.2 Performance comparison on classical MIL datasets

The performance of our neural network model with 'distribution' pooling filter (Distribution-Net) is compared with the performance of the best MIL methods in positive vs negative bag classification task on 5 classical MIL datasets: drug activity prediction datasets MUSK1 and MUSK2 (Dietterich et al., 1997) and animal image annotation datasets FOX, TIGER and ELEPHANT (Andrews et al., 2003). (See *Supp. C* for details of models and datasets.)

We used 10-fold cross validation and repeated each experiment 5 times. For each dataset, we have declared the mean of classification accuracies ($\pm$ standard error). We compared the performance of Distribution-Net with the performance of state-of-the-art MIL methods on classical MIL datasets in Table 3. While first part of the table contains methods utilizing traditional machine learning techniques (Andrews et al., 2003; Gärtner et al., 2002; Zhang & Goldman, 2002; Zhou et al., 2009; Wei et al., 2016), second part of the table accommodates methods employing neural networks (Wang et al., 2018; Ilse et al., 2018). The last part of the table shows the performance of our Distribution-Net, which outperformed all other methods on all datasets. Neural network based models generally outperformed the traditional machine learning based models. Furthermore, our Distribution-Net performed even better than other neural network based models (Wang et al., 2018; Ilse et al., 2018). Prominent difference between the models are the pooling filters. While other models employed point estimate based pooling filters (Wang et al., 2018; Ilse et al., 2018), Distribution-Net had 'distribution' pooling filter. It seems that Distribution-Net utilized the full distribution information captured by 'distribution' pooling filter over other models.

Table 3: Performances of different MIL methods on classical MIL datasets. First part: methods utilizing traditional machine learning techniques. Second part: methods employing neural networks. Last part: our own model with 'distribution' pooling filter (Distribution-Net). [1] (Andrews et al., 2003), [2] (Gärtner et al., 2002), [3] (Zhang & Goldman, 2002), [4] (Zhou et al., 2009), [5] (Wei et al., 2016), [6] (Wang et al., 2018), [7] (Ilse et al., 2018).

METHOD	MUSK1	MUSK2	FOX	TIGER	ELEPHANT
mi-SVM [1]	$0.874 \pm$ N/A	$0.836 \pm$ N/A	$0.582 \pm$ N/A	$0.784 \pm$ N/A	$0.822 \pm$ N/A
MI-SVM [1]	$0.779 \pm$ N/A	$0.843 \pm$ N/A	$0.578 \pm$ N/A	$0.840 \pm$ N/A	$0.843 \pm$ N/A
MI-Kernel [2]	0.880 ± 0.031	0.893 ± 0.015	0.603 ± 0.028	0.842 ± 0.010	0.843 ± 0.016
EM-DD [3]	0.849 ± 0.044	0.869 ± 0.048	0.609 ± 0.045	0.730 ± 0.043	0.771 ± 0.043
mi-Graph [4]	0.889 ± 0.033	0.903 ± 0.039	0.620 ± 0.044	0.860 ± 0.037	0.869 ± 0.035
miVLAD [5]	0.871 ± 0.043	0.872 ± 0.042	0.620 ± 0.044	0.811 ± 0.039	0.850 ± 0.036
miFV [5]	0.909 ± 0.040	0.884 ± 0.042	0.621 ± 0.049	0.813 ± 0.037	0.852 ± 0.036
mi-Net [6]	0.889 ± 0.039	0.858 ± 0.049	0.613 ± 0.035	0.824 ± 0.034	0.858 ± 0.037
MI-Net [6]	0.887 ± 0.041	0.859 ± 0.046	0.622 ± 0.038	0.830 ± 0.032	0.862 ± 0.034
MI-Net with DS [6]	0.894 ± 0.042	0.874 ± 0.043	0.630 ± 0.037	0.845 ± 0.039	0.872 ± 0.032
MI-Net with RC [6]	0.898 ± 0.043	0.873 ± 0.044	0.619 ± 0.047	0.836 ± 0.037	0.857 ± 0.040
Attention [7]	0.892 ± 0.040	0.858 ± 0.048	0.615 ± 0.043	0.839 ± 0.022	0.868 ± 0.022
Gated-Attention [7]	0.900 ± 0.050	0.863 ± 0.042	0.603 ± 0.029	0.845 ± 0.018	0.857 ± 0.027
Distribution-Net (ours)	$\mathbf{0.923} \pm 0.071$	$\mathbf{0.932} \pm 0.067$	$\mathbf{0.680} \pm 0.075$	$\mathbf{0.864} \pm 0.054$	$\mathbf{0.900} \pm 0.077$

5 CONCLUSION

We theoretically analyzed different MIL pooling filters and experimentally investigated their effects on the performance of an MIL model in some real world MIL tasks. We showed that distribution based pooling filters, in principle, are superior to point estimate based counterparts in terms of amount of information captured while obtaining bag level representations from extracted features. We designed a neural network based MIL framework and analyzed the performance of our framework with 5 different MIL pooling filters in 5 different MIL tasks formulated on the lymph node metastases dataset. We observed that while the performance of our framework in a particular task is different for different pooling filters, the performance of our framework with a specific pooling filter also changes from task to task. Furthermore, we noticed that although it is not the only one, 'distribution with attention' pooling filter is among the best performing pooling filters for almost all tasks. Hence, it seems to be a good candidate for all MIL tasks. Actually, this is in accordance with our theoretical findings in Sec. 2.2 that distribution based pooling filters capture richer information than point estimate based counterparts. Our model with 'distribution' pooling also outperformed state-of-the art methods with different MIL pooling filters on classical MIL datasets. On top of observed superiority of distribution based pooling filters, in future, we want to explore them with fully trainable hyper-parameters, such as number of bins and standard deviation.

REFERENCES

Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *Advances in neural information processing systems*, pp. 577–584, 2003.

Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Robust object tracking with online multiple instance learning. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1619–1632, 2011.

Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes Van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen AWM Van Der Laak, Meyke Hermsen, Quirine F Manson, Maschenka Balkenhol, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*, 318(22):2199–2210, 2017.

Martin Buhmann. Radial basis function. *Scholarpedia*, 5(5):9837, 2010.

Yixin Chen and James Z Wang. Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*, 5(Aug):913–939, 2004.

Yixin Chen, Jinbo Bi, and James Ze Wang. Miles: Multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1931–1947, 2006.

Thomas G Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923, 1998.

Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71, 1997.

Murat Dundar, Balaji Krishnapuram, RB Rao, and Glenn M Fung. Multiple instance learning for computer aided diagnosis. In *Advances in neural information processing systems*, pp. 425–432, 2007.

BS Everitt. The analysis of contingency tables. 1977.

Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

Ji Feng and Zhi-Hua Zhou. Deep miml network. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

James Foulds and Eibe Frank. A review of multi-instance learning assumptions. *The Knowledge Engineering Review*, 25(1):1–25, 2010.

James Richard Foulds. *Learning instance weights in multi-instance learning*. PhD thesis, The University of Waikato, 2008.

Thomas Gärtner, Peter A Flach, Adam Kowalczyk, and Alexander J Smola. Multi-instance kernels. In *ICML*, volume 2, pp. 7, 2002.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Henry Hsu and Peter A Lachenbruch. Paired t test. *Encyclopedia of Biostatistics*, 6, 2005.

Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International Conference on Machine Learning*, pp. 2127–2136, 2018.

Zach Jorgensen, Yan Zhou, and Meador Inge. A multiple instance learning strategy for combating good word attacks on spam filters. *Journal of Machine Learning Research*, 9(Jun):1115–1146, 2008.

Connie Kou, Hwee Kuan Lee, Ee-Chien Chang, and Teck Khim Ng. Enhancing transformation-based defenses against adversarial attacks with a distribution classifier. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BkgWahEFvr.

Ludmila I Kuncheva, Juan J Rodríguez, and Aaron S Jackson. Restricted set classification: Who is there? *Pattern Recognition*, 63:158–170, 2017.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pp. 3744–3753, 2019.

Xin Lu, Zhe Lin, Xiaohui Shen, Radomir Mech, and James Z Wang. Deep multi-patch aggregation network for image style, aesthetics, and quality estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 990–998, 2015.

Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. In *Advances in neural information processing systems*, pp. 570–576, 1998.

Charles A Micchelli. Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constructive approximation*, 2(1):11–22, 1986.

Mustafa Umit Oner, Hwee Kuan Lee, and Wing-Kin Sung. Weakly supervised clustering by exploiting unique class count. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=B1xIj3VYvr.

Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

PyTorch. PyTorch binary cross entropy loss function. https://pytorch.org/docs/stable/nn.html#bcewithlogitsloss, a. Accessed: 2020-05-28.

PyTorch. PyTorch cross entropy loss function. https://pytorch.org/docs/stable/nn.html#crossentropyloss, b. Accessed: 2020-05-28.

Gwenolé Quellec, Guy Cazuguel, Béatrice Cochener, and Mathieu Lamard. Multiple-instance learning for medical image and video analysis. *IEEE reviews in biomedical engineering*, 10: 213–234, 2017.

Jan Ramon and Luc De Raedt. Multi instance neural networks. 2000.

S Hamid Rezatofighi, Anton Milan, Ehsan Abbasnejad, Anthony Dick, Ian Reid, et al. Deepsetnet: Predicting sets with deep neural networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5257–5266. IEEE, 2017.

Robert Schaback. A practical guide to radial basis functions. *Electronic Resource*, 11:1–12, 2007.

Isaac J Schoenberg. Metric spaces and positive definite functions. *Transactions of the American Mathematical Society*, 44(3):522–536, 1938.

Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. In *Advances in neural information processing systems*, pp. 1289–1296, 2008.

Jinhui Tang, Haojie Li, Guo-Jun Qi, and Tat-Seng Chua. Image annotation by graph-based inference with integrated multiple/single instance representations. *IEEE Transactions on Multimedia*, 12(2): 131–141, 2010.

Xinggang Wang, Yongluan Yan, Peng Tang, Xiang Bai, and Wenyu Liu. Revisiting multiple instance neural networks. *Pattern Recognition*, 74:15–24, 2018.

Yun Wang, Juncheng Li, and Florian Metze. A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 31–35. IEEE, 2019.

Xiu-Shen Wei, Jianxin Wu, and Zhi-Hua Zhou. Scalable algorithms for multi-instance learning. *IEEE transactions on neural networks and learning systems*, 28(4):975–987, 2016.

Jiajun Wu, Yinan Yu, Chang Huang, and Kai Yu. Deep multiple instance learning for image classification and auto-annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3460–3469, 2015.

Yanhua Yang, Cheng Deng, Shangqian Gao, Wei Liu, Dapeng Tao, and Xinbo Gao. Discriminative multi-instance multitask learning for 3d action recognition. *IEEE Transactions on Multimedia*, 19 (3):519–529, 2016.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in neural information processing systems*, pp. 3391–3401, 2017.

Cha Zhang, John C Platt, and Paul A Viola. Multiple instance boosting for object detection. In *Advances in neural information processing systems*, pp. 1417–1424, 2006.

Min-Ling Zhang and Zhi-Hua Zhou. Improve multi-instance neural networks through feature selection. *Neural Processing Letters*, 19(1):1–10, 2004.

Min-Ling Zhang and Zhi-Hua Zhou. Multi-instance clustering with applications to multi-instance prediction. *Applied Intelligence*, 31(1):47–68, 2009.

Qi Zhang and Sally A Goldman. Em-dd: An improved multiple-instance learning technique. In *Advances in neural information processing systems*, pp. 1073–1080, 2002.

Qi Zhang, Sally A Goldman, Wei Yu, and Jason E Fritts. Content-based image retrieval using multiple-instance learning. In *ICML*, volume 1, pp. 2. Citeseer, 2002.

Yan Zhang, Jonathon Hare, and Adam Prügel-Bennett. Fspool: Learning set representations with featurewise sort pooling. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HJgBA2VYwH`.

Yong Zhang, Rui Zhao, Weiming Dong, Bao-Gang Hu, and Qiang Ji. Bilateral ordinal relevance multi-instance regression for facial action unit intensity estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7034–7043, 2018.

Zhi-Hua Zhou and Min-Ling Zhang. Neural networks for multi-instance learning. In *Proceedings of the International Conference on Intelligent Information Technology, Beijing, China*, pp. 455–459, 2002.

Zhi-Hua Zhou, Yu-Yin Sun, and Yu-Feng Li. Multi-instance learning by treating instances as non-iid samples. In *Proceedings of the 26th annual international conference on machine learning*, pp. 1249–1256. ACM, 2009.

Wentao Zhu, Qi Lou, Yeeleng Scott Vang, and Xiaohui Xie. Deep multi-instance networks with sparse label assignment for whole mammogram classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 603–611. Springer, 2017.

A PROOFS OF THE PROPOSITIONS

Lemma A.1 *Given two sets* $R = \{r_i | r_i \in \mathbb{R},\ i = 1, 2, \cdots, N_R\}$ *and* $S = \{s_j | s_j \in \mathbb{R},\ j = 1, 2, \cdots, N_S\}$. *Let* $q_R(v) = \sum_{i=1}^{N_R} e^{-\frac{1}{2\sigma^2}(v-r_i)^2}$ *and* $q_S(v) = \sum_{j=1}^{N_S} e^{-\frac{1}{2\sigma^2}(v-s_j)^2}$ *where* $0 < \sigma < \infty$. *If all* r_i *and* s_j *are different (i.e.* $r_i \neq s_j \forall_{i,j}$*), then* $\exists v \in \mathbb{R}\ q_R(v) \neq q_S(v)$.

Proof:

Prove by contradiction:

(i) Suppose, for the sake of contradiction, that $q_R(v) = q_S(v)\ \forall v$.

Let $Z = R \cup S = \{r_1, r_2, \cdots, r_{N_R}, s_1, s_2, \cdots, s_{N_S}\} = \{z_1, z_2, \cdots, z_{N_Z}\}$ where $N_Z = N_R + N_S$.

Let $\mathbf{\Phi} = [\Phi_{u_1 u_2}]$ where $\Phi_{u_1 u_2} = e^{-\frac{1}{2\sigma^2}\left(z_{u_1} - z_{u_2}\right)^2}\ \forall_{u_1, u_2}$.

