Multiple Instance Learning with Absolute Position Information

Meera Krishnamoorthy MEERAK@UMICH.EDU
Jenna Wiens WIENSJ@UMICH.EDU
Computer Science & Engineering, University of Michigan, Ann Arbor, MI, USA

Abstract

Most past work in multiple instance learning (MIL), which maps a group or bag of instances to a classification label, has focused on settings in which the order of instances does not contain information. In this paper, we define MIL with *absolute* position information: tasks in which instances of importance remain in similar positions across bags. Such problems arise, for example, in MIL with medical images in which there exists a common global alignment across images (e.g., in chest x-rays the heart is in a similar location). We also evaluate the performance of existing MIL methods on a set of new benchmark tasks and two real data tasks with varying amounts of absolute position information. We find that, despite being less computationally efficient than other approaches, transformer-based MIL methods are more accurate at classifying tasks with absolute position information. Thus, we investigate the ability of positional encodings, a mechanism typically only used in transformers, to improve the accuracy of other MIL approaches. Applied to the task of identifying pathological findings in chest x-rays, when augmented with positional encodings, standard MIL approaches perform significantly better than without (AUROC of 0.799, 95% CI: [0.791, 0.806] vs. 0.782, 95% CI: [0.774, 0.789]) and on-par with transformer-based methods (AUROC of 0.797, 95% CI: [0.790, 0.804]) while being 10 times faster. Our results suggest that one can efficiently and accurately classify MIL data with absolute position information using standard approaches by simply including positional encodings.

Data and Code Availability. This paper uses the MNIST dataset, which is publicly available (Deng, 2012), and the MIMIC-CXR dataset, which is available on the PhysioNet repository (Johnson et al., 2019). Our training and evaluation code, dependency specifications, and the list of images we use to train and evaluate all MIL methods on the real data tasks are available at https://github.com/MLD3/MILwAPI.

Institutional Review Board (IRB). This work is not regulated as human subjects research since data are de-identified and publicly available.

1. Introduction

In standard supervised learning, one instance is mapped to one label. In contrast, in multiple instance learning (MIL), several instances grouped in a bag are mapped to one label (Ilse et al., 2018). In recent years, MIL approaches have been applied to high-resolution imaging data. For example, in applications of computer vision to medical imaging tasks, to avoid the potential information loss that comes with downsampling images by 10 to 10,000 fold, machine learning practitioners instead divide large images into smaller patches that maintain resolution and then perform classification on the bag of patches (Lu et al., 2021; Wang et al., 2017). This process can result in improved classification accuracy over downsampling approaches (Seibold et al., 2021).

Past work in MIL has been largely evaluated in settings where the order of instances does not contain information (Zhang et al., 2022). However, this assumption does not always hold, especially in settings where there is a natural alignment across images. Thus, in this work, we formalize MIL with *absolute position information*. We define tasks with absolute position information as those in which the position of specific patches is consistent across bags and therefore useful for classifying those bags. The number and location of those patches can be unknown. There are many real-world MIL tasks with absolute position information: for example, pathological findings in chest x-rays (Figure 1) and MRIs (Johns Hopkins Medicine) often occur in similar positions across images.

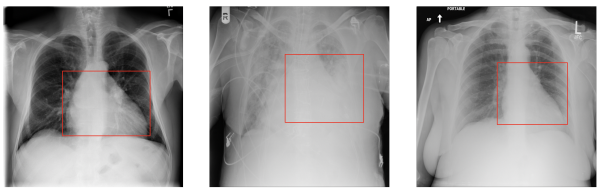

Figure 1: Examples of data with absolute position information. The three chest x-rays above from the publicly available CXR8 dataset (Wang et al., 2017) contain evidence of cardiomegaly, a condition that manifests on a chest x-ray as an enlarged heart. This evidence appears in similar positions across all images.

Along with formalizing MIL with absolute position information, we develop a set of benchmark tasks with varying amounts of absolute position information. We evaluate the performance of existing MIL methods on these tasks as well as on two real data tasks with different amounts of absolute position information that involve chest x-ray classification (Johnson et al., 2019). Currently, only transformer-based MIL approaches are designed to leverage position information (Shao et al., 2021), and thus are more accurate than existing non-transformer-based MIL methods on tasks with absolute position information. However, recent work has emphasized the need for computationally efficient alternatives to transformers (Poli et al., 2023).

Thus, we aim to improve the accuracy of non-transformer-based MIL methods, which are more computationally efficient than transformer-based MIL methods, on tasks with absolute position information. We perform an ablation study to understand the mechanisms that allow transformers to achieve better accuracy on tasks with absolute position information compared to non-transformers. Based on our findings, we develop a positional encoding wrapper that applies to non-transformer-based MIL methods. While positional encodings are not new (Vaswani et al., 2017), their use with non-transformer-based approaches has not been explored in the context of MIL. Though simple, our wrapper is effective at improving the accuracy of non-transformer-based MIL methods on benchmark and real data tasks, on par with that of transformers while being significantly faster.

Our contributions are as follows.

- **Problem Formalization.** We formalize the MIL problem with absolute position information.
- **Benchmark Task Creation.** We develop a set of benchmark tasks with varying amounts of absolute position information.
- **Baseline Evaluation.** We evaluate existing MIL methods on the benchmark and real data tasks. We find that transformers outperform current non-transformer-based methods on tasks with absolute position information.
- **Transformer Ablation Study.** We study the mechanisms that allow transformers to learn position information and find that transformers with the original attention mechanism can only learn position information via positional encodings, while transformers with the Nystrom-based attention mechanism can learn position information implicitly via their self-attention mechanism.
- **Wrapper.** We propose a wrapper that augments non-transformer-based MIL methods with positional encodings, a mechanism typically only used with transformers. This approach improves the accuracy of non-transformer-based MIL methods on a variety of tasks with absolute position information, on par with that of transformers while being significantly faster.

2. Background

In this section, we first define the MIL setting on which we focus: binary weakly supervised MIL, a popular setting explored in most past work in MIL (Ilse et al., 2018; Campanella et al., 2019; Zhang et al., 2022). We then describe the general structure of the MIL methods (both transformer-based and non-transformer-based) designed to solve this task. It is on this general structure that our proposed wrapper will apply.

2.1. Binary Weakly Supervised MIL

In binary MIL, one aims to predict a binary label $Y \in \{0, 1\}$ for a bag of instances $X = \{x_1, \ldots, x_n\}$, where each instance x_i consists of a feature vector or matrix. A bag has the label $Y = 1$ if at least one of the instances i in the bag has the instance-level

label $y_i = 1$. Otherwise, the bag has the label $Y = 0$. Fully supervised MIL and weakly supervised MIL differ in the labeled data available at training time. At training time, fully supervised MIL methods assume access to both the bag label Y and instance labels y_i for all bags and instances. In contrast, weakly supervised MIL methods like the ones we consider in this paper assume access only to the bag label Y at training time (Ilse et al., 2018). At inference time in both settings, one only has access to the bag of instances X and aims to predict bag label Y.

