How Effective Can Dropout Be in Multiple Instance Learning ?

Wenhui Zhu [*1] **Peijie Qiu** [*2] **Xiwen Chen** [*3] **Zhangsihao Yang** [1] **Aristeidis Sotiras** [2] **Abolfazl Razi** [3] **Yalin Wang** [1]

Abstract

Multiple Instance Learning (MIL) is a popular weakly-supervised method for various applications, with a particular interest in histological whole slide image (WSI) classification. Due to the gigapixel resolution of WSI, applications of MIL in WSI typically necessitate a two-stage training scheme: first, extract features from the pre-trained backbone and then perform MIL aggregation. However, it is well-known that this suboptimal training scheme suffers from "noisy" feature embeddings from the backbone and inherent weak supervision, hindering MIL from learning rich and generalizable features. However, the most commonly used technique (i.e., dropout) for mitigating this issue has yet to be explored in MIL. In this paper, we empirically explore how effective the dropout can be in MIL. Interestingly, we observe that dropping the top-k most important instances within a bag leads to better performance and generalization even under noise attack. Based on this key observation, we propose a novel MIL-specific dropout method, termed MIL-Dropout, which systematically determines which instances to drop. Experiments on five MIL benchmark datasets and two WSI datasets demonstrate that MIL-Dropout boosts the performance of current MIL methods with a negligible computational cost. The code is available at https://github.com/ChongQingNoSubway/MILDropout.

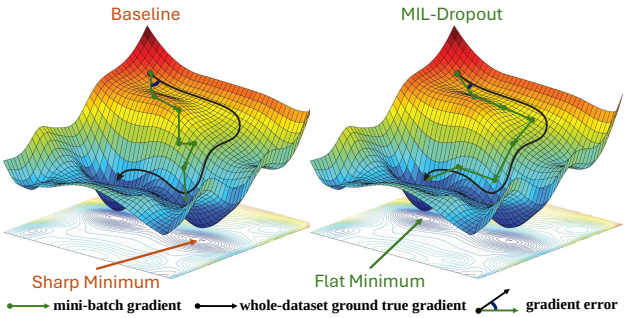

Figure 1. An illustrative example comparing the convergence trajectories of the baseline ABMIL without dropout (**Left**) and ABMIL with the proposed MIL-Dropout (**Right**). ABMIL without dropout is likely to follow an incorrect gradient direction initially and eventually converge to a sharp minimum. In contrast, ABMIL with the proposed MIL-Dropout typically achieves a lower gradient direction error and reaches a flatter minimum with a better generalization.

1. Introduction

Multiple instance learning (MIL) has gained significant attention in machine learning applications. MIL assigns a single label (bag-level label) to a collection of instances (a bag), making it a weakly-supervised classification method due to the absence of instance-level labels. MIL has been applied to various fields (Zhu et al., 2023b; Chen et al., 2024b), including digital pathology (Ilse et al., 2018; Li et al., 2021; Shao et al., 2021), video analysis (Babenko et al., 2010; Quellec et al., 2017), and time series classification (Early et al., 2023; Chen et al., 2024b). In particular, MIL is a defacto standard for histological whole slide image (WSI) classification in digital pathology. This is because WSIs are gigapixel images, making them challenging to process using traditional machine learning or deep learning methods. A WSI is treated as a bag comprising a collection of smaller tiles, each regarded as an instance. The MIL model takes a collection of WSI tiles as input and then assigns a single label to an entire WSI (e.g., malignant). Due to a large number of instances within a bag, deep learning based MIL methods (Wang et al., 2018b; Ilse et al., 2018; Shao et al., 2021; Zhang et al., 2022; Zhu et al., 2024; Qiu et al., 2023) typically necessitate a two-stage learning scheme, where features are first extracted by a pre-trained backbone and then combined by an MIL aggregator. However, this suboptimal training scheme suffers from "noisy" feature embeddings, typically resulting in reduced performance.

[*]Equal contribution [1]Arizona State University, USA. [2]Washington University in St. Louis, USA. [3]Clemson University, USA.. Correspondence to: Wenhui Zhu <wzhu59@asu.edu>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

Follow-up methods also argue that this issue is attributed to overfitting and the inability of the MIL aggregator to learn a rich representation in such a suboptimal learning scheme (Li et al., 2021). Although it is not formally discussed, (Li et al., 2021) randomly drops input instances (i.e., DropInstance in this paper) in their implementation, which partially evidences the effectiveness of this simple dropout-like method in regularizing MIL. Motivated by this observation, we argue that a formal and thorough investigation of the role of the dropout method in MIL is necessary.

For this purpose, we first conduct experiments on various dropout strategies in MIL. Our empirical investigation indicates that dropping the top-k most important instances (i.e., top-k DropInstance) leads to an increase in classification accuracy. At first glance, this appears counterintuitive, whereas an in-depth analysis reveals its theoretically intriguing properties. First, Different from findings in other deep learning models (Liu et al., 2023; Jastrzebski et al., 2020; Kavis et al., 2022), dropping top-k most important instances can help MIL reduce the gradient direction error until converging (see example in Fig. 1 and evidence in Fig. 4). Second, it encourages the MIL model to converge to a flatter minimum, which enhances generalizability (e.g., robustness to noise attack). Despite its simplicity, dropping important instances not only results in performance gain but also shows theoretical guarantees. However, determining the importance of instances and selecting which ones to drop remains an open question.