Then,

$$\mathbf{\Phi} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ -1 \\ -1 \\ \vdots \\ -1 \end{bmatrix} = \begin{bmatrix} q_R(z_1) - q_S(z_1) \\ q_R(z_2) - q_S(z_2) \\ \vdots \\ q_R(z_{N_Z}) - q_S(z_{N_Z}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

This means that $\mathbf{\Phi}$ is of rank $< N_Z$. Hence, $\mathbf{\Phi}$ has no inverse.

(ii) However, $\Phi_{u_1 u_2}$ is in the form of a Gaussian kernel, which is a positive definite radial basis function (Schaback, 2007), and all data points constituting $\mathbf{\Phi}$ are different from each other (i.e. $z_{u_1} \neq z_{u_2}\ \forall_{u_1 \neq u_2}$). Therefore, $\mathbf{\Phi}$ is a positive definite matrix and invertible (Schoenberg, 1938; Micchelli, 1986; Buhmann, 2010).

Hence, there is a contradiction between **(i)** and **(ii)**, so $\exists v \in \mathbb{R}\ q_R(v) \neq q_S(v)$.

■

Lemma A.2 *Given two sets* $R = \{r_i | r_i \in \mathbb{R},\ i = 1, 2, \cdots, N_R\}$ *and* $S = \{s_j | s_j \in \mathbb{R},\ j = 1, 2, \cdots, N_S\}$. *Let* $q_R(v) = \sum_{i=1}^{N_R} e^{-\frac{1}{2\sigma^2}(v-r_i)^2}$ *and* $q_S(v) = \sum_{j=1}^{N_S} e^{-\frac{1}{2\sigma^2}(v-s_j)^2}$ *where* $0 < \sigma < \infty$. *If* $R \neq S$, *then* $\exists v \in \mathbb{R}\ q_R(v) \neq q_S(v)$.

Proof:

We can write $R = R_{inter} \cup R_{diff}$ and $S = S_{inter} \cup S_{diff}$ where $R_{inter} = S_{inter} = R \cap S$, $R_{diff} = R - S$ and $S_{diff} = S - R$.

Then,

$$q_R(v) = \underbrace{\sum_{r \in R_{inter}} e^{-\frac{1}{2\sigma^2}(v-r)^2}}_{q_{R_{inter}}(v)} + \underbrace{\sum_{r \in R_{diff}} e^{-\frac{1}{2\sigma^2}(v-r)^2}}_{q_{R_{diff}}(v)} = q_{R_{inter}}(v) + q_{R_{diff}}(v)$$

$$q_S(v) = \underbrace{\sum_{s \in S_{inter}} e^{-\frac{1}{2\sigma^2}(v-s)^2}}_{q_{S_{inter}}(v)} + \underbrace{\sum_{s \in S_{diff}} e^{-\frac{1}{2\sigma^2}(v-s)^2}}_{q_{S_{inter}}(v)} = q_{S_{inter}}(v) + q_{S_{diff}}(v)$$

(i) $R_{inter} = S_{inter}$, so $q_{R_{inter}}(v) = q_{S_{inter}}(v)\ \forall v$.

(ii) Condition of lemma: $R \neq S$, so $R_{diff} \cup S_{diff} \neq \emptyset$ and $R_{diff} \cap S_{diff} = \emptyset$. Therefore, from Lemma A.1 $\exists v \in \mathbb{R}\ q_{R_{diff}}(v) \neq q_{S_{diff}}(v)$.

Hence, from **(i)** and **(ii)** $\exists v \in \mathbb{R}\ q_R(v) \neq q_S(v)$.

■

Proposition 1 *Given two feature matrices obtained from bags* $X_a = \{x_1^{(a)}, x_2^{(a)}, \cdots, x_N^{(a)}\}$ *and* $X_b = \{x_1^{(b)}, x_2^{(b)}, \cdots, x_N^{(b)}\}$*;*

- $\boldsymbol{F}_{X_a} = [f^j_{x_i^{(a)}} | f^j_{x_i^{(a)}} \in \mathbb{R},\ f^j_{x_i^{(a)}} \neq f^j_{x_u^{(a)}}\ \forall_{i \neq u},\ i, u = 1, 2, \cdots, N$ *and* $j = 1, 2, \cdots, J]$
- $\boldsymbol{F}_{X_b} = [f^j_{x_i^{(b)}} | f^j_{x_i^{(b)}} \in \mathbb{R},\ f^j_{x_i^{(b)}} \neq f^j_{x_u^{(b)}}\ \forall_{i \neq u},\ i, u = 1, 2, \cdots, N$ *and* $j = 1, 2, \cdots, J]$

and two pooling filters; 'max' pooling filter θ^{max}_{filter} *and 'distribution' pooling filter* θ^{dist}_{filter}*. Let* ${}_{max}\boldsymbol{h}_{X_a}$ *and* ${}_{max}\boldsymbol{h}_{X_b}$ *be bag level representations obtained by* θ^{max}_{filter} *from* $\boldsymbol{F}_{X_a}$ *and* $\boldsymbol{F}_{X_b}$*, respectively. Similarly, let* ${}_{dist}\boldsymbol{h}_{X_a}$ *and* ${}_{dist}\boldsymbol{h}_{X_b}$ *be bag level representations obtained by* θ^{dist}_{filter} *from* $\boldsymbol{F}_{X_a}$ *and* $\boldsymbol{F}_{X_b}$*, respectively. If* ${}_{max}\boldsymbol{h}_{X_a} \neq {}_{max}\boldsymbol{h}_{X_b}$*, then* ${}_{dist}\boldsymbol{h}_{X_a} \neq {}_{dist}\boldsymbol{h}_{X_b}$.

Proof:

Let $\boldsymbol{F}^j_{X_a}$ and $\boldsymbol{F}^j_{X_b}$ be j^{th} feature sets for bags X_a and X_b such that $\boldsymbol{F}^j_{X_a} = \{f^j_{x_1^{(a)}}, f^j_{x_2^{(a)}}, \cdots, f^j_{x_N^{(a)}}\}$ and $\boldsymbol{F}^j_{X_b} = \{f^j_{x_1^{(b)}}, f^j_{x_2^{(b)}}, \cdots, f^j_{x_N^{(b)}}\}$.

(i) For $\theta^{max}_{\text{filter}}$, bag level representations:

${}_{max}\boldsymbol{h}_{X_a} = [{}_{max}h^j_{X_a} \mid {}_{max}h^j_{X_a} = \max(\boldsymbol{F}^j_{X_a}) \in \mathbb{R}, j = 1, 2, \cdots, J] \in \mathbb{R}^J$

${}_{max}\boldsymbol{h}_{X_b} = [{}_{max}h^j_{X_b} \mid {}_{max}h^j_{X_b} = \max(\boldsymbol{F}^j_{X_b}) \in \mathbb{R}, j = 1, 2, \cdots, J] \in \mathbb{R}^J$

Condition of proposition: ${}_{max}\boldsymbol{h}_{X_a} \neq {}_{max}\boldsymbol{h}_{X_b}$, so $\exists j\ {}_{max}h^j_{X_a} \neq {}_{max}h^j_{X_b}$. Thus, $\exists j\ \boldsymbol{F}^j_{X_a} \neq \boldsymbol{F}^j_{X_b}$.