2.2. Components of an MIL Method

All weakly supervised MIL methods consist of two components (Ilse et al., 2018): a feature extractor that transforms instances x_i into lower dimensional feature vectors z_i; and an aggregator that maps all the feature vectors z_i learned from instances within a bag to a single classification probability $p \in [0, 1]$. Most MIL methods were developed to classify image data and therefore use a convolutional neural network as a feature extractor. The aggregator varies across MIL methods. Non-transformer-based approaches typically assume that instances are independently and identically distributed and thus use permutation invariant aggregators (Ilse et al., 2018; Lu et al., 2021; Zhang et al., 2022; Seibold et al., 2021), whereas transformer-based approaches incorporate position information via transformer-based aggregators (Shao et al., 2021; Zhao et al., 2022; Wölflein et al., 2023).

3. Related Work

3.1. Types of Position Information

Most past work in MIL has assumed that the order of instances does not contain information. In contrast, most work involving transformers has assumed that the order of inputs contains information and uses positional encodings to learn from this type of information. Specifically, the positional encodings in this past work were designed to learn absolute position information (Vaswani et al., 2017) and relative position information (Shaw et al., 2018; Shao et al., 2021). A dataset has relative position information if the distance between elements of importance within inputs is similar across inputs. Tasks with absolute position information will also contain relative position information if more than one element occurs in a similar position across inputs. However, a task with relative position information does not imply the presence of absolute position information. In this paper, we focus on MIL tasks with absolute position information.

3.2. MIL Methods and Tasks in Past Work

Most work in MIL has focused on developing methods to classify histopathology images (Ilse et al., 2018; Lu et al., 2021; Shao et al., 2021; Zhang et al., 2022). Early work in this area assumed that there was no information in the order of instances in this task. Recent work, however, *hypothesized* that the task of classifying histopathology images contains absolute and relative position information, and proposed transformer-based solutions (Shao et al., 2021; Zhao et al., 2022; Wölflein et al., 2023). However, as Zhang et al. showed, transformers do not always outperform standard approaches that do not leverage position in tasks involving histopathology images (Zhang et al., 2022). This suggests that there is not much position information in histopathology image classification tasks. In retrospect, this is perhaps unsurprising since there is no guarantee of global alignment across histopathology images (i.e. instances of importance are not guaranteed to be in similar positions across histopathology images (Ilse et al., 2018)).

Beyond tasks in histopathology, MIL methods have been developed to classify multiple views of medical image data (i.e. multiple views of a breast ultrasound or an echocardiogram) (van Tulder et al., 2021; Huang et al., 2024), where each instance is a different view of the medical image data. There is global alignment across these views, which allows for the consolidation of information across multiple images. However, the order of views relative to each other is not considered useful for classification, and thus these tasks do not have absolute position information.

Additionally, Seibold et al. developed an MIL method to classify chest x-rays, a task that we hypothesize has absolute position information (Seibold et al., 2021). However, the method they proposed is invariant to the order of instances. Thus, we hypothesize that it will not perform as well as transformers, which explicitly leverage position.

3.3. Efficient Methods for Leveraging Position

Applied to MIL tasks with absolute position information, we hypothesize that transformer-based methods will outperform standard approaches. One drawback of transformers, however, is their computational inefficiency due to their use of self-attention (Poli et al.,

2023). As an alternative to transformers, one naive solution to an MIL task with absolute position information might involve simply removing instances irrelevant to classification based on absolute position information (i.e., cropping large images to the patches of importance). However, removing irrelevant instances assumes certainty in the absolute position information (i.e., important patches never appear in specific regions). Such certainty is unlikely in practice. For example, the locations of objects in chest x-rays can vary based on how the x-ray was obtained, which introduces uncertainty in the absolute position information (Figure 1). Recognizing a need for MIL approaches that are computationally efficient and can leverage absolute position information subject to uncertainty, we develop a simple, efficient method that allows non-transformer-based methods to leverage absolute position information.

4. Methods

In MIL tasks with absolute position information, there is some information in the order of instances. Thus, we do not use the term *bag*. Instead, we consider each example as a *list* of instances. Below, we formalize our definition of absolute position information within an MIL dataset and present our proposed approach for leveraging such information.

Our definition of absolute position information depends on instance labels. Note that because we are in a weakly supervised MIL setting, we may not have access to instance labels during training and testing, and therefore cannot always measure absolute position information for a given dataset. Using synthetic data in which we have ground truth instance labels, we apply this definition, varying the amount of absolute position information. We explore how classification performance varies with absolute position information.

4.1. Formalization of Absolute Position Information.

Consider dataset D defined as

$$D = \{X^{(i)}, Y^{(i)}\}_{i=1}^{N},$$

where each $X^{(i)}$ is a list of instances $x_j^{(i)} \in \mathbb{R}^{d' \times d \times d}$ for $j = 1, \ldots, n$

$$X^{(i)} = [x_1^{(i)}, \ldots, x_n^{(i)}]$$

Thus, D contains N lists and each list $X^{(i)}$ contains n instances, where n is the same across lists.

Each instance has a corresponding binary label $y_j^{(i)} \in \{0, 1\}$. The label $Y^{(i)}$ of each list $X^{(i)}$ is the maximum of the instance labels $y_j^{(i)}$

$$Y^{(i)} = \max([y_1^{(i)}, \ldots, y_n^{(i)}])$$

Thus, similar to how other MIL work defines their bag-level labels, $Y^{(i)} = 1$ if at least one instance $y_j^{(i)} = 1$. Otherwise, $Y^{(i)} = 0$.

We define the percentage of absolute position information in dataset D, G_D, using $c_j \in \mathbb{Z}$ and $N^+ \in \mathbb{Z}$, where c_j is the number of instances with label $y = 1$ that occur in position j in all lists in the dataset,

$$c_j = \sum_{i=1}^{N} y_j^{(i)}$$

and N^+ is the total number of instances within all lists with the label $y = 1$ in the dataset

$$N^+ = \sum_{j=1}^{n} c_j$$

Informally, we define absolute position information based on the expected benefit of leveraging the position of instances during classification. Consider a dataset in which every instance with the corresponding list label $Y^{(i)} = 1$ has the instance label $y_j^{(i)} = 1$ for all j. This dataset has 0% absolute position information, because the position of an instance in the lists with label $Y^{(i)} = 1$ is not useful for classification. In other words, as the number of positions where instances with the label $y_j^{(i)} = 1$ could appear grows, the less useful the positions of those instances are for classification, and thus the less absolute position information the corresponding dataset has. Therefore, when comparing datasets that have the same parameters except for the number of positions where instances have the label $y_j^{(i)} = 1$, the dataset with the fewest positions where instances have the label $y_j^{(i)} = 1$ will have most absolute position information. As a real-world example, chest x-rays where pathological findings are localized to a small number of positions within the heart have more absolute position information than chest x-rays where pathological findings could be anywhere in the chest.