Leveraging the advantageous properties of top-k DropInstance, we propose a novel MIL-Dropout method that systematically determines the importance of instances and identifies which ones to drop. This approach helps reduce gradient direction errors and achieves better generalization by converging to flatter local minima (see example in Fig 1 (**Right**) and evidence in Fig. 3). Our method involves two key components: (i) a novel averaging-based attention mechanism to efficiently determine instance importance and (ii) a query-based instance selection mechanism to select the instance set to drop. Extensive experiments on various MIL benchmarks demonstrate that our method can be seamlessly integrated into existing MIL frameworks to effectively improve their performance with negligible computational cost.

2. Related Work

MIL Methods. The introduction of MI-Net (Wang et al., 2018b) has significantly elevated the prominence of bag-level MIL methods, which utilize only bag-level labels for supervision. This approach addresses the ambiguity associated with propagating bag-level labels to individual instances inherent in instance-level MIL methods (Feng & Zhou, 2017; Hou et al., 2016; Xu et al., 2019). Particularly in WSI classification, empirical studies have consistently demonstrated that bag-level MIL methods generally outperform their instance-level counterparts (Shao et al., 2021; Wang et al., 2018b). Recent advancements in bag-level MIL methods have primarily focused on improving instance-level MIL aggregation. This has been achieved through the incorporation of attention mechanisms (Ilse et al., 2018), transformers (Shao et al., 2021; Xiang & Zhang, 2023), pseudo bag (Zhang et al., 2022), and non-local attention (Li et al., 2021) to effectively capture correlations between instances. In addition, some studies have begun focusing on negative mining within instances, using masked approaches to uncover more positive instances and incorporating a complex teacher-student model (Tang et al., 2023). However, the rationale behind masking instances remains unclear, and these methods cannot be easily generalized to a wider range of MIL tasks. In contrast, we focus on demonstrating the effectiveness of dropout in regularizing MIL to achieve better performance and generalization. Instead of enhancing MIL aggregation, we focus on integrating the proposed a general MIL-Dropout into various MIL aggregators to boost their performance.

Dropout methods. Dropout has been empirically shown to be an effective technique for mitigating overfitting in a variety of computer vision studies (Hinton et al., 2012; Srivastava et al., 2014). Numerous follow-up studies have extended the concept of dropout, exploring a wide range of methodologies. Notable examples include spatial-dropout (Tompson et al., 2015), dropout based on contiguous regions (Ghiasi et al., 2018), and attention-based dropout (Choe & Shim, 2019). However, the application of dropout in MIL has not been explored. In the context of MIL, where the MIL aggregator operates on instance embedding features devoid of spatial context, the dropouts based on spatial context are not applicable. Despite sharing similarities with the attention-based dropout, our MIL-Dropout is tailored to MIL applications, which drops instances instead of neurons in other applications (e.g., natural image classification). It is worth noting that this paper also serves as a theoretical supplement to PDL (Zhu et al., 2023a).

3. Preliminary.

The MIL is usually treated as a binary classification problem, the goal is to learn a direct mapping from a bag of N instances $\boldsymbol{X} = \{\boldsymbol{x}_n \mid n = 1, \cdots, N\}$ to a binary label $Y \in \{0, 1\}$. In most real scenarios, the instance-level labels $\{y_n \mid n = 1, \cdots, N\}$ are unknown, making it a weakly-supervised problem:

$$Y = \begin{cases} 0, \text{ iff } \sum_n y_n = 0, \\ 1, \text{ otherwise.} \end{cases} \tag{1}$$

We considered embedding-based MIL as an example due

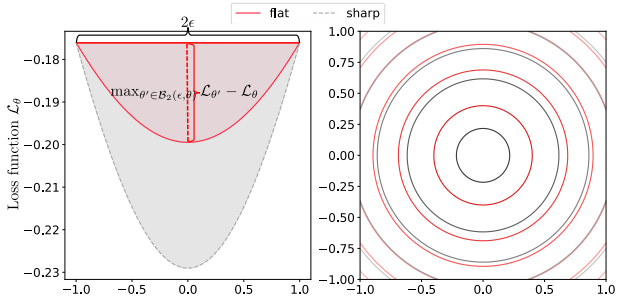

Figure 2. An conceptual illustration for a flat and sharp minimum in 1D curvature (**Left**) and 2D landscape (**Right**) of the loss function $\mathcal{L}_\theta$.