(ii) For $\theta^{dist}_{\text{filter}}$, bag level representations:

$$ {}_{dist}\boldsymbol{h}_{X_a} = \begin{bmatrix} \tilde{p}^1_{X_a}(v) \\ \vdots \\ \tilde{p}^j_{X_a}(v) \\ \vdots \\ \tilde{p}^J_{X_a}(v) \end{bmatrix} \text{ where } \tilde{p}^j_{X_a}(v) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\left(v - f^j_{x_i^{(a)}}\right)^2} \forall_{j=1,2,\cdots,J} $$

We can also re-write $\tilde{p}^j_{X_a}(v)$ as $\tilde{p}^j_{X_a}(v) = \frac{1}{N} \frac{1}{\sqrt{2\pi\sigma^2}} \sum_{r \in \boldsymbol{F}^j_{X_a}} e^{-\frac{1}{2\sigma^2}(v-r)^2}$

and

$$ {}_{dist}\boldsymbol{h}_{X_b} = \begin{bmatrix} \tilde{p}^1_{X_b}(v) \\ \vdots \\ \tilde{p}^j_{X_b}(v) \\ \vdots \\ \tilde{p}^J_{X_b}(v) \end{bmatrix} \text{ where } \tilde{p}^j_{X_b}(v) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\left(v - f^j_{x_i^{(b)}}\right)^2} \forall_{j=1,2,\cdots,J} $$

We can also re-write $\tilde{p}^j_{X_b}(v)$ as $\tilde{p}^j_{X_b}(v) = \frac{1}{N} \frac{1}{\sqrt{2\pi\sigma^2}} \sum_{s \in \boldsymbol{F}^j_{X_b}} e^{-\frac{1}{2\sigma^2}(v-s)^2}$

From (i) and (ii): We know that $\exists j\ \boldsymbol{F}^j_{X_a} \neq \boldsymbol{F}^j_{X_b}$, so by using Lemma A.2 $\exists j\ \exists v\ \tilde{p}^j_{X_a}(v) \neq \tilde{p}^j_{X_b}(v)$.

Hence, since $\exists j\ \exists v\ \tilde{p}^j_{X_a}(v) \neq \tilde{p}^j_{X_b}(v)$, ${}_{dist}\boldsymbol{h}_{X_a} \neq {}_{dist}\boldsymbol{h}_{X_b}$.

■

Proposition 2 *Given a feature matrix $\boldsymbol{F}_X = [f^j_{x_i} | f^j_{x_i} \in \mathbb{R},\ i = 1, 2, \cdots, N$ and $j = 1, 2, \cdots, J]$ obtained from a bag $X = \{x_1, x_2, \cdots, x_N\}$, 'mean' pooling filter θ^{mean}_{filter} and 'distribution' pooling filter θ^{dist}_{filter}. Let ${}_{mean}\boldsymbol{h}_X = [{}_{mean}h^j_X \mid {}_{mean}h^j_X \in \mathbb{R},\ j = 1, 2, \cdots, J]$ and ${}_{dist}\boldsymbol{h}_X = [\tilde{p}^j_X | \tilde{p}^j_X \in \mathbb{P}, j = 1, 2, \cdots, J]$ be bag level representations obtained from $\boldsymbol{F}_X$ by θ^{mean}_{filter} and θ^{dist}_{filter}, respectively. Then, ${}_{mean}h^j_X = \mathbb{E}[V^j]\ \forall_{j=1,2,\cdots,J}$ where $V^j \sim \tilde{p}^j_X$. Note that $\mathbb{P}$ is the set of all possible marginal distributions.*

Proof:

(i) For $\theta^{mean}_{\text{filter}}$, bag level representation:

${}_{mean}\boldsymbol{h}_X = [{}_{mean}h^j_X \mid {}_{mean}h^j_X \in \mathbb{R}, j = 1, 2, \cdots, J] \in \mathbb{R}^J$

${}_{mean}h^j_X = \frac{1}{N}\sum_{i=1}^N f^j_{x_i}$.

(ii) For $\theta^{dist}_{\text{filter}}$, bag level representation:

$$ {}_{dist}\boldsymbol{h}_X = \begin{bmatrix} \tilde{p}^1_X(v) \\ \vdots \\ \tilde{p}^j_X(v) \\ \vdots \\ \tilde{p}^J_X(v) \end{bmatrix} \text{ where } \tilde{p}^j_X(v) = \frac{1}{N}\sum_{i=1}^N \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\left(v - f^j_{x_i}\right)^2} \forall_{j=1,2,\cdots,J} $$

From (i) and (ii): by using definition of expected value:

$$ {}_{mean}h^j_X = \int v \tilde{p}^j_X(v) dv\ \forall_{j=1,2,\cdots,J} $$

Hence, ${}_{mean}h^j_X = \mathbb{E}[V^j]\ \forall_{j=1,2,\cdots,J}$ where $V^j \sim \tilde{p}^j_X$.

■

Proposition 3 *Given a feature matrix $\boldsymbol{F}_X = [f^j_{x_i} | f^j_{x_i} \in \mathbb{R},\ i = 1, 2, \cdots, N$ and $j = 1, 2, \cdots, J]$ obtained from a bag $X = \{x_1, x_2, \cdots, x_N\}$, 'attention' pooling filter θ^{att}_{filter} with attention weights $w_i > 0\ \forall_i$, $\sum_{i=1}^N w_i = 1$ and 'distribution' pooling filter θ^{dist}_{filter}. Let ${}_{att}\boldsymbol{h}_X = [{}_{att}h^j_X \mid {}_{att}h^j_X \in \mathbb{R},\ j = 1, 2, \cdots, J]$ be bag level representation obtained from $\boldsymbol{F}_X$ by θ^{att}_{filter}. If attention weights are accessible, then ${}_{att}h^j_X = N \times \mathbb{E}[V^j]\ \forall_{j=1,2,\cdots,J}$ where $V^j \sim \tilde{p}^j_X$ and ${}_{dist}\boldsymbol{h}_X = [\tilde{p}^j_X | \tilde{p}^j_X \in \mathbb{P}, j = 1, 2, \cdots, J]$ is the bag level representation obtained by θ^{dist}_{filter} from $\boldsymbol{G}_X = [g^j_{x_i} | g^j_{x_i} = w_i f^j_{x_i} \in \mathbb{R},\ i = 1, 2, \cdots, N$ and $j = 1, 2, \cdots, J]$. Note that $\mathbb{P}$ is the set of all possible marginal distributions.*

Proof:

(i) For $\theta^{att}_{\text{filter}}$, bag level representation:

${}_{att}\boldsymbol{h}_X = [{}_{att}h^j_X \mid {}_{att}h^j_X \in \mathbb{R}, j = 1, 2, \cdots, J] \in \mathbb{R}^J$

${}_{att}h^j_X = \sum_{i=1}^N w_i f^j_{x_i}\ \forall_j$ where attention weight $w_i > 0\ \forall_i$ and $\sum_{i=1}^N w_i = 1$

We can re-write ${}_{att}h^j_X$ as:

$_{att}h_X^j = N \times \frac{1}{N}\sum_{i=1}^{N} g_{x_i}^j \; \forall_j$ where $g_{x_i}^j = w_i f_{x_i}^j$

Now it is nothing but 'mean' pooling of weighted features multiplied with a scalar, so we can use Proposition 2 by writing 'distribution' pooling in terms of weighted features $g_{x_i}^j = w_i f_{x_i}^j$.