Applying this intuition, absolute position information is minimized when all instances with the label

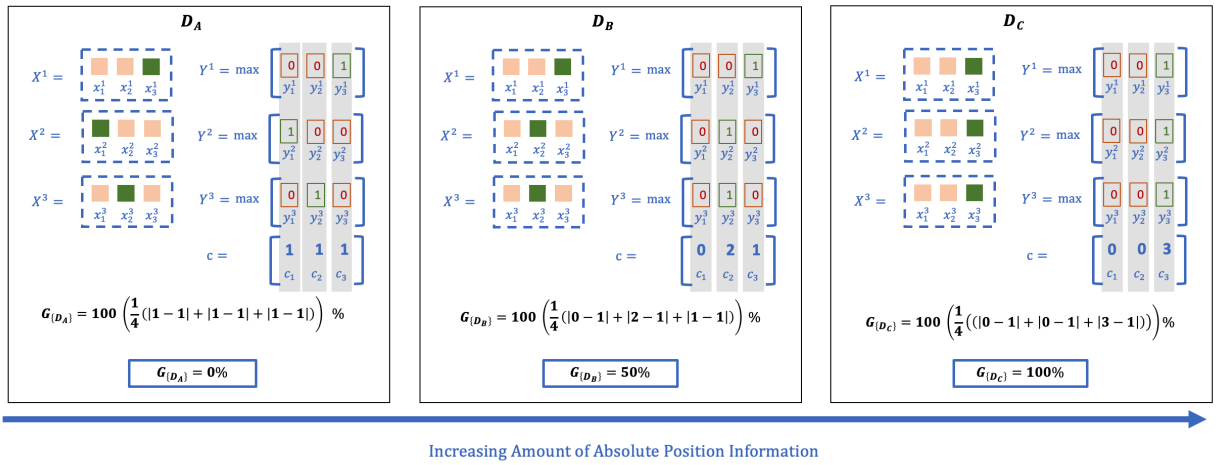

Figure 2: An illustrative example of toy datasets with varying percentages of absolute position information. Dataset D_A has 0% absolute position information. Dataset D_B has 50% absolute position information. Dataset D_C has 100% absolute position information. In these examples, $N = 3, n = 3, N^+ = 3$, and $N^+/n = 1$.

$y_j^{(i)} = 1$ are equally distributed among all positions

$$c_j = \frac{N^+}{n} \quad \forall j$$

Absolute position information is maximized when all instances are located in a single position $j^* \in \{1, n\}$

$$c_{j^*} = N^+$$
$$c_j = 0 \quad \forall j \neq j^*$$

Thus, we can define absolute position information using the average absolute value of the distances of c_j from N^+/n for all j, a quantity we call A_D

$$A_D = \frac{1}{n} \sum_{j=1}^{n} \left| c_j - \frac{N^+}{n} \right|$$

This is similar to the variance of c_j. We use the average absolute value of the distances instead of variance in our definition of absolute position information. Given that D_m is a dataset where all instances with the label $y_j^{(i)} = 1$ are equally distributed among m positions in the ordered lists in D_m, we want the difference in absolute position information between D_m and D_{m-1} to be the same regardless of m. The average absolute value of the distances weighs outliers (i.e., large differences between c_j and N^+/n) more similarly to small differences between c_j and N^+/n than the variance of distances. We discuss this further in Appendix Section A.

We convert A_D into a percentage so that absolute position information can be comparable across datasets with different N^+ and n values. To do this, we divide A_D by its maximum value $\max(A_D)$, which occurs when $c_{j^*} = N^+$ for a single position $j^* \in [1, n]$ and $c_j = 0 \quad \forall \quad j \neq j^*$

$$\begin{aligned} \max(A_D) &= \frac{1}{n}\left((n-1)\frac{N^+}{n} + \left(N^+ - \frac{N^+}{n}\right)\right) \\ &= 2\frac{N^+}{n^2}(n-1) \end{aligned}$$

and absolute position information as follows.

Definition: The percentage of absolute position information in dataset D, G_D, is defined as

$$G_D = \frac{A_D}{\max(A_D)} = \frac{n}{2(n-1)N^+} \sum_{j=1}^{n} \left| c_j - \frac{N^+}{n} \right| \%$$

An illustrative example of datasets with varying amounts of absolute position information is in Figure 2.

4.2. Proposed Wrapper

To enable standard approaches to leverage absolute position information, we developed a general purpose wrapper. This wrapper is summarized in Figure 3. We concatenate an absolute positional encoding to the output of the feature extractor within the MIL method pipeline and let th aggregator learn from both the output of the feature extractor and the positional encoding. Below, we justify the wrapper's use of an absolute positional encoding, concatenation, and late fusion.

Justification for Use of Positional Encodings. We use an absolute positional encoding in the wrapper, given its ability to learn absolute position information (Vaswani et al., 2017; Shaw et al., 2018). Specifically, we use one of the most popular absolute positional encodings, which encodes position via sine and cosine functions of different frequencies (Vaswani et al., 2017). We opted for a positional encoding over alternatives, given the ease with which it could be used in a wrapper. While alternatives like convolutional, capsule, and recurrent networks can model position information within inputs (Islam et al., 2020), adding these layers to permutation invariant aggregators would increase the amount of computation compared to using a positional encoding.

Justification for Use of Concatenation and Late Fusion. In our proposed wrapper, we concatenate the positional encodings to the output of the feature extractor. While transformers typically sum positional encodings with their inputs (Vaswani et al., 2017), we chose to concatenate the positional encodings to ensure that our models could separately leverage the features learned from the feature extractor and the position information provided by the positional encoding. While this concatenation doubles the computation required of the aggregator, we hypothesized that this would not significantly increase the overall classification time of any of our methods. We concatenated positional encodings to the output of the feature extractor instead of the input to the feature extractor because feature extractors are often pretrained on non-MIL data (like ImageNet) that do not depend on lists and therefore do not have positions (Deng et al., 2010; Lu et al., 2021; Shao et al., 2021; Zhang et al., 2022).

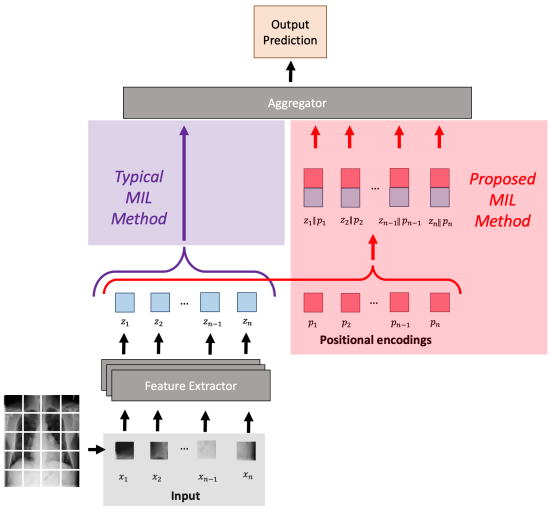

Figure 3: In comparison to existing non-transformer-based MIL methods that are permutation invariant, we concatenate a positional encoding to the output of the feature extractor within the MIL method pipeline and let the aggregator learn from the output of the feature extractor and the positional encoding.

5. Experimental Setup

In our experiments, we first evaluate multiple MIL methods (the non-transformer-based methods ABD-MIL (Ilse et al., 2018), CLAM-SB (Lu et al., 2021), CLAM-MB (Lu et al., 2021), SG-MIL (Seibold et al., 2021), and DTFD (Zhang et al., 2022); and the transformer-based methods TransMIL (Shao et al., 2021) and DAS-MIL (Wölflein et al., 2023)) in their ability to classify data with varying amounts of absolute position information. SG-MIL is the method developed specifically to classify chest x-rays. We then perform an ablation study with TransMIL to examine the mechanisms it uses to leverage absolute position information. Finally, we determine the extent to which our positional encoding wrapper can improve the performance of all non-transformer-based methods. The aforementioned approaches were selected because they are among the most widely used and most accurate MIL methods. We further describe these methods in Appendix Section B and our training and hyperparameter tuning procedure for these methods in Appendix Section C.