to its popularity. The standard workflow of an embedding-based MIL involves (i) projecting instances into feature embeddings via an instance-level feature extractor and (ii) aggregating the instance-level features into a bag-level prediction through an MIL aggregator. Specifically, the instance-level feature extractor typically consists of a pretrained backbone feature extractor $f_{\texttt{backbone}}$ (e.g., ResNet) and a trainable shallow feature extractor f_θ parameterized by θ (e.g., multi-layer perception), due to the intractability to optimize the backbone (Li et al., 2021). The input instances $\boldsymbol{X}$ is projected into D-dimensional feature vectors $\boldsymbol{V} = \{\boldsymbol{v}_n \mid n = 1, \cdots, N\} \in \mathbb{R}^{N \times D}$ by applying the instance-level feature extractors: $\boldsymbol{V} = f_\theta(f_{\texttt{backbone}}(\boldsymbol{X}))$. The MIL aggregator is then applied to the instance-level feature embeddings to obtain the bag-level probabilistic prediction $\hat{\boldsymbol{Y}} \in [0, 1]$:

$$\hat{\boldsymbol{Y}} = f_\omega(\rho_\psi(\{\boldsymbol{v}_n \mid n = 1, \cdots, N\})), \tag{2}$$

where $f_\omega(\cdot)$ is a bag-level classifier parameterized by ω, and ρ_ψ is a permutation-invariant MIL pooling function parameterized by ψ. The learning of an MIL can be achieved by optimizing the binary cross-entropy loss over trainable parameters $\{\theta, \psi, \omega\}$.

Without loss of generality, we consider attention-based MIL pooling (ABMIL) (Ilse et al., 2018) as an example. This is because most embedding-based MIL models (Zhang et al., 2022; Shao et al., 2021) fall within the regime of ABMIL. Mathematically, an ABMIL pooling function is defined as

$$\begin{aligned} \rho_\psi(\{\boldsymbol{v}_n \mid n = 1, \cdots, N\}) &= \sum_{n=1}^{N} \alpha_n \boldsymbol{v}_n, \\ \text{with } \alpha_n &= \text{softmax}(\boldsymbol{w}_1^T \tanh(\boldsymbol{w}_2 \boldsymbol{v}_n^T)), \end{aligned} \tag{3}$$

where $\psi = \{\mathbf{w}_1 \in \mathbb{R}^{D \times 1}, \mathbf{w}_2 \in \mathbb{R}^{D \times L}\}$ is the trainable parameter, and α_n implies the importance of the n-th instance.

3.1. Analysis of Dropout in MIL

In an MIL framework, Dropout is typically applied to the shallow feature extractor f_θ. For simplicity, considering f_θ as an MLP with L layers, the feature map at the l-th layer of f_θ is a 2-dimensional matrix $\boldsymbol{f}^{(l)} \in \mathbb{R}^{N \times D^{(l)}}$, where $D^{(l)}$ represents the embedding dimension. In this scenario, we consider randomly zeroing out either entries (Hinton et al., 2012) or entire instances (Tompson et al., 2015) in $\boldsymbol{f}^{(l)}$. For brevity, we denote the former as *DropNeuron* and the latter as *DropInstance*. For a given rate $p \in [0, 1]$, both DropNeuron and DropInstance can be defined as performing an element-wise masking operation over the feature map $\boldsymbol{f}^{(l)}$ at the l-th layer of f_θ:

$$\hat{\boldsymbol{f}}^{(l)} = \boldsymbol{f}^{(l)} \odot \boldsymbol{M}^{(l)}, \tag{4}$$

where $\odot$ denotes element-wise multiplication, and $\boldsymbol{M}^{(l)} \in \mathbb{R}^{N \times D^{(l)}}$ is the binary Dropout mask at the l-th layer. In the regime of DropNeuron, each entry $M_{n,d}^{(l)}$ in $\boldsymbol{M}^{(l)}$ are from a Bernoulli distribution: $M_{n,d}^{(l)} \sim Bernoulli(p)$. In contrast, each row in $\boldsymbol{M}^{(l)}$ has the same entry and is sampled from a Bernoulli distribution in the case of DropInstance: $M_n^{(l)} \sim Bernoulli(p)$. However, how and when to impose Dropout remains an open question in the context of MIL, which has yet to be thoroughly explored by previous MIL works. In this section, we provide insight into these two aspects by conducting empirical and theoretical analyses in the relatively complex WSI classification (e.g. Camelyon16) using a representative ABMIL framework. Please refer to Appendix A for the details of these investigation experiments. *Our investigation reveals two main findings: (i) DropInstance leads to flatter local minima and better generalizability compared with DropNeuron, and (ii) DropInstance helps reduce gradient direction errors in learning trajectory, improving data fitting and performance.*

3.2. DropNeuron vs DropInstance.

A common way to assess the effectiveness of Dropout is to examine the robustness of an MIL after applying Dropout. For this purpose, we compute the sharpness of a MIL model (Dinh et al., 2017), which pertains to how sensitive the model's performance is to changes in the model parameters. We consider the ϵ-sharpness (Dinh et al., 2017) as the sharpness measure, which is defined using the local curvature of the loss function. Let $\mathcal{B}_2(\epsilon, \theta)$ be an Euclidean ball centered on a minimum θ with radius ϵ. Then, for a non-negative valued loss function $\mathcal{L}_\theta$, the ϵ-sharpness is proportional to the following quantity:

$$\epsilon\text{-sharpness} \propto \frac{\max_{\theta' \in \mathcal{B}_2(\epsilon,\theta)} \mathcal{L}_{\theta'} - \mathcal{L}_\theta}{1 + \mathcal{L}_\theta}. \tag{5}$$

Geometrically, the ϵ-sharpness is proportional to the height of the area enclosed by the curvature of $\mathcal{L}_\theta$ in Fig. 2(**Left**).

