An Iterative Approach for Multiple Instance Learning Problems

Kaili Wang∗ Jose Oramas† Tinne Tuytelaars∗

∗KU Leuven, ESAT-PSI, †University of Antwerp, IDLab

Abstract

Multiple Instance learning (MIL) algorithms are tasked with learning how to associate sets of elements with specific set-level outputs. Towards this goal, the main challenge of MIL lies in modelling the underlying structure that characterizes sets of elements. Existing methods addressing MIL problems are usually tailored to address either: a specific underlying set structure; specific prediction tasks, e.g. classification, regression; or a combination of both. Here we present an approach where a set representation is learned, iteratively, by looking at the constituent elements of each set one at a time. The iterative analysis of set elements enables our approach with the capability to update the set representation so that it reflects whether relevant elements have been detected and whether the underlying structure has been matched. These features provide our method with some model explanation capabilities. Despite its simplicity, the proposed approach not only effectively models different types of underlying set structures, but it is also capable of handling both classification and regression tasks – all this while requiring minimal modifications. An extensive empirical evaluation shows that the proposed method is able to reach and surpass the state-of-the-art.

1 Introduction

Traditional single-instance classification methods focus on learning a mapping between a feature vector (extracted from a single instance) w.r.t. a specific class label of interest. In a complementary fashion, Multiple Instance Learning (MIL) (Sammut and Webb 2011) algorithms are tasked with learning how to associate a set of elements, usually referred to as a "bag", with a specific label. In comparison, MIL methods usually require weaker supervision in the form of set-level labels. The capability of making predictions over groups of elements while requiring weak supervision is a characteristic that makes this family of methods attractive to address several real-world applications. Examples include drug activity prediction, image classification (Carbonneau et al. 2018) [Wei and Zhou 2016], image retrieval, sound classification (Briggs, Fern, and Raich 2012), anomaly detection (Elmoudi 2018), medical imaging (Sudharshan et al. 2019) and web-mining.

Performing predictions at the level of sets of elements introduces several challenges. On the one hand, the label of a set can be defined as a function of specific instance-level characteristics of the elements that compose it. On the other hand, this set label can be defined as a function of relationships occurring between the constituent elements. Since several relationships are possible between the elements, this later scenario constitutes a more challenging problem. As can be found in the MIL literature (Amores 2013 [Carbonneau et al. 2018], [Foulds and Frank 2010]), several algorithms have been proposed during the last decade tailored to address specific tasks/goals, formulated as an MIL problem. Each of these tasks possessing specific set characteristics, usually referred to as the "Multiple Instance (MI) assumptions". The most common of these assumptions, i.e. the standard MI assumption, states that a set is positive if it contains at least one positive element (usually called witness), otherwise the set is negative. Here we propose a general method is able to go beyond this standard assumption model other possible underlying MI structures.

The proposed method follows a two-step iterative set pooling approach to address MIL problems. Given a set of elements, in a first step, each of the elements is encoded by an Instance Descriptor Unit. Then, in a second step, each of the feature-encoded elements is passed to the Iterative Set Pooling Unit. This unit is tasked with embedding and iteratively aggregating all the information from the different elements into a set-level representation (Fig. [1]). Our approach
is flexible in the sense that it is capable of characterizing sets of elements, i.e. modeling the underlying MI assumption, by iteratively looking at each of the elements that compose the set one at a time. Our extensive evaluation shows strengths of our method at three fronts. First, it is able to model several MI assumptions, e.g. single or multiple witness element detection, counting or collective assumptions. Second, it can address several type of prediction problems, i.e. classification and regression. Third, it obtains competitive or superior performance w.r.t. the state-of-the-art. All of this while requiring minimal modifications. Moreover, a deeper analysis of our iterative set pooling unit suggests that it is capable of highlighting, internally, the element (or group of elements) that triggers the given prediction, thus, possessing explainability capabilities.

Our contributions are three-fold: i) a novel iterative method powered with feedback mechanisms that is capable of modelling the underlying assumption / relationship that characterizes the elements in a set without the need of explicit heuristics, ii) a robust framework able to handle various typical assumptions considered by MIL problems, and iii) an approach to address MIL problems powered with explainability capabilities.