We evaluate all approaches on a suite of benchmark tasks in which the amount of absolute position information varies and two real-world tasks. While the percent of absolute position information in the real-world task is unknown, we expect this task to contain absolute position information due to past work that provides evidence that the position of instances of importance in this task are consistent across examples (Wang et al., 2017). We present details on the synthetic and real-data tasks in the following subsections.

5.1. Benchmark Synthetic Data Task: Classifying MNIST Lists

The goals of the experiments on the benchmark tasks are to evaluate the performance of MIL methods with and without the wrapper on tasks with varying amounts of absolute position information and training data. We vary the amounts of training data to identify the MIL methods that are most data efficient and to change the difficulty of the tasks, to identify how useful positional encodings are as task difficulty increases. Thus, we developed tasks in which we can change the amount of absolute position information and increase the amount of training and validation data.

We consider a variation of the MNIST Bag task that we call the MNIST List task. The MNIST Bag task is a task that has been used to evaluate MIL methods in past work (Ilse et al., 2018). In the MNIST Bag task, each instance $x_j^{(i)} \in [0, 255]^{1\times 28\times 28}$ in a bag is an image sampled from the MNIST dataset (Deng, 2012). A bag has the label $y_j^{(i)} = 1$ if at least one instance in the bag $x_j^{(i)}$ is an image of a '9' and a bag has the label $y_j^{(i)} = 0$ otherwise. The data in the MNIST Bag task have 0% absolute position information.

In the MNIST List task, we define the instances and list labels in the same way as the MNIST Bag task. Unlike in the MNIST Bag task, in the MNIST List task, we evaluate the ability of methods to classify datasets with different amounts of absolute position information: specifically, we vary absolute position information according to $100(w/(n-1))\%$ for $w = 0, \cdots, n-1$. Furthermore, only one instance in each list with the label $Y^{(i)} = 1$ has the label $y_j^{(i)} = 1$ and each list is of size $n = 10$. We describe the method that we use to create datasets for the MNIST List task in Appendix Section D. Using that method, we create 30 training sets (10 of size 100, 10 of size 1000, and 10 of size 10000), 30 validation sets (10 of size 100, 10 of size 1000, and 10 of size 10000), and 10 test sets (of size 250).

While we are focused on tasks with absolute position information, we also investigate the ability of the wrapper in the context of relative position information. To do this, we create a dataset with weak absolute position information but strong relative position information. Specifically, we modify the dataset for the MNIST List Task with 100 training bags and 22% absolute position information such that given a list with an image of a '9' is in position j, that list will have an image of an '8' in position $j-1$.

5.2. Real-World Tasks: Classifying Cardiomegaly and Pulmonary Edema in Chest X-Rays

We compare the performance of MIL methods on two real data tasks that do not have instance labels, and thus have unknown amounts of absolute position information. Based on domain knowledge, however, we hypothesize that one task (classifying cardiomegaly) has likely strong absolute position information, and the other task (classifying pulmonary edema) has likely weak absolute position information. This allows us to explore the benefit of the wrapper in real-world settings with different amounts of absolute position information. This also allows us to test performance gains from the wrapper on lists that potentially have instances labeled $y_j^{(i)} = 1$ for multiple j. In our synthetic data setting, only one instance per list corresponds to the label $y_j^{(i)} = 1$.

We evaluate the performance of several MIL methods in classifying cardiomegaly and pulmonary edema in chest x-rays using the MIMIC-CXR dataset, which is composed of 377,110 chest x-rays taken at the Beth Israel Deaconess Medical Center (Johnson et al., 2019). The chest x-rays range in size but are generally around size 3000 x 2000 pixels.

The term "cardiomegaly" refers to a condition in which one has an enlarged heart (Amin and Siddiqui, 2022). The heart is in a similar position across chest x-rays because chest x-rays are taken with patients in similar positions, although there can be slight variation (i.e. if the patients are at slightly different angles) (Wang et al., 2017). Thus, classifying cardiomegaly in chest x-rays is a problem with likely strong absolute position information.

The term "pulmonary edema" refers to a condition in which one has an abnormal amount of fluid in the lungs (Malek and Soufi, 2023). Pathological findings for pulmonary edema can occur in multiple locations in the lungs (Malek and Soufi, 2023). Thus, classifying pulmonary edema in chest x-rays is a problem with likely weak absolute position information.

For the task of classifying cardiomegaly and pulmonary edema in chest x-rays, $x_j^{(i)} \in [0, 255]^{3\times512\times512}$ is a patch sampled from a chest x-ray and $n = 24$ is the number of instances within a list . The chest x-rays are not annotated, so unlike our synthetic datasets, we do not know the underlying $y_j^{(i)}$ for any of our patches. We only have labels, at the chest x-ray or list level, assigned using the CheXpert labeler, a rule-based algorithm that extracts findings from radiology reports (Irvin et al., 2019). For classifying cardiomegaly, a chest x-ray is labeled $Y^{(i)} = 1$ if cardiomegaly is present in the corresponding radiology report and 0 otherwise. For classifying pulmonary edema, a chest x-ray is labeled $Y^{(i)} = 1$ if pulmonary edema is present in the corresponding radiology report and 0 otherwise. We describe the preprocessing, pretraining, and data split method that we use in Appendix Section E.

5.3. Evaluation Metrics

We evaluate the accuracy and computational efficiency of the aforementioned MIL methods. For medical image classification, high accuracy is necessary to ensure that appropriate diagnoses will be made and speed ensures that diagnoses can be given promptly.

We evaluate all methods' computational efficiency using wall clock time at evaluation (which corresponds to wall clock time at training). The wall clock time reported for the methods on the real data tasks is the wall clock time of the aggregator because the time needed to compute the features from the feature extractor is constant across methods.

We evaluate accuracy on the real and synthetic data tasks differently because some of our synthetic datasets have a small number of training and validation lists, leading to large performance differences among mode ls. For the synthetic data tasks, for each approach, we train 10 models on each dataset with 10 different random seeds respectively, and report the median and IQR of the AUROCs of each of the 10 models applied to the test sets. For the real data tasks, given the size of the training sets (N≈35,000), we train one model per approach and report the median and the 95% confidence interval of the AUROC of 1000 bootstrapped samples of our test set.

Whenever we compare two methods' performances and mention a statistically significant difference in performance, we measure that statistical significance via a bootstrap hypothesis test with a significance level of 0.05 (Efron and Tibshirani, 1993).

6. Results and Discussion

Through our experiments, we probe the following questions

- **Baseline Evaluation.** How do existing methods perform on tasks with absolute position information?
- **Transformer Ablation Study.** What mechanisms do transformers use to learn absolute position information?
- **Wrapper Effectiveness.**
  - Does a simple positional encoding wrapper-based approach lead to improved performance on tasks with absolute position information?
  - How does wrapper-based performance compare to transformer-based MIL methods?
- **Wrapper Robustness.** How does the performance gain from our positional encoding wrapper vary with the amount of absolute position information and the amount of training data in the task?
- **Other Types of Position Information.** Does our wrapper lead to improved performance on tasks with only relative position information?