(ii) For $\theta_{\text{filter}}^{dist}$, bag level representation:

$$_{dist}\boldsymbol{h}_X = \begin{bmatrix} \tilde{p}_X^1(v) \\ \vdots \\ \tilde{p}_X^j(v) \\ \vdots \\ \tilde{p}_X^J(v) \end{bmatrix} \text{ where } \tilde{p}_X^j(v) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\left(v - g_{x_i}^j\right)^2} \forall_{j=1,2,\cdots,J}$$

From (i) and (ii): by using definition of expected value:

$$_{att}h_X^j = N \int v\tilde{p}_X^j(v)dv \; \forall_{j=1,2,\cdots,J}$$

Hence, $_{att}h_X^j = N \times \mathbb{E}[V^j] \; \forall_{j=1,2,\cdots,J}$ where $V^j \sim \tilde{p}_X^j$.

■

Proposition A.1 *Given a feature matrix $\boldsymbol{F}_X = [f_{x_i}^j | f_{x_i}^j \in \mathbb{R},\ i = 1, 2, \cdots, N$ and $j = 1, 2, \cdots, J]$ obtained from a bag $X = \{x_1, x_2, \cdots, x_N\}$, 'mean' pooling filter θ_{filter}^{mean} and 'attention' pooling filter θ_{filter}^{att}. Let $_{mean}\boldsymbol{h}_X$ and $_{att}\boldsymbol{h}_X$ be bag level representations obtained from $\boldsymbol{F}_X$ by θ_{filter}^{mean} and θ_{filter}^{att}, respectively. Then, for attention weights set to $\frac{1}{N}$ in 'attention' pooling filter, $_{mean}\boldsymbol{h}_X =_{att} \boldsymbol{h}_X$.*

Proof:

(i) For $\theta_{\text{filter}}^{mean}$, bag level representation:

$_{mean}\boldsymbol{h}_X = [_{mean}h_X^j \mid _{mean}h_X^j \in \mathbb{R}, j = 1, 2, \cdots, J] \in \mathbb{R}^J$

$_{mean}h_X^j = \frac{1}{N}\sum_{i=1}^{N} f_{x_i}^j$

(ii) For $\theta_{\text{filter}}^{att}$, bag level representation:

$_{att}\boldsymbol{h}_X = [_{att}h_X^j \mid _{att}h_X^j \in \mathbb{R}, j = 1, 2, \cdots, J] \in \mathbb{R}^J$

$_{att}h_X^j = \sum_{i=1}^{N} w_i f_{x_i}^j$ where attention weight $w_i > 0 \; \forall_i$ and $\sum_{i=1}^{N} w_i = 1$

From (i) and (ii): For $w_i = \frac{1}{N} \; \forall_i$, $_{att}h_X^j =_{mean} h_X^j \; \forall_j$, so $_{att}\boldsymbol{h}_X =_{mean} \boldsymbol{h}_X$.

■

Proposition A.2 *Given a feature matrix $\boldsymbol{F}_X = [f_{x_i}^j | f_{x_i}^j \in \mathbb{R},\ i = 1, 2, \cdots, N$ and $j = 1, 2, \cdots, J]$ obtained from a bag $X = \{x_1, x_2, \cdots, x_N\}$, 'distribution' pooling filter θ_{filter}^{dist} and 'distribution with attention' pooling filter $\theta_{filter}^{dist\_w\_att}$. Let $_{dist}\boldsymbol{h}_X$ and $_{dist\_w\_att}\boldsymbol{h}_X$ be bag level representations obtained from $\boldsymbol{F}_X$ by θ_{filter}^{dist} and $\theta_{filter}^{dist\_w\_att}$, respectively. Then, for attention weights set to $\frac{1}{N}$ in 'distribution with attention' pooling filter, $_{dist}\boldsymbol{h}_X = _{dist\_w\_att}\boldsymbol{h}_X$.*

Proof:

(i) For $\theta_{\text{filter}}^{dist}$, bag level representation:

$$_{dist}\boldsymbol{h}_X = \begin{bmatrix} _{dist}\tilde{p}_X^1(v) \\ \vdots \\ _{dist}\tilde{p}_X^j(v) \\ \vdots \\ _{dist}\tilde{p}_X^J(v) \end{bmatrix} \text{ where } _{dist}\tilde{p}_X^j(v) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\left(v - f_{x_i}^j\right)^2} \forall_{j=1,2,\cdots,J}$$

(ii) For $\theta_{\text{filter}}^{dist\_w\_att}$, bag level representation:

$$_{dist\_w\_att}\boldsymbol{h}_X = \begin{bmatrix} _{dist\_w\_att}\tilde{p}_X^1(v) \\ \vdots \\ _{dist\_w\_att}\tilde{p}_X^j(v) \\ \vdots \\ _{dist\_w\_att}\tilde{p}_X^J(v) \end{bmatrix} \text{ where } _{dist\_w\_att}\tilde{p}_X^j(v) = \sum_{i=1}^{N} w_i \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\left(v - f_{x_i}^j\right)^2} \forall_{j=1,2,\cdots,J}$$

Attention weight $w_i > 0 \; \forall_i$ and $\sum_{i=1}^{N} w_i = 1$

From (i) and (ii): For $w_i = \frac{1}{N} \; \forall_i$, $_{dist\_w\_att}\tilde{p}_X^j(v) =_{dist} \tilde{p}_X^j(v) \; \forall_j$, so $_{dist}\boldsymbol{h}_X =_{dist\_w\_att} \boldsymbol{h}_X$.

■

B EXPERIMENTS ON LYMPH NODE METASTASES DATASET

We investigated the effect of MIL pooling filter on the performance of an MIL model in a particular real world MIL task. We designed a neural network based MIL framework and analyzed the performance of our framework with 5 different MIL pooling filters in 5 different MIL tasks formulated on a real world lymph node metastases dataset.

Code for the experiments was made publicly available at: `https://bit.ly/mil_pooling_filters`