This paper is organized as follows: Section 2 describes related work. Section 3 introduces some of the MI assumptions traditionally considered in the literature. In Section 4 we present details of the proposed method. We conduct and extensive evaluation in Section 5 and conclude in Section 6.

2 Related Work
Over the last decade various approaches have been proposed to address several types of MIL problems. Since our work is based on deep neural networks, we position our approach w.r.t. efforts based on neural networks, specifically those with deep architectures. Please refer to (Amores 2013; Carbonneau et al. 2018; Foulds and Frank 2010) for detailed surveys covering non-deep methods.

(Ramon and De Raedt 2000) constitutes one of the first efforts towards addressing MIL problems through neural networks. The proposed multiple instance neural network (MINN) estimates instance probabilities which are aggregated at the last layer using a convex max operator in order to predict a set probability. This idea is further extended in (Wan 2018) which uses a neural network to learn a set representation and directly carry out set classification without estimating instance-level probabilities or labels. In parallel, (Ilse, Tomczak, and Welling 2018) proposed an attention mechanism to learn a pooling operation over instances. The weights learned for the attention mechanism on instances can serve as indicators of the contribution of each instance to the final decision – thus, producing explainable predictions. (Liu, Yan, and Ouyang 2017) proposed a similar idea, using the computed set representations, to measure distances between image sets. (Yan et al. 2018) proposed to update the contributions of the instances by observing all the instances of the set a predefined number of iterations. Along a different direction, (Tibo, Jaeger, and Frasconi 2018) proposed a hierarchical set representation in which each set is internally divided into subsets until reaching the instance level. Very recently, (Tu et al. 2019) proposed to consider the elements in the sets to be non-i.i.d. and used graph neural networks to learn a set embedding.

Similar to (Ilse, Tomczak, and Welling 2018; Wan 2018) we embed the instance features from each set into a common space from which a set representation is learned. This set representation is used to make directly set predictions related to MIL problems. Similar to (Tu et al. 2019) and (Yan et al. 2018) we aim at learning the underlying structure within the sets. Different from (Tu et al. 2019), our method does not rely on hand-tuned parameters, e.g. distance thresholds to define edges in the graph, and other manual graph construction. Moreover, the improvement in performance displayed by our method is not sensitive to the possible lack of structure within each set. Compared to (Yan et al. 2018), our method only requires a single pass through all the instances. Moreover, our method is able to go beyond binary classification tasks and handle more complex classification and regression tasks. Finally, most of the works mentioned above operate under the standard MI learning assumption. In contrast, the proposed approach is able to learn the underlying structure of set of instances, thus, being robust to several MIL assumptions/problems (Foulds and Frank 2010).

3 Multiple Instance Assumptions
Before describing the proposed approach, we introduce set characteristics or assumptions that have been commonly considered in order to define set-level labels. While different surveys (Amores 2013; Carbonneau et al. 2018; Foulds and Frank 2010) have grouped these assumptions based on different criteria, we focus on the following general assumptions, which can be adapted to meet more specific ones.

3.1 Standard Multiple Instance Assumption
Given the set $X = \{x_1, x_2, ..., x_m\}$ of instances $x_i$ with latent instance-level labels $C_i = \{c_1, c_2, ..., c_m\}$, traditional MIL problems aim at the prediction of binary set-level labels $y_j$ for each set $X_j$. Under the standard MI assumption, a set $X_j$ is positive, i.e. $y_j = 1$, if and only if, at least one of the elements/instances $x_i$ that compose it satisfies a predefined desired property $\alpha$.

$$y_j = \begin{cases} 1, & \text{if } \exists x_i \in X_j : c_i = \alpha \\ 0, & \text{if } \forall x_i \in X_j : c_i \neq \alpha \end{cases}$$

3.2 Collective Multiple Instance Assumption
Under the standard MI assumption only a small subset of elements, possessing a given property, contribute to the prediction of the set label. In contrast, under the collective MI assumption, all the elements $x_i$ contribute equally to the predicted set label.