6.1. Baseline Evaluation.

How do existing methods perform on tasks with absolute position information? On the real data task with likely strong absolute position information (classifying cardiomegaly in chest x-rays), TransMIL outperforms the transformer-based method DAS-MIL and all non-transformer-based methods. However, it is slower than these methods (Table 2, Appendix Table 4). DAS-MIL does not outperform non-transformer-based methods (Table 2). This is because DAS-MIL uses a relative positional encoding, not an absolute positional encoding, and thus is not able to leverage

the absolute position information in this task as well as TransMIL. ABDMIL outperforms all other non-transformer-based methods, including SG-MIL, both in terms of AUROC (0.782, 95% CI: [0.774, 0.789] vs. 0.755, 95% CI: [0.747, 0.764]) and speed (11.87 vs. 31.82 seconds).

On the synthetic data task, when trained on 100 lists, we find TransMIL only has a higher AUROC than non-transformer-based methods on data with $>$ 89% absolute position information. However, TransMIL's performance on tasks with $\leq$ 89% absolute position information improves as the number of training lists increases. Thus TransMIL especially struggles to classify tasks with small amounts of absolute position information when given small amounts of training data (Figure 4, Appendix Figure 5, Appendix Figure 6).

Thus, while TransMIL is slower than non-transformers and struggles to classify lists with weak absolute position information in limited data settings, it can exceed the AUROC of non-transformers on tasks with absolute position information.

6.2. Transformer Ablation Study.

What mechanisms do transformers use to learn absolute position information? Given that DAS-MIL cannot leverage absolute position information as well as TransMIL, in this section, we examine TransMIL's components in their ability to learn absolute position information. When we replace TransMIL's attention mechanism (the Nystrom attention approximation) with the original attention mechanism, the resulting transformer cannot leverage absolute position information without positional encodings. On the real data task with likely strong absolute position information, the performance of that transformer without positional encodings is significantly worse than with positional encodings (Table 1). We see similar results on the synthetic data tasks with absolute position information (Appendix Table 6). Thus, the Nystrom attention approximation can leverage absolute position information. This finding contradicts past work which has assumed that transformers are position invariant without positional encodings (Jung et al., 2020).

Table 1: 95% CI of AUROC of TransMIL-Based Methods on the real data task with likely strong absolute position information. The * signifies a statistically significant difference in performance between transformers with and without positional encodings (PE) when they use the original attention mechanism.

Attention Type	Test AUROC without PE	Test AUROC with PE
Nystrom	0.800 (0.793, 0.806)	0.797 (0.790, 0.804)
Original	0.785 (0.778, 0.791)*	0.805 (0.798, 0.812)

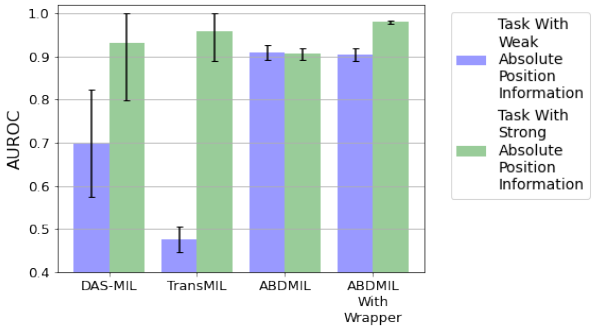

Figure 4: Median and IQR of the AUROC of transformer-based (DAS-MIL, TransMIL) and non-transformer-based (ABDMIL, ABDMIL+Wrapper) MIL methods on the synthetic data tasks with 0% (weak) and 100% (strong) absolute position information.

6.3. Wrapper Effectiveness.

Does a simple positional encoding wrapper-based approach lead to improved performance on tasks with absolute position information? On the synthetic data task, when trained on 100 lists, all non-transformer-based methods benefit from the wrapper when classifying data with $\geq$ 67% absolute position information. Specifically, when classifying data with 67% absolute position information, ABDMIL with the wrapper performs significantly better than ABDMIL without the wrapper (0.955, IQR: [0.948, 0.961] vs. 0.919, IQR: [0.902, 0.937]). This trend continues for ABDMIL and other non-transformer-based methods on synthetic datasets with $\geq$ 67% absolute position information (Appendix Figure 5). We see similar results on the real data task with likely strong absolute position information (classifying cardiomegaly in chest x-rays), for ABDMIL and other non-transformer-based methods, (Table 2, Appendix Table 4).

One potential concern is that our wrapper would significantly increase MIL methods' classification

Table 2: 95% CI of AUROC and speed of transformer-based (DAS-MIL, TransMIL) and non-transformer-based (ABDMIL, ABDMIL+Wrapper) MIL methods on the real data task with likely strong absolute position information.

Model	Test AUROC	Time (s)
DAS-MIL	0.779 (0.771, 0.787)	55.39
TransMIL	0.797 (0.790, 0.804)	122.60
ABDMIL	0.782 (0.774, 0.789)	11.87
ABDMIL+Wrapper	0.799 (0.791, 0.806)	13.17

time. We find that all methods experience no change or a marginal (10%) increase in classification time when using the wrapper versus not (Table 2, Appendix Table 4, Appendix Table 5). Thus, the wrapper does not significantly increase the speed of any method. Furthermore, in terms of computational efficiency, our wrapper is on par with TransMIL's CNN-based positional encoding.

How does wrapper-based performance compare to transformer-based MIL methods All non-transformers with the wrapper except for SG-MIL achieve higher AUROC than DAS-MIL. We see this when comparing ABDMIL and DAS-MIL on the synthetic data task with 100 training ordered lists and strong absolute position information (0.980, IQR: [0.976, 0.984] vs. 0.930, IQR: [0.797, 1.063]) and the real data task with likely strong absolute position information (Figure 4, Table 2).

Furthermore, the AUROC of all non-transformers with the wrapper except for SG-MIL is on par with that of TransMIL. We see this when comparing ABDMIL and TransMIL on the synthetic data task with 100 training ordered lists and strong absolute position information (0.980, IQR: [0.976, 0.984] vs. 0.957, IQR: [0.890, 1.024]) and the real data task with likely strong absolute position information (Figure 4, Table 2).

SG-MIL's performance with the wrapper is slightly lower than that of the transformers on the real data task with likely strong absolute position information (0.766, 95% CI: [0.758, 0.773] vs. 0.779, 95% CI: [0.771, 0.787] and 0.797, 95% CI: [0.790, 0.804]) (Table 2. Appendix Table 4). However, similar to their performance without the wrapper, all non-transformers with the wrapper, including SG-MIL, are faster than the transformer and outperform transformers on tasks with weak absolute position information (Figure 4). Thus, on all tasks, one can use a non-transformer to perform better than or comparably to transformers while being faster.

Table 3: 95% CI of AUROC and speed of MIL methods on the real data task with likely weak absolute position information.