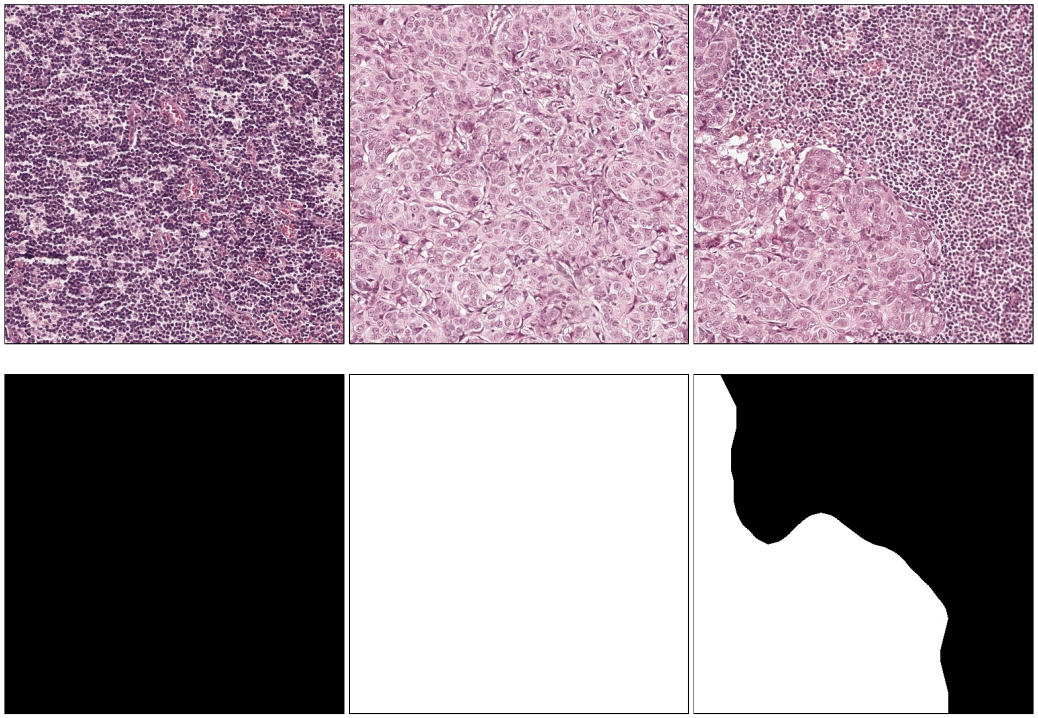
B.1 LYMPH NODE METASTASES DATASET

The lymph node metastases dataset is adapted from Oner et al. (2020). The original dataset consists of 512×512 images cropped from histopathology slides of lymph node sections (Bejnordi et al., 2017) and has corresponding ground truth metastases segmentation masks. There are three types of images in this dataset: *fully normal* - all cells are normal, *fully metastases* - all cells are metastases and *boundary* - mixture of normal and metastases cells. Example images of each type are shown in Figure 3. In order to make a clear distinction between these three types of images, we filtered out the images with: (i) $0 <$ percent metastases ≤ 20 and (ii) $80 \leq$ percent metastases < 100. Moreover, in order to obtain a balanced dataset, we dropped some of the images in the test set coming from one specific slide.

Our dataset is publicly available with this paper. For our dataset, number of images and percent metastases histograms in training, validation and test sets are given in Table 4 and Figure 4, respectively.

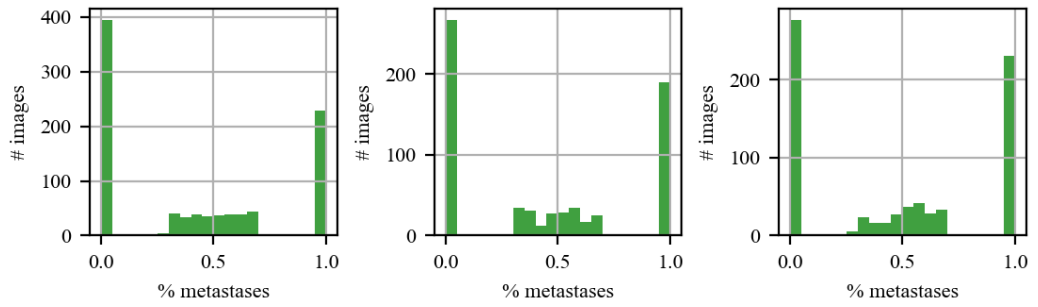
Table 4: Lymph node metastases dataset - number of images in training, validation and test sets.

	Fully normal	Fully metastases	Boundary	Total
Training	395	228	310	933
Validation	267	190	211	668
Test	277	231	228	736



(a) Fully normal (b) Fully metastases (c) Boundary

Figure 3: Examples of three types of images in the dataset are shown together with their corresponfing ground truth metastases masks: (a) *fully normal* - all cells are normal, (b) *fully metastases* - all cells are metastases and (c) *boundary* - mixture of normal and metastases cells.



(a) Training (b) Validation (c) Test

Figure 4: Percent metastases histograms for training, validation and test sets.

B.2 Neural network architectures and hyper-parameters

We designed a neural network based MIL framework. We used ResNet18 (He et al., 2016) architecture without batch normalization as feature extractor module, θ_{feature}, and a three layer multi-layer-perceptron as bag level representation transformation module, $\theta_{\text{transform}}$. We tested this framework with 5 different MIL pooling filters in θ_{filter} module on 5 different MIL tasks formulated. Hence, we had 25 different models sharing the same architecture in θ_{feature} module. Moreover, all models have the same hidden layers in $\theta_{\text{transform}}$ module; however, note that number of input nodes in $\theta_{\text{transform}}$ module depends on θ_{filter} and number of output nodes in $\theta_{\text{transform}}$ module depends on MIL task. Please refer to the provided code for more details.

During training of the models, each image was treated as a bag. We prepared bags on-the-go during training by randomly cropping 32×32 patches over the images. Each bag was created with 64

cropped patches (instances). Data augmentation was also applied on the cropped patches. We used batch size of 32 and extracted 32 features for each instance inside a bag. In the models with 'distribution' pooling filter, kernel density estimation with a Gaussian kernel was used to estimate marginal distributions of extracted features. We used standard deviation of $\sigma = 0.0167$ in Gaussian kernel and 21 bins to sample the estimated distributions. Furthermore, for attention mechanism, we used the same architecture in (Ilse et al., 2018) in the models with 'attention' pooling and 'distribution with attention' pooling filters. We trained models by using ADAM optimizer with a learning rate of $lr = 1e-4$ and $L2$ regularization on the weights with a weight decay of $weight\_decay = 0.0005$. Each model was randomly initialized and trained end-to-end with early-stopping criteria on validation set performance. The architecture and list of hyper-parameters used in MIL models are given in Table 5.

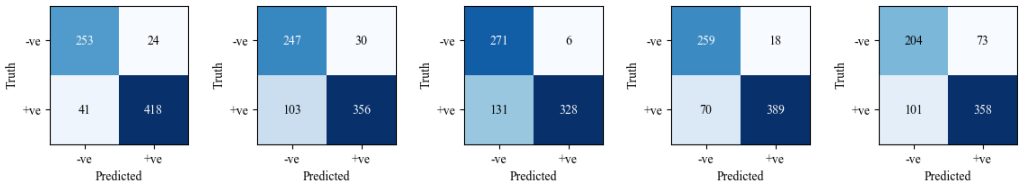
During testing, we created 100 bags from each image and tested with the trained model. Final prediction for each image in the test set was obtained by averaging 100 predictions.