$$y_j = \begin{cases} 1, & \text{if } P(y_j|X_j) > \tau \\ 0, & \text{otherwise} \end{cases}$$

where $\tau$ is a threshold value and the set-level score $P(y_j|X_j)$ is computed from the contributions $P(c_i = \alpha|x_i)$ of each of the $m$ elements $x_i$ as follows:

$$P(y_j|X_j) = \frac{1}{m} \sum_{i=1}^{m} P(c_i = \alpha|x_i)$$

(1)
An alternative point is defined by the collective weighted MI assumption under which different elements \(x_i\) may have different degrees of influence in the label of the set as determined by their weight \(w(x_i)\). Under this weighted assumption, the set score \(Pr(y_j|X_j)\) is computed as:

\[
Pr(y_j|X_j) = \frac{1}{z} \sum_{i=1}^{z} w(x_i) Pr(c_i = \alpha|x_i)
\]

with the normalization term \(z = \sum_{i=1}^{m} w(x_i)\).

### 3.3 Rank-based MI Assumption

This assumption assumes that there exists a property \(r(.)\) for every element \(x_i\) in the set under which the elements, or a subset of them, can be ranked on a fixed order. Taking the property \(r\) into account, a set is considered positive if its elements follow a specific order (as in sequential data). Otherwise it is considered negative. More formally:

\[
y_j = \begin{cases} 
1, & \text{if } r(x_1) \prec r(x_2) \prec r(x_3) \prec \ldots \prec r(x_m) \\
0, & \text{otherwise}
\end{cases}
\]

This type of assumption is related to problems processing sequential data, e.g. action recognition (Sharma, Kiros, and Salakhutdinov 2015) or language modeling (Sundermeyer, Schlüter, and Ney 2012).

### 4 Proposed Method

The proposed approach consists of three main components. Given a set \(X = \{x_1, x_2, \ldots, x_m\}\) of \(m\) elements \(x_i\), each of the elements \(x_i\) is encoded into a feature representation \(f_i\) through the **Instance Description Unit** (Section 4.1). Then, each element is fed to the **Iterative Set Pooling Unit** (Section 4.2), producing the aggregated set representation \(S\). Finally, a prediction \(\hat{y}\) is obtained by evaluating the set representation via the **Prediction Unit** (Section 4.3).

#### 4.1 Instance Description Unit

This component receives the set elements in raw form, i.e. each of the instances \(x_i \in \mathbb{R}^{[m \times d]}\) that compose it, in its original format. It is tasked with encoding the input set data into a format that can be processed by the rest of the pipeline.

As such, it provides the proposed method with robustness to different data formats/modalities. More formally, given a dataset \(\{X_j, y_j\}\) of sets \(X_j\) paired with their corresponding set-level labels \(y_j\), each of the sets \(X_j\) is encoded into a feature \(F_j = \{f_1, f_2, \ldots, f_m\}\). This is achieved by pushing each of the elements \(x_i\) that compose it, through a feature encoder \(\tau(.)\) producing the instance-level representation \(f_i = \tau(x_i), f_i \in \mathbb{R}^{[m \times n]}\).

Selection of this component depends on the modality of the data to be processed, e.g. VGG (Simonyan and Zisserman 2014) or ResNet (He et al. 2015) features for still images, Word2Vec (Mikolov et al. 2013) or TF-IDF for text data or rank-pooled features (Fernando et al. 2016) or dynamic images (Bilen et al. 2018) for video data.

#### 4.2 Iterative Set Pooling Unit

The main goal of this component is to derive a set-level representation \(S_j \in \mathbb{R}^{[m \times n]}\) that is able to encode all the elements \(x_i\), and any possible underlying structure between them. We aim at learning a set representation that is independent of both the cardinality \(m\) of the set and the nature of the underlying structure. Starting from the element-level representations \(F_j\) computed in the previous step, this is achieved by iteratively looking at the representations \(f_i\), from each of the elements \(x_i\), one at a time. In each iteration \(i\) an updated set-level representation \(S^i_j\) is computed. In parallel, a feedback loop provides information regarding the state of the set representation that will be considered at the next iteration \(i + 1\). Finally, after observing all the \(m\) elements \(x_i\) in the set, the final set representation \(S^m_j\) is taken as the output \(S_j\) of this component.

The notion behind this iterative set pooling idea is that elements observed a specific iterations can be used to compute a more informed set-level representation at later iterations. Thus, allowing to encode underlying relationships or structures among the elements of the set. While this iterative assumption may hint at a sequence structure requirement within each set, our empirical evaluation strongly suggests this not to be the case. Moreover, this provides the proposed approach with robustness towards sets possessing a sequence-like structure, while not enforcing the requirement of the existence of such a structure.