Model	Test AUROC	Time (s)
DAS-MIL	0.855 (0.850, 0.860)	51.31
TransMIL	0.861 (0.856, 0.866)	96.23
ABDMIL	0.857 (0.851, 0.862)	13.85
ABDMIL+Wrapper	0.860 (0.855, 0.865)	10.27

6.4. Wrapper Robustness.

How does the performance gain from our positional encoding wrapper vary with the amount of absolute position information and the amount of training data in the task? While the wrapper improves performance on tasks with absolute position information, it does not hurt performance on tasks with 0% absolute position information. On the synthetic data task with 0% absolute position information, when trained on 100 lists, ABDMIL performs the same with and without the wrapper (0.904, IQR: [0.890, 0.918] vs. 0.909, IQR: [0.891, 0.927]) (Figure 4). This trend holds for all other non-transformer-based methods when trained on the same or more lists (Appendix Figure 5). Similarly, on the real data task with likely weak absolute position information (classifying pulmonary edema in chest x-rays), ABDMIL performs the same with and without the wrapper and this trend holds for all other non-transformer-based methods (Table 3, Appendix Table 5). It is important to note that on these tasks, with weak to no position information, standard methods without the wrapper achieve comparable performance to transformers (Table 3).

On the synthetic data tasks, while the wrapper improves performance significantly on methods trained on 100 lists, it improves performance less on methods trained on 1,000 lists and does not impact the performance of methods trained on 10,000 lists. Thus, the wrapper is most helpful in limited data settings (Appendix Figure 5). As dataset size increases, task difficulty decreases, evidenced by the performance of all methods without positional encodings increasing. This leaves little room for performance improvements via positional encodings.

6.5. Other Types of Position Information.

Does our wrapper lead to improved performance on tasks with only relative position information? On a synthetic data task with weak absolute position but strong relative position, the wrapper is unable to improve the performance of ABDMIL (0.897, IQR: [0.883, 0.911] without the wrapper vs. 0.906, IQR: [0.882, 0.930] with) (Appendix Table 7). This is because our wrapper does not encode relative position information. Past work has developed relative positional encodings for transformers (Shaw et al., 2018; Wölflein et al., 2023). We hypothesize that augmenting standard approaches like ABDMIL with these encodings could improve their performance on tasks with only relative position information.

7. Conclusion

In this work, we formalize the MIL problem with absolute position information, identify mechanisms that allow transformers to learn position information, and based on that, propose a wrapper that augments non-transformer-based MIL methods with positional encodings. Our main findings are that 1) despite being slower and less data efficient, current transformers outperform current non-transformers on tasks with absolute position information; 2) these transformers can leverage absolute position information via the Nystrom attention approximation and positional encodings; and 3) a simple positional encoding wrapper can significantly improve the performance of non-transformer-based methods. The last finding is perhaps the most significant since it suggests we do not need to rely solely on computationally and data inefficient transformer-based MIL approaches to solve tasks with absolute position information. Our findings held across several synthetic data tasks and two real data tasks.

Our work is not without limitations. We do not evaluate the effect of the wrapper on methods that explicitly pretrain their feature extractor in a manner that deviates from our pretraining strategy, like the standard MIL approaches DSMIL and MIL-RNN; and the transformer-based MIL approach SETMIL (Campanella et al., 2019; Li et al., 2021; Zhao et al., 2022). However, none of the pretraining strategies and the aggregators of corresponding methods in past work incorporated the position of instances. Thus, we hypothesize that the wrapper should improve the performance of the non-transformer-based MIL methods with feature extractors pretrained in ways that differ from our pretraining strategy. Nevertheless, we leave the exploration of the impact of pretraining strategies on the performance of MIL methods on tasks with absolute position information as future work. Another potential future direction is to develop a wrapper that could improve the performance of standard approaches on tasks with weak absolute position information but strong relative position information.

Overall, this work formalizes the MIL problem with absolute position information and demonstrates the utility of a simple positional encoding wrapper in MIL classification on tasks with absolute position information. Our results contribute to the growing body of work that finds that attention may not be all you need (Poli et al., 2023), and non-transformer-based methods can achieve comparable performance with greater computational efficiency than transformers on tasks that traditionally use transformers.

8. Acknowledgements

This work was supported by the D. Dan & Betty Kahn Foundation through a grant to U-M. M.K. received funding from the Graduate Fellowship for STEM Diversity. The views and conclusions in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of any funding sources. We also thank the anonymous reviewers for their valuable feedback.

References

Hina Amin and Waqas J. Siddiqui. Cardiomegaly, 2022. URL https://www.ncbi.nlm.nih.gov/books/NBK542296/.

Jimmy Lei Ba and Diederik P. Kingma. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015.

Gabriele Campanella, Matthew G. Hanna, Luke Geneslaw, Allen Miraflor, Vitor Werneck Krauss Silva, Klaus J. Busam, Edi Brogi, Victor E. Reuter, David S. Klimstra, and Thomas J. Fuchs. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. *Nature Medicine*, 25, 2019. ISSN 1546170X. doi: 10.1038/s41591-019-0508-1.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. doi: 10.1109/cvpr.2009.5206848.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Number 57 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, Boca Raton, Florida, USA, 1993.

Zhe Huang, Benjamin S. Wessler, and Michael C. Hughes. Detecting heart disease from multi-view ultrasound images via supervised attention multiple instance learning, 2024.

Maximilian Ilse, Jakub M. Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *35th International Conference on Machine Learning*, volume 5, 2018.

Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *33rd AAAI Conference on Artificial Intelligence*, 2019. doi: 10.1609/aaai.v33i01.3301590.

Md Amirul Islam, Sen Jia, and Neil D.B. Bruce. How much position information do convolutional neural networks encode? In *8th International Conference on Learning Representations*, 2020.

Johns Hopkins Medicine. Brain tumors and brain cancer. URL https://www.hopkinsmedicine.org/health/conditions-and-diseases/brain-tumor.

Alistair E.W. Johnson, Tom J. Pollard, Seth J. Berkowitz, Nathaniel R. Greenbaum, Matthew P. Lungren, Chih ying Deng, Roger G. Mark, and Steven Horng. Mimic-cxr, a de-identified publicly available database of chest radiographs with free-text reports. *Scientific Data*, 6, 2019. ISSN 20524463. doi: 10.1038/s41597-019-0322-0.

Yunjae Jung, Donghyeon Cho, Sanghyun Woo, and In So Kweon. Global-and-local relative position embedding for unsupervised video summarization. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 12370 LNCS, 2020. doi: 10.1007/978-3-030-58595-2_11.

Bin Li, Yin Li, and Kevin W. Eliceiri. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2021. doi: 10.1109/CVPR46437.2021.01409.

Ming Y. Lu, Drew F.K. Williamson, Tiffany Y. Chen, Richard J. Chen, Matteo Barbieri, and Faisal Mahmood. Data-efficient and weakly supervised computational pathology on whole-slide images. *Nature Biomedical Engineering*, 5, 2021. ISSN 2157846X. doi: 10.1038/s41551-020-00682-w.

Ryan Malek and Shadi Soufi. Pulmonary edema, 2023. URL https://www.ncbi.nlm.nih.gov/books/NBK557611/.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y. Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. In *Proceedings of the 40 th International Conference on Machine Learning*, 2023.

Constantin Seibold, Jens Kleesiek, Heinz Peter Schlemmer, and Rainer Stiefelhagen. Self-guided multiple instance learning for weakly supervised disease classification and localization in chest radiographs. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 12626 LNCS, 2021. doi: 10.1007/978-3-030-69541-5_37.