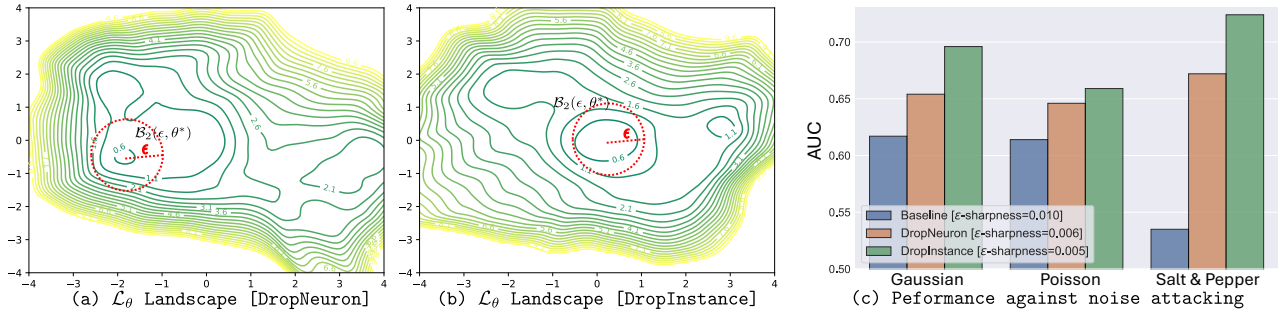

Figure 3. The landscape of the loss function $\mathcal{L}_\theta$ for two different dropout strategies (a) DropNeuron and (b) DropInstance as well as (c) the performance of MIL models against different noise attacks. We mark the Euclidean ball $\mathcal{B}_2(\epsilon, \theta^*)$ around the optimal parameter θ^* (see Eq. 5) in subpanel figure (a) and (b) with a red circle. We observe that the landscape of $\mathcal{L}_\theta$ in the DropInstance scenario leads to flatter minima compared to DropNeuron, which also results in a better performance in AUC.

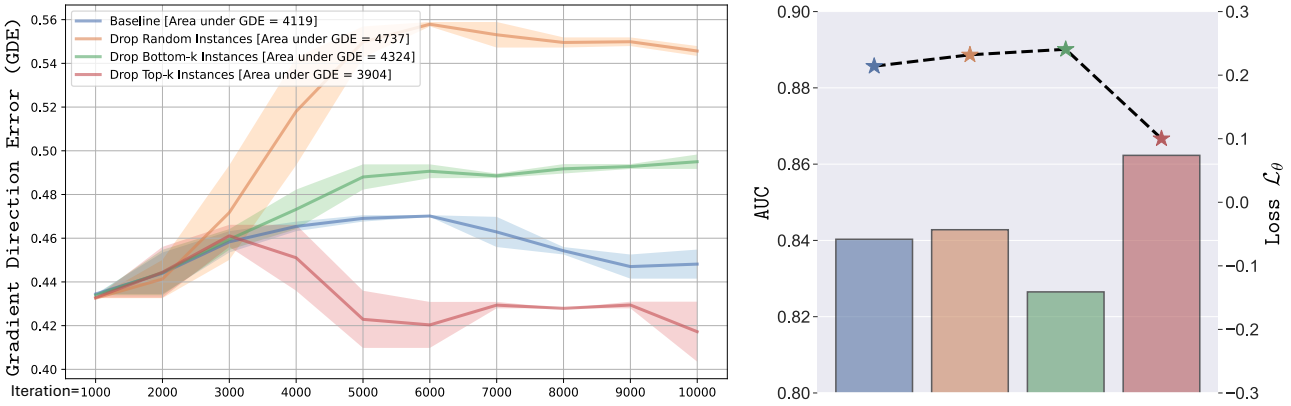

Figure 4. The comparison of change of GDE (**Left**) over the first 10,000 iterations as well as performance and loss (line plot) and AUC (bar plot) when using different instance dropout strategies (**Right**), where the area under GDE is the area enclosed by GDE and the x-axis. Dropping the top-k instances shows the smallest GDE, training loss , and highest AUC among all four strategies.

In the 2D landscape of $\mathcal{L}_\theta$ in Fig. 2(**Right**), the loss function $\mathcal{L}_\theta$ changes more rapidly for sharper minima. As suggested by (Keskar et al., 2016; Foret et al., 2021; Zhang & Xu, 2024), for a specific ϵ, a smaller volume enclosed by the curvature of $\mathcal{L}_\theta$ typically indicates flatter local minima (see Fig. 2), which leads to better generalizability. By applying the second-order Taylor approximation, ϵ-sharpness in Eq. 5 can be rewritten as

$$\epsilon\text{-sharpness} = \frac{\|\nabla_\theta^2 \mathcal{L}_\theta\|_2 \epsilon^2}{2(1 + \mathcal{L}_\theta)}, \tag{6}$$

which enables us to directly compute the ϵ-sharpness. We assess the robustness of the trained MIL model against three different types of noise (i.e., Gaussian, Poisson, and Salt & Pepper noise) by attacking the input tiles/patches with them. The implementation of this experiment is detailed in Appendix A. We observe that adding Dropout lowers the ϵ-sharpness value and results in flatter minima (Fig. 3 (a) and (b)), which also improves the bag-level classification performance (Fig. 3 (c)). In particular, adding DropInstance leads to the best performance. This is also supported by the sharpness analysis of the MIL model (see Fig. 3 (b)), where DropInstance leads to flatter local minima that typically have better generalizability. We conjecture this may be because DropInstance is similar to data augmentation, which generates random instance combinations to improve the performance. However, this observation prompts a question: *Are all instances worth being dropped out, or is the improvement due to specific combinations after DropInstances?*