In practice, this iterative mechanism can be implemented through Recurrent Neural Networks (Schuster and Paliwal 1997), Long Short Term Memory (LSTM) Networks (Hochreiter and Schmidhuber 1997), Gated Recurrent Units (Cho et al. 2014), or any other machinery with means to allow information persistence across multiple observations \(x_i\). Here, we implement this component through LSTMs given their robustness of modeling structures within a set with high cardinality. This will ensure that the learned set representation can encode structures between all the elements in the set, independently of the cardinality of the set. More specifically, we use Bi-directional LSTMs which observe the elements in a set from the left-to-right and right-to-left directions. This will further ensure that the context in which the elements of the set occur is properly modelled.
4.3 Prediction Unit

Having a set-level representation $S_j$ for set $X_j$, this component is tasked with making a set label prediction $\hat{y}_j = g(S_j)$ that will serve as final output for the pipeline. The selection of the prediction function $g(.)$ is related to the task of interest. This unit provides our method with flexibility to address both classification and regression prediction tasks.

4.4 Explaining Model Predictions

Up to this point, we have presented an iterative method to make predictions from a set-level representation $S_j$ through the use of a prediction function $g(.)$. While being able to make accurate predictions is of importance, being able to provide an explanation supporting the prediction made is a desirable property for any automatic system. In MIL algorithms, these explanations usually come in the form of highlighting the elements or instances $x_i^*$ of the set which determine the predicted set label $\hat{y}_j$.

In the proposed approach this can be achieved by probing the set representation $S_j$ after each of the elements $x_i$ are embedded on it. More specifically, on an initial step we can push every element $x_i$ through the set pooling unit and store the set representation $S_j^0$ computed after the embedding of the $i^{th}$ element. Then, the relevant elements $x_i^*$ can be highlighted by identifying the elements $x_i$ with strong effect in the computed set representation $S_j^0$. Finally, the selection of elements $x_i^*$ can be further verified, by the response $\hat{y}_j^* = g(S_j^0)$ that their corresponding set-level representations $S_j^0$ produce when evaluated by the prediction unit.

5 Experimental Evaluation

We conduct a series of experiments to assess empirically the performance of the proposed approach under different data modalities and considering different MI assumptions.

5.1 Drug Activity Prediction

First, we conduct experiments on the Drug Activity Prediction task proposed in (Foulds and Frank 2010). This is a standard benchmark used to assess the performance of MIL methods. It is composed of two sets, i.e., MUSK1 and MUSK2 which consist of 47/45 and 39/63 positive/negative sets, respectively. Elements within a set correspond to different conformations of a molecule, with each conformation being described by a 166-dimensional vector. The task is to predict whether new molecules will be musks or non-musks.

**Discussion:** A quick glance at Table 1 shows that the proposed method has comparable performance as state-of-the-art methods. More specifically, on MUSK1 it surpasses the closest method by ~2 percentage points (pp). For the case of MUSK2, its performance is on par with most of the competitors, with the exception of the Dynamic Pooling method which achieves superior performance at the cost of a larger number of computations. Overall the proposed method achieves ~90% classification accuracy on this task.

5.2 MI Predictions on Simplified Visual Data

This experiment focuses on performing MI predictions based on visual data. Following the protocol from (Ilse, Tomczak, and Welling 2018) we use images from the MNIST dataset (LeCun and Cortes 2010) to construct image sets to define four scenarios, each following a different assumption: Single digit occurrence, Multiple digit occurrence, Digit sequences and Single digit counting. For each scenario we sample images from MNIST to construct 500 image sets for training and 200 sets for testing. Label balance is preserved within each data split.

For this series of experiments, we use a LeNet (Lecun et al. 1998) as instance descriptor unit and a LSTM with an input and cell state with 500 dimensions, respectively. We compare the obtained performance w.r.t. the attention-based model from (Ilse, Tomczak, and Welling 2018) and the dynamic pooling method from (Yan et al. 2018). Mean error rate in the binary classification task is adopted as performance metric in these experiments.