Table 5: Experiments on lymph node metastases dataset - architecture and list of hyper-parameters used in the MIL models. *'dist_w_att': 'distribution with attention' pooling*

Architecture	input-$32x32x3$ ResNet18 w/o BN 'distribution' / 'dist_w_att' / 'mean' / 'attention' / 'max' pooling Dropout(0.5) fc-128 + ReLU Dropout(0.5) fc-32 + ReLU Dropout(0.5) fc-2 *(+ve/-ve, ucc, 2-task)* / fc-3 *(3-class)* / fc-1 *(regression)* softmax *(+ve/-ve, ucc, 3-class)* / sigmoid *(2-task)* / None *(regression)*
image size	512×512
patch size	32×32
# instances per bag	64
# features	32
# bins in 'distribution' filters	21
σ in Gaussian kernel	0.0167
Optimizer	ADAM
Learning rate	$1e-4$
$L2$ regularization weight decay	0.0005
batch size	32

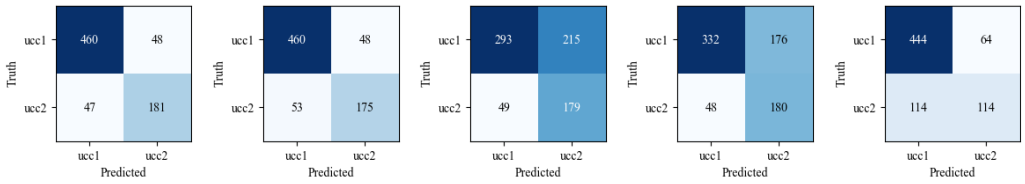
B.3 CONFUSION MATRICES FOR CLASSIFICATION TASKS

Confusion matrices for +ve/-ve bag classification, ucc classification, 3-class classification, metastases task of 2-task classification and normal task of 2-task classification are given in Figure 5, 6, 7, 8 and 9.



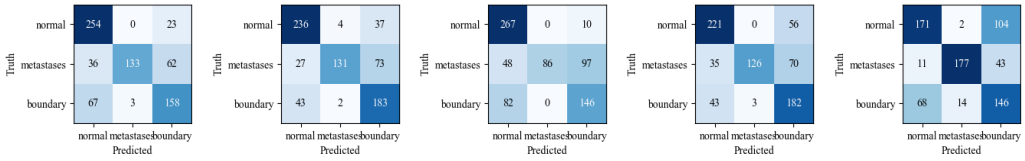
(a) distribution (b) dist_w_att (c) mean (d) attention (e) max

Figure 5: Confusion matrices for +ve/-ve bag classification.



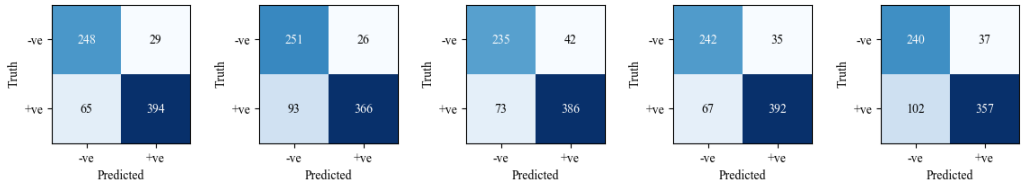
(a) distribution (b) dist_w_att (c) mean (d) attention (e) max

Figure 6: Confusion matrices for ucc classification.



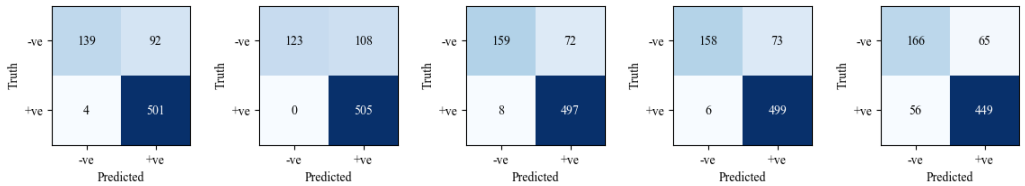
(a) distribution (b) dist_w_att (c) mean (d) attention (e) max

Figure 7: Confusion matrices for 3-class classification.



(a) distribution (b) dist_w_att (c) mean (d) attention (e) max

Figure 8: Confusion matrices for metastases task in 2-task classification.



(a) distribution (b) dist_w_att (c) mean (d) attention (e) max

Figure 9: Confusion matrices for metastases task in 2-task classification.

C EXPERIMENTS ON CLASSICAL MIL DATASETS

This section compares the performance of our neural network model with 'distribution' pooling filter (Distribution-Net) with the performance of the best MIL methods on classical MIL task of positive vs negative bag classification on 5 classical MIL datasets: drug activity prediction datasets MUSK1 and MUSK2 (Dietterich et al., 1997) and animal image annotation datasets FOX, TIGER and ELEPHANT (Andrews et al., 2003). Summary of classical MIL datasets is given in Table 6.

Table 6: Summary of classical MIL datasets

	# bags			# instances per bag			
	positive	negative	total	min	max	average	# features
MUSK1	47	45	92	2	40	5.17	166
MUSK2	39	63	102	1	1044	64.69	166
FOX	100	100	200	2	13	6.6	230
TIGER	100	100	200	1	13	6.1	230
ELEPHANT	100	100	200	2	13	6.96	230

The summary of architectures and hyper-parameters used in MIL models on 'MUSK' and 'Animal' datasets are given in Table 7 and Table 8, respectively. Note that we used mini-batch training with bags including equal number of instances. We created bags by sampling from available instances of each sample (a drug with multiple conformations for 'MUSK' datasets and an image with multiple segments in 'Animal' datasets). When the number of available instances of a sample is less than number of instances required to create a bag we used available instances more than once in a bag. We have determined number of instances with cross-validation on the validation sets.

Table 7: MUSK datasets - architecture and list of hyper-parameters used in the MIL models.

Architecture	input-166 fc-64 + ReLU Dropout(0.5) fc-32 + ReLU Dropout(0.5) fc-32 + Sigmoid 'distribution' pooling Dropout(0.5) fc-64 + ReLU Dropout(0.5) fc-32 + ReLU Dropout(0.5) fc-2 softmax
# instances per bag	16
# features	32
# bins in 'distribution' pooling filters	11
σ in Gaussian kernel	0.1
Optimizer	ADAM
Learning rate	$5e-4$
$L2$ regularization weight decay	0.1
batch size	8

Table 8: Animal datasets - architecture and list of hyper-parameters used in the MIL models.

Architecture	input-230 fc-256 + ReLU Dropout(0.5) fc-128 + ReLU Dropout(0.5) fc-64 + ReLU Dropout(0.5) fc-32 + Sigmoid 'distribution' pooling Dropout(0.5) fc-384 + ReLU Dropout(0.5) fc-192 + ReLU Dropout(0.5) fc-2 softmax
# instances per bag	16
# features	32
# bins in 'distribution' pooling filters	11
σ in Gaussian kernel	0.1
Optimizer	Adam
Learning rate	$5e-6$
$L2$ regularization weight decay	0.1
batch size	8

The summary of architectures and hyper-parameters used in MIL models of (Wang et al., 2018) and (Ilse et al., 2018) on 'MUSK' and 'Animal' datasets are given in Table 9.

Table 9: MUSK and Animal datasets - architecture and list of hyper-parameters used in the MIL models of (Wang et al., 2018) and (Ilse et al., 2018).