3.3. How to impose Dropout.

Here, we further investigate how to apply DropInstance. Previous studies have revealed that the effectiveness of algorithms or modules (e.g.Dropout) can be reflected by the gradient direction error (GDE) or gradient variance during model optimization (Liu et al., 2023; Jastrzebski et al., 2020; Kavis et al., 2022). The gradient direction error quantifies the dissimilarity between the mini-batch gradient g_{step} and whole dataset gradient $\hat{g}$:

$$\text{GDE} = \frac{1}{|G|} \sum_{g_{\text{step}} \in G} \frac{1}{2} \left(1 - \frac{\langle g_{\text{step}}, \hat{g} \rangle}{\|g_{\text{step}}\|_2 \cdot \|\hat{g}\|_2}\right),$$

where G is a set of mini-batch gradients. Leveraging GDE, we investigate the impact of three different DropInstance strategies, including dropping (i) top-k instances, (ii) bottom-k instances, and (iii) random instances. Here, top/bottom-k refers to dropping top/bottom k most important instances based on the importance of instances (i.e., attention scores in any attention-based MIL framework) sorted in descending order. For this purpose, we plot the change of GDE as a function of training iterations. As shown in Fig. 4 (**Left**), the GDE of standard MIL does not decrease during model learning. Likewise, the GDE of the MIL models that apply random and bottom-k instance dropout even increases across training iterations. This suggests that the parameter updates are not moving in the correct direction, potentially compromising the fitting of the MIL model. Among the four aforementioned DropInstance strategies, only dropping the top-k instances leads to decreased GDE by model training. Further comparison of the aforementioned four DropInstance strategies in terms of classification performance suggests that a smaller GDE typically leads to better classification performance and lower training loss as shown in Fig. 4 (**Right**).

3.4. Explanation.

In summary, empirical results show that dropping the top-k most important instances typically leads to better performance and gradient error direction. In the context of MIL, a positive bag is defined by the presence of at least one positive instance. However, in practice, the positive instance within each bag diversity leads many models to learn a single set of highly similar positive instances to make decisions. Dropping top-k instances deactivates these learned discriminative instances (zeroing them out in the feature map). With the bag label remaining unchanged, this forces the network to seek additional positive instance representations. From a gradient perspective, mini-batch gradients are generated by different sub-networks that are closer to the whole dataset gradient. This indicates that dropping these top-k instances allows the network to continue learning effectively and enhances performance, explaining why the top-k strategy results in lower gradient error.

4. Proposed MIL-Dropout

Contingent on the above discovery, we introduce a simple yet effective dropout method (termed MIL-Dropout), which can be easily plugged into existing MIL frameworks. MIL-Dropout mainly leverages the instance-based dropout operation to drop the top-k most important instances, as well as instances similar to them. The most important step revolves around selecting these instances in a systematic way. In a previous investigation, the attention map, which indicates the importance of each instance, is obtained from Eq. (11). However, we argue that a key issue with this attention map is that it is produced by previous optimization iteration. Due to the instability of MIL training under weak supervision, top-k instance selection based on this attention map may introduce additional noise, especially at the very beginning of the training. In addition, the top-k most important instances at different layers of the feature extractor f_θ can potentially be different. To mitigate this issue, we propose a non-parametric attention mechanism to select the top-k instances for dropout. Specifically, we note that various methods for natural image processing employ average pooling to compress channel information and produce an attention map (Woo et al., 2018; Park et al., 2018). Inspired by this, we extend average pooling to our method to determine the importance of instances at the l-th layer of the feature extractor f_θ, i.e., $\alpha_n^{(l)}$:

$$\alpha_n^{(l)} = \text{sigmoid}\left(\{\text{avgpool}(\boldsymbol{f}_n^{(l)}) \mid n = 1, \cdots, N\}\right)$$
$$\text{with } \text{avgpool}(\boldsymbol{f}_n^{(l)}) = \frac{1}{D^{(l)}} \sum_{i=1}^{D^{(l)}} \boldsymbol{f}_{n,i}^{(l)}. \quad (7)$$

Here, the average pooling is applied to each instance at embedding dimension $D^{(l)}$ after passing a sigmoid activation function to obtain the importance weight of each instance. Based on the instance importance $\alpha_n^{(l)}$, we first select the top-k most important instances and then group the embedding features $\boldsymbol{f}^{(l)}$ at the l-th layer of f_θ into two groups: (i) top-k $\boldsymbol{f}_T^{(l)}$ and (ii) remaining feature $\boldsymbol{f}_R^{(l)}$.