**Single Digit Occurrence** In this scenario we follow the standard MI assumption and label a set as positive if at least one digit '9' occurs in the set. The digit '9' is selected since it can be easily mistaken with digit '4' and '7' (Ilse, Tomczak, and Welling 2018), thus, introducing some element-level ambiguity. We define sets with mean cardinality $m=10$, and verify the effect that $m$ has on performance by testing two standard deviation values, $\sigma=2$ and $\sigma=8$. We repeat this experiment five times generating different sets and weight initializations. We report mean performance in Table 2 (column II and III).

**Discussion:** The results indicate that, in this task, our performance is comparable with the state-of-the-art for lower values of $\sigma$ and superior as $\sigma$ increases. This is to some extent expected, since at lower $\sigma$ the cardinality (i.e. the number of elements) of each set is almost fixed. This setting is favorable for the attention-based method since it operates in a feed-forward fashion. Yet, note the high standard deviation in performance produced by this baseline. On the contrary, at higher $\sigma$ values there is a higher variation of cardinality across sets. Under this setting, feed-forward approaches start to produce higher errors. Here our method produces superior performance, ~1.4 percentage points (pp) w.r.t. to the state-of-the-art.

**Multiple Digit Occurrence** This is an extension of the previous scenario in which instead of focusing on the occurrence of a single digit class, the model should recognize the occurrence of instances of two digit classes. More specifically, a set is labeled positive if both digits ‘3’ and ‘6’ occur in it, without considering the order of occurrence. For this

Table 1: Performance on the drug activity prediction task

| Method                  | MUSK1   | MUSK2   |
|-------------------------|---------|---------|
| Atten. Based            | 89.2 ±4.0 | 85.8 ±4.8 | |
| Gated Atten. Based      | 90.0 ±5.0 | 85.8 ±4.8 | |
| Dyn. Pool               | 90.7 ±3.6 | **92.6 ±4.3** | |
| Ours                    | **93.2 ±6.0** | 85.4 ± 5.5 | |

(1) Please refer to the supplementary material for more details.
Table 2: Mean error rate (in percentage points) of experiments considering digits from the MNIST dataset.

| Method          | single digit ($\sigma=2$) | single digit ($\sigma=8$) | multiple digits | digit sequence | digit counting |
|-----------------|----------------------------|----------------------------|-----------------|----------------|---------------|
| Atten. Based    | 2.8 ± 4.8                  | 4.5 ± 0.4                  | 28.5 ± 0.7      | 47.3 ± 3.2     | 33.4 ± 19.3   |
| Gated Atten. Based | 4.0 ± 0.9                  | 4.6 ± 0.5                  | 27.4 ± 0.9      | 47.0 ± 2.9     | 11.9 ± 3.6    |
| Dyn. Pool       | 5.6 ± 1.1                  | 6.1 ± 1.2                  | 28.5 ± 6.6      | 47.9 ± 2.5     | 25.4 ± 1.8    |
| Ours            | 3.5 ± 1.1                  | 3.1 ± 0.5                  | 6.4 ± 1.4       | 2.8 ± 0.7      | 9.0 ± 2.7     |

scenario 1,000 sets are sampled for training. Results are reported in Table 2 (column IV).

Discussion: It is remarkable that when making this simple extension of considering the occurrence of multiple digits, i.e. ‘3’ and ‘6’, the state-of-the-art methods suffer a significant drop in performance. This drop put the state-of-the-art extension of considering the occurrence of multiple digits, namely Lookbook (Yoo et al. 2016). Images in this dataset are divided into two domains: catalog clothing images and their corresponding human model images where a person is wearing the clothing product. Each clothing product has one catalog image and several human model images. We only consider the products with five or more human model images, resulting in 6616 unique products (latent classes $c_i$) with around 63k images in total. Every product image has 5-55 human model images. The training set contains 4000 classes while the validation and test sets have 616 and 2000 classes, respectively. We run two experiments on this dataset as described in the following sections.

5.3 MI Predictions on Realistic Visual Data

We complement experiments from Section 5.2 by considering sets composed by more complex visual data. Towards this goal, we use images from a fashion-related dataset, specifically Lookbook (Yoo et al. 2016). Images in this dataset are divided into two domains: catalog clothing images and their corresponding human model images where a person is wearing the clothing product. Each clothing product has one catalog image and several human model images. We only consider the products with five or more human model images, resulting in 6616 unique products (latent classes $c_i$) with around 63k images in total. Every product image has 5-55 human model images. The training set contains 4000 classes while the validation and test sets have 616 and 2000 classes, respectively. We run two experiments on this dataset as described in the following sections.