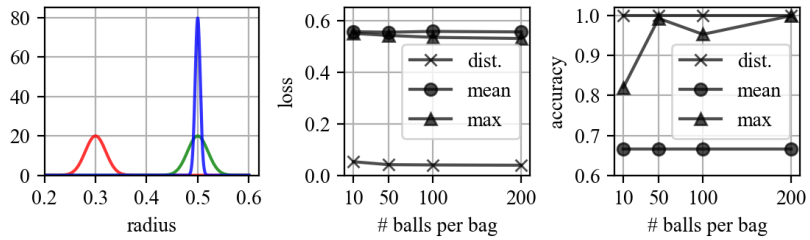
Architecture	input-166 fc-256 + ReLU Dropout fc-128 + ReLU Dropout fc-64 + ReLU Dropout 'max', 'mean', 'attention' pooling fc-1 + Sigmoid
Optimizer	SGD
Learning rate	$5e-4$ (MUSK1, MUSK2, Fox) / $1e-4$ (Tiger, Elephant)
Momentum	0.9
$L2$ regularization weight decay	0.03 (MUSK2) / 0.01 (Tiger) / 0.005 (MUSK1, Fox, Elephant)
batch size	1

D DISTRIBUTION POOLING: EXPLOITING THE FULL INFORMATION IN DISTRIBUTIONS

'Distribution' pooling extracts full information of the distribution. However, 'mean' and 'max' pooling only provide point estimates. This experiment aims to show that 'distribution' pooling is more powerful than point estimators like 'mean' and 'max' pooling. Let's have a hypothetical factory producing metal balls for ball bearings. The factory has three production lines, namely *red*, *green* and *blue*, and radius of the balls produced in these lines are normally distributed: $r_{red} \sim \mathcal{N}(\mu = 0.3, \sigma = 0.02)$, $r_{green} \sim \mathcal{N}(\mu = 0.5, \sigma = 0.02)$ and $r_{blue} \sim \mathcal{N}(\mu = 0.5, \sigma = 0.005)$, as shown in Figure 10a. Then, our MIL task is to classify bags of metal balls from 3 different production lines, i.e. 3-class MIL classification task.

We generated radius data for 900 bags of metal balls with corresponding production line labels. There were 300 bags from each production line and each bag had 200 balls. We designed an MIL framework such that a bag level representation was obtained by using an MIL pooling filter over the radius values of the balls in a bag. Then, the bag level representation was fed to a linear classifier to predict the production line label of the bag. List of hyper-parameters used in MIL models is given in Table 10. We splitted data into training, validation and test sets with 600, 150 and 150 bags, respectively. In each set, there were equal number of bags from each class. We trained MIL models with 'distribution', 'mean' and 'max' pooling filters on the bags with 10, 50, 100 and 200 balls per bag. All models were trained on the training set with categorical cross-entropy loss and fine-tuned on the validation set. Loss and accuracy values of the models on the test set are given in Figure 10b and 10c, respectively. As it is seen in Figure 10c, models with 'distribution' pooling classified all the bags perfectly by exploiting the full information in the radius distribution of balls in the bags and outperformed the models with 'mean' and 'max' pooling. While models with 'mean' pooling distinguished bags of *red* line from others, it couldn't distinguish the bags of *green* line from the bags of *blue* line since their mean radius values were close to each other. For models with 'max' pooling, they can distinguish bags of *red* line from others. For bags from *green* and *blue* lines, the 'max' polling can distinguish them only when the number of balls inside the bags is big enough. Moreover, loss of models with 'distribution' pooling is much lower than the others, so they are much more confident in their predictions.

Figure 11 shows test set confusion matrices. Models with 'distribution' pooling classified all the bags perfectly by exploiting the full information in the radius distribution of balls in the bags and outperformed the models with 'mean' and 'max' pooling. While models with 'mean' pooling distinguished bags of *red* line from others, it couldn't distinguish the bags of *green* line from the bags of *blue* line since their mean radius values were close to each other. For models with 'max'



(a) (b) (c)

Figure 10: 3-class MIL classification task: classifying bags of metal balls from 3 different production lines. (a) Distribution of radius of balls produced in *red*, *green* and *blue* production lines. (b) Loss and (c) accuracy values of MIL models with 'distribution' (*dist.*) pooling, 'mean' pooling and 'max' pooling filters on the test set bags with 10, 50, 100 and 200 balls per bag.

Table 10: Classifying bags of metal balls - architecture and list of hyper-parameters used in the MIL models.

Architecture	input-1 'mean' / 'max' / 'distribution' pooling fc-3 softmax
# balls per bag	10 / 50 / 100 / 200
# features	1
# bins in 'distribution' pooling filters	101
σ in Gaussian kernel	0.005
Optimizer	ADAM
Learning rate	$1e-2$
$L2$ regularization weight decay	0.0005
batch size	64

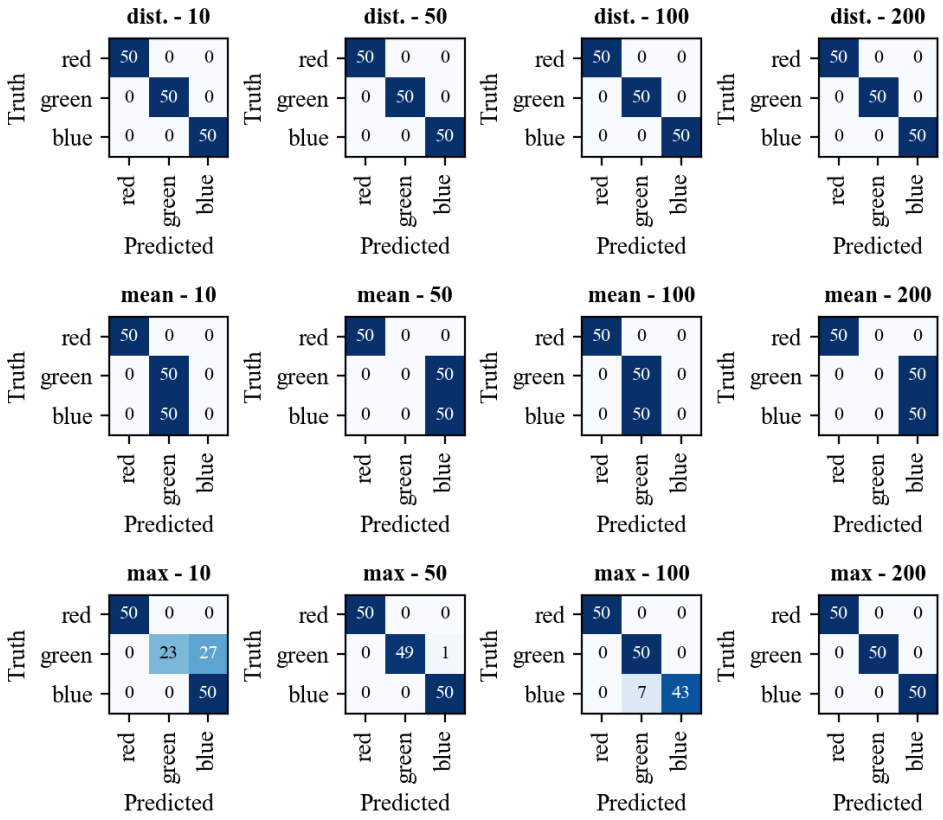


Figure 11: Confusion matrices for models with 'distribution' *(dist.)* pooling, 'mean' pooling and 'max' pooling filters on the test set bags with 10, 50, 100 and 200 balls per bag.

pooling, they can distinguish bags of *red* line from others. For bags from *green* and *blue* lines, the 'max' polling can distinguish them only when the number of balls inside the bags is big enough.