Zhuchen Shao, Hao Bian, Yang Chen, Yifeng Wang, Jian Zhang, Xiangyang Ji, and Yongbing Zhang. Transmil: Transformer based correlated multiple instance learning for whole slide image classification. In *Advances in Neural Information Processing Systems*, volume 3, 2021.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representa-

tions. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 2, 2018. doi: 10.18653/v1/n18-2074.

Gijs van Tulder, Yao Tong, and Elena Marchiori. Multi-view analysis of unregistered medical images using cross-view transformers. *CoRR*, abs/2103.11390, 2021. URL https://arxiv.org/abs/2103.11390.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 2017-December, 2017.

Arsh Verma and Makarand Tapaswi. Can we adopt self-supervised pretraining for chest x-rays? In *ML4H Extended Abstract Collection*, 2022. doi: https://arxiv.org/pdf/2211.12931.pdf.

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *30th IEEE Conference on Computer Vision and Pattern Recognition*, volume 2017-January, 2017. doi: 10.1109/CVPR.2017.369.

Georg Wölflein, Lucie Charlotte Magister, Pietro Liò, David J Harrison, and Ognjen Arandjelović. Deep multiple instance learning with distance-aware self-attention. In *arXiv Preprint*, 2023. doi: https://arxiv.org/abs/2305.10552.

Hongrun Zhang, Yanda Meng, Yitian Zhao, Yihong Qiao, Xiaoyun Yang, Sarah E. Coupland, and Yalin Zheng. Dtfd-mil: Double-tier feature distillation multiple instance learning for histopathology whole slide image classification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2022-June, 2022. doi: 10.1109/CVPR52688.2022.01824.

Yu Zhao, Zhenyu Lin, Kai Sun, Yidan Zhang, Junzhou Huang, Liansheng Wang, and Jianhua Yao. Setmil: Spatial encoding transformer-based multiple instance learning for pathological image analysis. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 13432 LNCS, 2022. doi: 10.1007/978-3-031-16434-7_7.

Appendix A. Comparing Variance and Average Absolute Distance

Consider two datasets, D_m and D_{m-1}, each with N ordered lists of size n. In D_{m-1}, all instances with the label $y_j^{(i)} = 1$ are equally distributed among $m-1$ positions in the N ordered lists. In D_m, all instances with the label $y_j^{(i)} = 1$ are equally distributed among m positions in the N ordered list.

The difference in the variance of c_j in D_{m-1} and D_m ($V_{D_{m-1}} - V_{D_m}$) depends on m

$$V_{D_{m-1}} - V_{D_m} = \frac{{N^+}^2}{n(m-1)m}$$

The difference in the average absolute value of the distances of c_j from N^+/n in D_{m-1} and D_m ($A_{D_{m-1}} - A_{D_m}$) is independent of m.

$$A_{D_{m-1}} - A_{D_m} = \frac{2N^+}{n^2}$$

We define the difference in absolute position information between 2 datasets with the same N and n independently of m (the number of positions the instances with the label $y_j^{(i)}$ span). Thus we use A_D in our definition of absolute position information instead of V_D.

Appendix B. Defining MIL Methods

In our paper, we evaluate the ability of the following methods with and without positional encodings to solve tasks with absolute position information. These approaches were selected because they are either among the most widely used MIL methods or they evaluated their MIL methods on similar tasks to the tasks in this paper.

- Attention based deep MIL (ABDMIL) (Ilse et al., 2018): Unlike early MIL methods that used the maximum and mean operators to aggregate the features from each instance, ABDMIL proposed a more complex permutation invariant aggregator: a trainable weighted average where the weights are computed with a 2-layer neural network.
- Multiple Instance Learning with Self-Guided Loss (SG-MIL) (Seibold et al., 2021): SG-MIL aggregates features from each instance using the softmax average pooling operator, and iterates on this representation using a standard cross-entropy loss function as well as a loss function that assigns pseudo-labels to instances.
- Single-attention-branch CLAM (CLAM-SB) and multi-attention-branch CLAM (CLAM-MB) (Lu et al., 2021) build upon ABDMIL by supervising it via an additional task that assigned pseudo labels to instances.
- Double tier feature distillation (DTFD) (Zhang et al., 2022) builds upon ABDMIL by breaking up large bags into smaller pseudo-bags to better learn from bags with a small number of informative instances.
- TransMIL (Shao et al., 2021): TransMIL uses the Nystrom attention approximation of the attention mechanism in the aggregation step instead of the computationally inefficient original attention mechanism and contains a specialized positional encoding that can learn absolute position information.
- Distance-aware self-attention (DAS-MIL) (Wölflein et al., 2023) models the relative spatial information among inputs in the self-attention mechanism of a transformer instead of using a positional encoding.

Appendix C. Hyperparameters and Training

The feature extractor used for all models trained on synthetic data has a similar architecture to the feature extractor used in (Ilse et al., 2018). It has 2 convolutional layers followed by three linear layers. Both convolutional layers have a kernel size of 5. The first convolutional layer has 40 output channels. The second convolutional layer has 100 output channels. The linear layers all have 500 output features. The feature extractor used by all models trained on the real data uses a Densenet-121 architecture with a learning rate of $1e-5$ and a weight decay of $1e-6$ (Irvin et al., 2019).

The final hyperparameters and the hyperparameters searched over for all MIL methods on all datasets available on Github. The hyperparameters not specified on Github are the hyperparameters reported in the papers corresponding to CLAM-SB, CLAM-MB, and TransMIL (Lu et al., 2021; Shao et al., 2021).

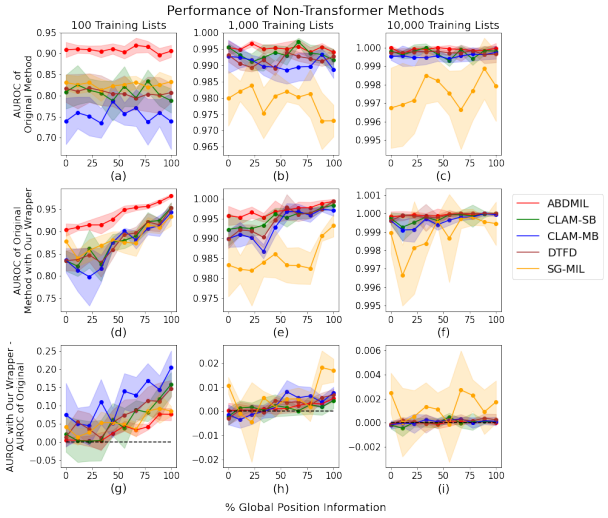

Figure 5: All subplots show how the median and IQR of a measure of performance of various non-transformer-based MIL methods (on the y-axis) varies with % absolute position information in each test set (on the x-axis). The measure of performance is AUROC of each original method for subfigures a-c, AUROC of each method augmented with the wrapper for subfigures d-f, and the change in AUROC of each method augmented with compared to the method without the wrapper for subfigures g-i.

We implement and train our models using Pytorch version 1.13.1, CUDA version 11.6, Ubuntu 20.04.5, 8 NVIDIA RTX A6000 GPUs, an Adam optimizer (Ba and Kingma, 2015), and a batch size of 1 as is typical for multiple instance learning. We train for at least 10 iterations, and then until validation performance does not improve for 5 iterations, selecting the model for which validation performance was best. We train for a maximum of 500 epochs.