Discussion: As can be seen Table 2 under this scenario, the proposed method leads the performance table by a large margin of ~45 pp. This is to some extent expected since the LSTM network used to implement our iterative set pooling unit is designed to handle sets whose instances posses an underlying sequential structure.

Digit Sequences

Similar to the previous setting, in this scenario multiple elements are of interest within each set, however, the order of occurrence of these do matter. More specifically, a set is labeled positive if an instance of digit ‘3’ occurs earlier, i.e. it has a lower index in the set, than one of digit ‘6’. This scenario follows the Rank-based MI assumption presented in Section 6 Quantitative results are reported in Table 2 (column V).

Discussion: As can be seen Table 2 (column VI) the same trend can be observed: our method has superior performance and higher stability than the attention-based model. When conducting this counting task, our method obtains a performance that is superior by 24 pp w.r.t. the attention-based model and by 16 pp w.r.t. the dynamic pooling. These results support the capability of the method to handle regression problems.

Outlier Detection

This is a binary classification task where the goal is to indicate whether a given set of images contains an outlier image. Image sets in this experiment are composed exclusively by human model images of the same clothing product (class). This experiment follows an inverse version of the standard MI assumption where sets are considered positive, i.e without outliers, if all its constituent images (elements) belong to the same clothing product. Otherwise, if one of the images in the set does not belong to the clothing product, the set is considered negative, i.e. with an outlier. In this experiment all the sets have 5 images.

Discussion: The result in Table 2 indicates that our method still achieves superior performance in a realistic dataset. More concretely, our method produces a mean error that is 3 pp lower w.r.t. attention-based method and significant 31 pp lower w.r.t. the dynamic pooling baseline. It is noteworthy that while somewhat related to the standard MI
Table 3: Performance on the outlier detection experiment.

| Method       | Error rate |
|--------------|------------|
| Attn. Based  | 14.0 ± 0.9 |
| Dyn. Pool    | 42.1 ± 0.3 |
| Ours         | 10.9 ± 0.5 |

Cross-domain clothing retrieval For this experiment, human model images are used as queries while catalog images serve as database, thus, defining a many-to-one retrieval. The cardinality of each set is the same as the number of human model images of each product (class). We conduct two variants of this experiment. On the first variant we use the complete image, as it is originally provided. The second is an occluded variant where every human model image in a set is divided into a 4 × 4 grid of 16 blocks. 12 of these blocks are occluded by setting all the pixels therein to black. By doing so, every single image in a set can only show part of the information while their combination (i.e. the whole set) represents the complete clothing item. Catalog images in the database are not occluded in this experiment.

As baselines, in addition to the attention-based model we follow DeepFashion [Liu et al. 2016], and train a model to perform retrieval by computing the distances by considering single image representations instead of set-based representations. Following the multiple queries approach from [Arandjelović and Zisserman 2012], we report performance of three variants of this method: Single-AVE, where the distance of each set is computed as the average of the distances from every image in the set w.r.t. an item in the database; Single-MIN, where the distance of the set is defined as the minimum distance of an image in the set w.r.t. an item in the database; and Single Fea. AVE, where the distance of the set is calculated as the distance of a prototype element w.r.t. an item in the database. As prototype element we use the average feature representation of \( \mathbf{f} \), from the representation \( f \), of every element in the set. We refer to these baselines as Single-image models.

This retrieval task is to some extent related to the collective MI assumption (Sec. 3) since all the elements in the set contribute to the task handled by the model.

Discussion: Table 3 shows that in the original setting our method tends to obtain superior recall values in the majority of the cases, with the exception of the case when the closest 20 items (recall@20) are considered. When looking at the occluded variant of the experiment, a quick glance at Table 3 shows that, compared to the original setting, absolute performance values on this setting are much lower. This is to be expected since this is a more challenging scenario where the model needs to learn the information cumulatively by aggregating information from parts of different images. In this occluded setting, our method clearly outperforms the attention-based, dynamic pooling and the methods based on single-image distances. This could be attributed to the information persistence component that is part of our method. This component allows our method to select what to remember and what to ignore from each of the elements that it observes when updating the set representation used to compute distances. The difference w.r.t. to the Single-AVE and Single-MIN baselines is quite remarkable given that they require a significant larger number of element-wise distance computations w.r.t. items in the database. This may lead to scalability issues when the dataset size increases, as the computation cost will grow exponentially.