We argue that simply dropping the top-k instances is insufficient. This is because instances within a bag may not be independent and identically distributed (i.i.d.) in most real-world applications (Tu et al., 2019a; Ilse et al., 2018). This is particularly true for WSI classification, where pathologists consider both the contextual information around a single area and the correlation information between different areas when making decisions. Based on these facts, we also drop instances that are highly similar to the selected top-k instances to further encourage the model to learn diverse representations. For this purpose, we propose a query mechanism to select G number of instances that is highly similar to the selected top-k instances from the remaining features $\boldsymbol{f}_R^{(l)}$ based on their similarity:

$$S_{i,j} = \frac{\boldsymbol{f}_{T,i}^{(l)} (\boldsymbol{f}_{R,j}^{(l)})^\top}{\|\boldsymbol{f}_{T,i}^{(l)}\|_2 \|\boldsymbol{f}_{R,j}^{(l)}\|_2} \quad \forall i \in [K], j \in [N-K], \quad (8)$$

where $S_{i,j}$ denote the similarity score between $\boldsymbol{f}_{T,i}^{(l)}$ and $\boldsymbol{f}_{R,j}^{(l)}$ features, with a higher $S_{i,j}$ indicating higher similarity.

The top-k instances are then combined with their $K \times G$ similar instances to form the final dropout instance set indexed by $\mathcal{A}$, containing $N_{\mathcal{A}}$ instances. Then the remaining instances are indexed by $\bar{\mathcal{A}}$, containing $N_{\bar{\mathcal{A}}}$ instances. Following the convention in Eq. (4), MIL-Dropout can be achieved by masking out the selected instance set $\mathcal{A}$:

$$\tilde{\boldsymbol{f}}_{\mathcal{A}}^{(l)} = \gamma(\boldsymbol{f}^{(l)} \odot \boldsymbol{M}) \quad \text{with} \quad \boldsymbol{M}_{\mathcal{A}} = 0 \text{ and } \boldsymbol{M}_{\bar{\mathcal{A}}} = 1. \tag{9}$$

Here, we add a normalization term $\gamma = N/(N - K(1+G))$ to stabilize the training. Note that each top-k instance may have duplicate similar instances, necessitating deduplication. Therefore, the final number of dropout instances is typically less than $K + (K \times G)$. Similar to other dropout methods (Srivastava et al., 2014; Hinton et al., 2012), we do not apply MIL-Dropout during inference. The algorithmic taxonomy of the MIL-Dropout is shown in Algorithm 1.

4.0.1. Complexity analysis.

Our MIL-Dropout only adds additional two hyperparameters (i.e., top-k number K and similarity instance number G), without involving additional learnable parameters. In each iteration, the main overhead is computing the mask, which is presented as,

$$\underbrace{\mathcal{O}(N \log N)}_{\text{Top-k Sorting}} + \underbrace{\mathcal{O}(K(N-K)D^{(l)})}_{\text{Compute Similarity}} + \underbrace{\mathcal{O}(K(N-K)\log N)}_{\text{Subsequent Sorting}}.$$

The complexity is substantially reduced to the fact that $\mathcal{O}(ND^{(l)})$ due to K is typically much smaller than N, e.g., $K = 20$, whereas $N \approx 5000 \sim 10000$.

Algorithm 1 MIL-Dropout Mechanism

Input: Input feature map $\boldsymbol{f}^{(l)} = \{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_N\}$, K and G
Output: Processed Bag $\hat{\boldsymbol{f}^{(l)}}$ with MIL-Dropout

Initial: $M \leftarrow \mathbf{1}_{K \times D^{(l)}}$ (*# Initial mask*)
Select top-k important instances: $(\boldsymbol{f}_T^{(l)}, \boldsymbol{f}_R^{(l)}) \leftarrow \texttt{split}(\boldsymbol{f}^{(l)})$, $\mathcal{A} \leftarrow [K]$ *(Eq. 7)*
Compute the similarity matrix between rest instances $\boldsymbol{f}_R^{(l)}$ and top-K instances $\boldsymbol{f}_T^{(l)}$
(*# Obtain G instances from $\boldsymbol{f}_R^{(l)}$ that are most similar to every top-K instance*)
for $i = 1$ to K **do**
  $A_i = \arg\max_{S \subseteq R, |S|=G} \sum_{j \in S} S_{i,j}$ *(Eq. 8)*
  $\mathcal{A} \leftarrow \mathcal{A} \cup A_i$
end for
$M[\mathcal{A}, :] = 0$
$\hat{\boldsymbol{f}^{(l)}} \leftarrow \gamma(\boldsymbol{f}^{(l)} \odot M)$ *(Eq. 9)*
return $\hat{\boldsymbol{f}^{(l)}}$ (*# Masking and normalization*)

5. Experimental designs

5.1. MIL benchmarks.

The benchmark datasets include MUSK1, MUSK2, FOX, TIGER, and ELEPHANT, which are commonly used to evaluate and compare the performance of MIL algorithms. MUSK1 and MUSK2 (Dietterich et al., 1997) focus on molecule classification, with each bag containing instances representing atoms. Conversely, FOX, TIGER, and ELEPHANT (Andrews et al., 2002) involve image classification, where each bag represents images and contains instances that represent patches within those images. As consistent with the experimental protocols outlined in (Li et al., 2021), all experiments are conducted five times with a 10-fold cross-validation. We report the mean ($\pm$ std) for all MIL benchmark datasets.