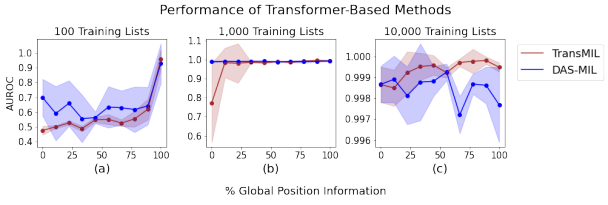

Figure 6: All subplots show how the median and IQR of the AUROC of transformer-based MIL methods (on the y-axis) varies with % absolute position information in each test set (on the x-axis)

Appendix D. MNIST Ordered List Task Data Creation

To create datasets for the MNIST Ordered List task, we first create datasets with ordered lists with 100% absolute position information. Half of the ordered lists in these datasets have the label $Y^{(i)} = 1$ and the other half of ordered lists have the label $Y^{(i)} = 0$. Each ordered list with the label $Y^{(i)} = 0$ contains n images randomly sampled from the set of images in the MNIST dataset that do not contain a '9'. Each ordered list with the label $Y^{(i)} = 1$ is created such that the first $n-1$ images are randomly sampled from the set of images in the MNIST dataset that do not contain a '9', and the $n-$th image is randomly sampled from the set of images in the MNIST dataset that do contain a '9'. Then, to create a dataset with a variable amount of absolute position information, we shuffle the positions of instances in each ordered list in the dataset $D = \{X^{(i)}, Y^{(i)}\}_{i=1}^{N}$ with 100% absolute position information such that the dataset's percentage of absolute position information becomes $100(w/(n-1))\%$ for $w = [0, \ldots, n-2]$

Algorithm 1: Shuffling Instance Positions

for $i \in [1, \ldots, N]$ **do**
  $Z = X^{(i)}$
  $v = \texttt{RandomInt}(0, n - w - 1)$
  for $j \in [1, \ldots, n]$ **do**
    if $j > v$ **then**
      $z_j = x^{(i)}_{j-v}$
    end
    else
      $z_j = x^{(i)}_{n-(v-j)}$
    end
  end
  $X^{(i)} = Z$;
end

Table 4: 95% CI of AUROC and Classification Time of MIL methods with and without the wrapper on the real data task with likely strong absolute position information (classifying cardiomegaly in chest x-rays), (classifying Cardiomegaly in chest x-rays).

Base Model	Use Wrapper?	Test AUROC	Time (s)
CLAM-SB	No	0.781 (0.774, 0.788)	13.58
	Yes	0.799 (0.792, 0.806)	15.41
CLAM-MB	No	0.778 (0.771, 0.786)	12.49
	Yes	0.798 (0.790, 0.805)	14.32
DTFD	No	0.777 (0.769, 0.785)	48.3
	Yes	0.792 (0.785, 0.799)	46.3
SG-MIL	No	0.755 (0.747, 0.764)	31.8
	Yes	0.766 (0.758, 0.773)	32.2

Table 5: 95% CI of AUROC and Classification Time of MIL methods with and without the wrapper on the real data task with likely weak absolute position information (classifying Pulmonary Edema in chest x-rays).

Base Model	Use Wrapper?	Test AUROC	Time (s)
CLAM-SB	No	0.859 (0.853, 0.864)	10.11
	Yes	0.860 (0.856, 0.866)	10.93
CLAM-MB	No	0.859 (0.854, 0.864)	10.33
	Yes	0.859 (0.854, 0.865)	10.87
DTFD	No	0.856 (0.852, 0.862)	38.63
	Yes	0.857 (0.852, 0.862)	33.96
SG-MIL	No	0.840 (0.835, 0.846)	28.05
	Yes	0.836 (0.830, 0.841)	29.17

Appendix E. Preprocessing and Data Split for Real Data Task

Preprocesssing and Pretraining. To create ordered lists, we resize each chest x-ray to be of size 3000 x 2000 (the average size of most chest x-rays) and then extract 24 non-overlapping patches from the chest x-ray to serve as instances in the ordered list. We exclude chest x-rays labeled uncertain by the CheXpert labeler because past work did not find a statistically significant advantage to including them (Irvin et al., 2019).

To give the best chance to MIL methods, we pretrain the feature extractor (described in Appendix Section C) of all MIL methods for 3 epochs on data from the CheXpert dataset, a related dataset of 224,316 chest x-rays taken at the Stanford Hospital (Verma and Tapaswi, 2022; Irvin et al., 2019). We pretrain the feature extractor in a supervised manner on patches extracted from the chest x-rays. We assigned all the patches the image label and trained on the patches given this label. For the task of classifying cardiomegaly, our pretraining set contains 10,249 images (7,617 with cardiomegaly and 2,632 without). For the task of classifying pulmonary edema, our pretraining set contains 21,747 images (16,403 with pulmonary edema and 5,344 without).

Training, Validation, and Test Sets. After pretraining, we fix the feature extractor of all methods and apply each MIL method to data from the MIMIC-CXR Database (Johnson et al., 2019). We assigned each patient in the MIMIC-CXR database to belong to either the training, validation, or test split. Specifically, for the task of classifying cardiomegaly, our training set contains 35,117 images (26,955 with cardiomegaly and 8,162 without), our validation set contains 14,106 images (10,923 with cardiomegaly and 3,183 without), and our test set contains 21,239 images (16,212 with cardiomegaly and 5,027 without). Specifically, for the task of classifying pulmonary edema, our training set contains 34,436 images (21,484 with pulmonary edema and 12,952 without), our validation set contains 14,272 images (9,039 with pulmonary edema and 5,233 without), and our test set contains 14,195 images (13,050 with pulmonary edema and 7,809 without).

Table 6: Median and IQR of AUROC of Transformer-Based methods on the synthetic data task. This task contains 1,000 training ordered lists with 100% absolute position information. The * identifies whether there is no overlap in IQRs of the performance of methods with and without positional encodings (PE).

Attention Type	Test AUROC without PE	Test AUROC with PE
Nystrom	0.995 (0.993, 0.997)	0.997 (0.995, 0.999)
Original	0.992 (0.989, 0.994)	0.999 (0.995, 1.00)*

Table 7: IQR of AUROC of ABDMIL on the synthetic data task with only relative position information. This task contains 100 training ordered lists with 22% absolute position information but strong relative position information.

Method	Test AUROC
ABDMIL	0.897 (0.883, 0.911)
ABDMIL+Wrapper	0.906 (0.882, 0.930)

Appendix F. Run Time and Performance for All Methods

The classification times of all methods on the synthetic data task are available on Github. Appendix Figure 5 shows the AUROC of all non-transformer-based methods not in the main text on the synthetic data. Appendix Figure 6 shows the AUROC of the transformer-based methods not in the main text on the synthetic data. Appendix Table 4 lists the classification time and performance of all methods not in the main text on the real data tasks with absolute position information. Appendix Table 5 lists the classification time and performance of all methods not in the main text on the real data tasks without absolute position information. Appendix Table 6 lists the performance of TransMIL with and without its positional encodings on synthetic data tasks.