Moreover, in both occluded and non-occluded datasets, we notice that the Single-image model baselines have a superior performance w.r.t. the attention-based model and dynamic pooling model. We hypothesize that is because the single-image models can better exploit important features, e.g. discriminative visual patches, since they compute distances directly in an element-wise fashion. In contrast, it is likely that some of these nuances might get averaged out by the feature aggregation step that is present in the attention-based model.

5.4 Explaining Model Predictions

In this section we analyze the explanation capabilities of our method. Towards this goal, in Fig. 3 we show the predicted output after observing each element of the set. Since the set pooling unit utilizes a Bi-LSTM, which processes forward and backward directions of the set together, we show the two directions of the set. In addition, we verify the capabilities of the proposed set representation to encode the underlying MI assumption. This could be indicated by reflecting signif-
Figure 3: t-SNE visualization of the learned set representation. The first two rows show examples of predictions on true positive and true negative sets, except for the Digit Counting experiment, which shows two sets containing 4 and 0, elements of interest, respectively. The third row shows the prediction of 20 examples overlaid on the t-SNE space for the digit-based experiments.

icant variations in the $S_i^j$ when observing the elements $x_i$ involved in the MI assumption. We ease the visualization of the high-dimensional $S_i^j$ representation by plotting its corresponding t-SNE (van der Maaten and Hinton 2008) projection in Fig. 3. See the supp. material for more examples.

Discussion: In Fig. 3, we can notice that each time one of the elements that determine the MI assumption are observed, the set representation $S_i^j$ is updated in such a way that there is a significant change in the prediction made by the model. This is further supported by the state of the internal representation $S_i^j$ as shown by the corresponding t-SNE visualizations (Fig. 3). For the Single Digit Occurrence case, we notice that the representation gets updated to a different region of the space when the digit of interest is observed. More specifically, from the third row of Fig. 3 it is clear that the space is divided into two parts: the set representation of negative sets changes within the bottom-left region, while for positive sets, once the digit of interest occurs, the set representation jumps to the top-right region and ends there. Similarly, for the Multiple Digit Occurrence, Digit Sequences and Digit Counting cases, we notice that the representation shifts, significantly, to specific regions (green and magenta dots) every time one of the digits of interest is observed. Moreover, for Multiple Digit Occurrence and Digit Sequences the representation seems to always reach a common region once the underlying MI assumption has been completely satisfied.

6 Conclusion

We presented an iterative approach to address MIL problems. Our method is capable of learning the underlying structure that characterizes each of the sets by looking at its constituent elements one at a time. Despite its simplicity the proposed method is able to effectively model a variety of underlying MI assumptions and handle both classification and regression task while requiring minimum modifications. A deeper analysis of the learned set representation reveals that our method is able to highlight witness elements which are relevant to the underlying set structure, thus, providing some explanation capabilities.

References

[Amores 2013] Amores, J. 2013. Multiple instance classification: Review, taxonomy and comparative study. Artif. Intell. 201:81–105.

[Arandjelovic and Zisserman 2012] Arandjelovic, R., and Zisserman, A. 2012. Multiple queries for large scale specific object retrieval. In BMVC.

[Bilen et al. 2018] Bilen, H.; Fernando, B.; Gavves, E.; and Vedaldi, A. 2018. Action recognition with dynamic image networks. TPAMI 40(12):2799–2813.

[Briggs, Fern, and Raich 2012] Briggs, F.; Fern, X. Z.; and Raich, R. 2012. Rank-loss support instance machines for miml instance annotation. In KDD. 534–542.
Figure 4: Prediction score post set representation update after observing each element from the set. For reference, we present both the forward (top) and backward (bottom) directions in which the elements of the set are observed.

[Carbonneau et al. 2018] Carbonneau, M.-A.; Cheplygina, V.; Granger, E.; and Gagnon, G. 2018. Multiple instance learning: A survey of problem characteristics and applications. ArXiv abs/1612.03365.