5.2. WSI datasets.

The CAMELOYON16 dataset aims to identify metastatic breast cancer in lymph node tissue and consists of high-resolution digital WSIs. It is divided into a training set of 270 samples and a testing set of 129 samples. The TCGA-NSCLC dataset primarily identifies two subtypes of lung cancer: lung squamous cell carcinoma and lung adenocarcinoma. As outlined in (Li et al., 2021), 1037 WSIs were divided into a training set of 744 WSIs, a validation set of 83 WSIs, and a testing set of 210 WSIs. Following the threshold (Li et al., 2021) and OTSU (Zhang et al., 2022) preprocessing methods, each WSI was divided into non-overlapping 224×224 patches at a magnification of $\times 20$. Two sets of patches were extracted using different frameworks: ResNet-50 from DTFD-MIL, yielding 1024-dimensional vectors per patch, and the SimCL contrastive learning framework from DSMIL, yielding 512-dimensional vectors per patch.

5.3. Baselines.

On five MIL benchmark datasets, we mainly plug MIL-Dropout into ABMIL as a proof of concept, while comparing with other baseline methods: mi-Net, MI-Net, MI-Net with DS, MI-Net wth RC (Wang et al., 2018b), ABMIL, ABMIL-Gated (Ilse et al., 2018), GNN-MIL (Tu et al., 2019b), DP-MINN (Yan et al., 2018), NLMIL (Wang et al., 2018a), ANLMIL (Zhu et al., 2019), and DSMIL (Li et al., 2021). On the WSI datasets, we plug MIL-Dropout into four state-of-the-art MIL aggregators and their variants to valid the effectiveness of the proposed method, i.e., ABMIL (Ilse et al., 2018), DSMIL (Li et al., 2021), TransMIL (Shao et al., 2021), and DTFD-MIL (Zhang et al., 2022).

5.4. Evaluation metrics and implementation details.

For the MIL benchmark dataset, we performed five times with a 10-fold cross-validation for each method and reported

Table 1. Performance comparison on MIL benchmark datasets. Each experiment is performed five times with 10-fold cross-validation. We reported the mean of the classification accuracy ($\pm$ the standard deviation of the mean).

Methods	MUSK1	MUSK2	FOX	TIGER	ELEPHANT
mi-Net	0.889 ± 0.039	0.858 ± 0.049	0.613 ± 0.035	0.824 ± 0.034	0.858 ± 0.037
MI-Net	0.887 ± 0.041	0.859 ± 0.046	0.622 ± 0.038	0.830 ± 0.032	0.862 ± 0.034
MI-Net with DS	0.894 ± 0.042	0.874 ± 0.043	0.630 ± 0.037	0.845 ± 0.039	0.872 ± 0.032
MI-Net with RC	0.898 ± 0.043	0.873 ± 0.044	0.619 ± 0.047	0.836 ± 0.037	0.857 ± 0.040
ABMIL	0.892 ± 0.040	0.858 ± 0.048	0.615 ± 0.043	0.839 ± 0.022	0.868 ± 0.022
ABMIL-Gated	0.900 ± 0.050	0.863 ± 0.042	0.603 ± 0.029	0.845 ± 0.018	0.857 ± 0.027
GNN-MIL	0.917 ± 0.048	0.892 ± 0.011	0.679 ± 0.007	0.876 ± 0.015	0.903 ± 0.010
DP-MINN	0.907 ± 0.036	0.926 ± 0.043	0.655 ± 0.052	0.897 ± 0.028	0.894 ± 0.030
NLMIL	0.921 ± 0.017	0.910 ± 0.009	0.703 ± 0.035	0.857 ± 0.013	0.876 ± 0.011
ANLMIL	0.912 ± 0.009	0.822 ± 0.084	0.643 ± 0.012	0.733 ± 0.068	0.883 ± 0.014
DSMIL	0.932 ± 0.023	0.930 ± 0.020	0.729 ± 0.018	0.869 ± 0.008	0.925 ± 0.007
ABMIL + MIL-Dropout	0.964 ± 0.033	0.954 ± 0.019	$\mathbf{0.789 \pm 0.043}$	0.917 ± 0.036	$\mathbf{0.934 \pm 0.046}$
ABMIL-Gated + MIL-Dropout	$\mathbf{0.967 \pm 0.019}$	$\mathbf{0.958 \pm 0.021}$	0.788 ± 0.016	$\mathbf{0.919 \pm 0.033}$	0.927 ± 0.033

the mean of the classification accuracy and the standard deviation. For two WSI datasets, we report the classification mean of accuracy, F1, and AUC were reported with standard deviation based on running five times of each experiment without fixing random seed. All baselines were implemented with the parameter configurations specified in their original papers. To incorporate the MIL dropout, we employed a unified specific architecture involving adding three fully connected layers as the shallow feature extractor f_θ before entering MIL aggregation. Furthermore, We reported all detailed experiment settings in Appendix B.