[Cho et al. 2014] Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In EMNLP.

[Elmoufidi 2018] Elmoufidi, A. 2018. Anomaly classification in digital mammography based on multiple-instance learning. IET Image Processing 12:320–328(8).

[Fernando et al. 2016] Fernando, B.; Gavves, E.; Oramas M., J.; Ghodrati, A.; and Tuytelaars, T. 2016. Rank pooling for action recognition. In TPAMI.

[Foulds and Frank 2010] Foulds, J., and Frank, E. 2010. A review of multi-instance learning assumptions. The Knowledge Engineering Review 25(1):125.

[He et al. 2015] He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep residual learning for image recognition. In CVPR.

[Hochreiter and Schmidhuber 1997] Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. Neural Computation 9(8):1735–1780.

[Ilse, Tomczak, and Welling 2018] Ilse, M.; Tomczak, J. M.; and Welling, M. 2018. Attention-based deep multiple instance learning. arXiv preprint arXiv:1802.04712.

[LeCun and Cortes 2010] LeCun, Y., and Cortes, C. 2010. MNIST handwritten digit database.

[Lecun et al. 1998] Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE.

[Liu et al. 2016] Liu, Z.; Luo, P.; Qiu, S.; Wang, X.; and Tang, X. 2016. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[Liu, Yan, and Ouyang 2017] Liu, Y.; Yan, J.; and Ouyang, W. 2017. Quality aware network for set to set recognition.

[Mikolov et al. 2013] Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In Burges, C. J. C.; Bottou, L.; Welling, M.; Ghahramani, Z.; and Weinberger, K. Q., eds., Advances in Neural Information Processing Systems 26. Curran Associates, Inc. 3111–3119.

[Ramon and De Raedt 2000] Ramon, J., and De Raedt, L. 2000. Multiple instance neural networks. In ICML Workshop on Attribute-value and Relational Learning.

[Sammut and Webb 2011] Sammut, C., and Webb, G. I. 2011. Multi-instance learning. Encyclopedia of Machine Learning.

[Schuster and Paliwal 1997] Schuster, M., and Paliwal, K. K. 1997. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing 45(11):2673–2681.

[Sharma, Kiros, and Salakhutdinov 2015] Sharma, S.; Kiros, R.; and Salakhutdinov, R. 2015. Action recognition using visual attention. CoRR abs/1511.04119.

[Simonyan and Zisserman 2014] Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.

[Sudharshan et al. 2019] Sudharshan, P.; Petitjean, C.; Spanhol, F.; Oliveira, L. E.; Heutte, L.; and Honeine, P. 2019. Multiple instance learning for histopathological breast can-
cer image classification. *Expert Systems with Applications* 117:103 – 111.

[Sundermeyer, Schlüter, and Ney 2012] Sundermeyer, M.; Schlüter, R.; and Ney, H. 2012. LSTM neural networks for language modeling. In *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*, 194–197.

[Tibo, Jaeger, and Frasconi 2018] Tibo, A.; Jaeger, M.; and Frasconi, P. 2018. Learning and interpreting multi-multi-instance learning networks. *arXiv:1810.11514*.

[Tu et al. 2019] Tu, M.; Huang, J.; He, X.; and Zhou, B. 2019. Multiple instance learning with graph neural networks. In *ICML Workshop on Learning and Reasoning with Graph-Structured Representations*.

[van der Maaten and Hinton 2008] van der Maaten, L., and Hinton, G. 2008. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*.

[Wan 2018] 2018. Revisiting multiple instance neural networks. *Pattern Recognition* 74:1524.

[Wei and Zhou 2016] Wei, X.-S., and Zhou, Z.-H. 2016. An empirical study on image bag generators for multi-instance learning. *Machine Learning* 105(2):155–198.

[Yan et al. 2018] Yan, Y.; Wang, X.; Guo, X.; Fang, J.; Liu, W.; and Huang, J. 2018. Deep multi-instance learning with dynamic pooling. In *Asian Conference on Machine Learning*.

[Yoo et al. 2016] Yoo, D.; Kim, N.; Park, S.; Paek, A. S.; and Kweon, I. 2016. Pixel-level domain transfer. *ECCV*.