6. Results

6.1. MIL benchmark results.

The integration of the proposed MIL-Dropout into ABMIL and ABMIL-Gated leads to superior performance compared to all prior state-of-the-art (SOTA) methods across five MIL benchmark datasets, as shown in Table 1. Notably, ABMIL augmented with MIL-Dropout achieves SOTA accuracy of 78.9% on the FOX dataset and 93.4% on the ELEPHANT dataset, along with an average accuracy improvement of 9.72% across the five datasets compared to the baseline ABMIL. Similarly, ABMIL-Gated with Dropout reaches SOTA accuracy of 96.7% on MUSK1, 95.8% on MUSK2, and 91.9% on TIGER, demonstrating an average accuracy improvement of 9.82% across the five datasets over the baseline ABMIL-Gate. Through extensive experiments, we observe that integrating our proposed Dropout into the two simplest bags embedding MIL aggregation (ABMIL and ABMIL-Gated) can also achieve SOTA performance.

6.2. WSI results.

As shown in Table 2, we perform a comparative study to evaluate the effectiveness of integrating the proposed MIL-Dropout into current MIL aggregators on CAMELYON16 and TCGA-NSCLC WSI datasets. Experimental results show that MIL-Dropout boosts the performance of four different types of MIL aggregators across these two WSI datasets with features extracted using different means. Specifically, applying MIL-Dropout results in average improvements of 2.3% in accuracy, 2.46% in F1 score, and 2.48% in AUC, respectively, on the CAMELYON16 dataset. Similarly, on the TCGA-NSCLC dataset, MIL-Dropout leads to average increases of 2.05% in accuracy, 2.44% in F1 score, and 1.80% in AUC. Remarkably, integrating MIL-Dropout into simple attention-based MIL aggregators (ABMIL and ABMIL-Gated) achieve a performance that is comparable to or even surpasses state-of-the-art methods. This shows that even ABMIL can significantly improve representation learning with MIL-Dropout, addressing underfitting and enhancing latent feature discovery. Interestingly, we observe that MIL aggregators trained with features extracted using SimCLR outperforms that trained with features extracted by SimCLR on the CAMELYON16 dataset, but shows similar performance on the TCGA-NSCLC dataset. With the proposed MIL-Dropout, the performance of both ImageNet and self-supervised pre-trained backbone is largely indistinguishable under optimal conditions. On the TCGA-NSCLC dataset, both extractors result in similar performance, suggesting that more training samples improve the generalization of self-supervised feature extractors. However, our MIL-Dropout can enhance the generalization of different feature extractors without an additional self-supervised training stage.

6.3. Ablation analysis.

We perform ablation studies on two key hyperparameters in MIL-Dropout: the number of top-k instances (K) and the number of instances similar to the top-k instances (G). These analyses are carried out on the CAMELYON16 and

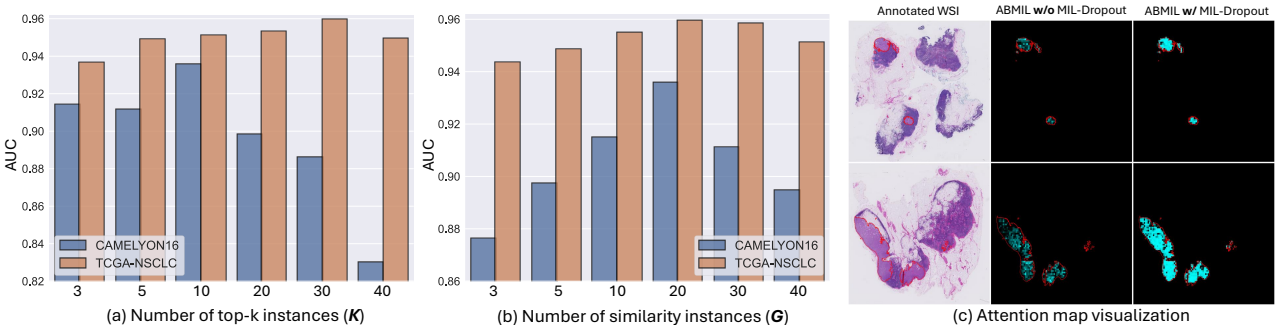

Figure 5. Ablation studies on the number of top-k instances K (a) and similarity instance S (b) using CAMELYON16 and TCGA-NSCLC datasets. (c) Attention map from ABMIL without and with MIL-Dropout, with tumor regions outlined in red. Brighter cyan in columns two and three indicates higher tumor probability (higher attention score) for corresponding locations.

Table 2. Comparison of performance before and after plugging MIL-Dropout into four different types of MIL aggregators and their variants on CAMELOYON16 and TCGA-NSCLC datasets. Δ denotes the performance gains after the integration of MIL-Dropout. The classification accuracy (%), F1 score (%), and AUC (%) are reported (± the standard deviation of the mean) by running each experiment five times.

		CAMELOYON16						TCGA-NSCLC					
		ImageNet Pretrained			SimCLR Pretrained			ImageNet Pretrained			SimCLR Pretrained		
		Accuracy	F1	AUC	Accuracy	F1	AUC	Accuracy	F1	AUC	Accuracy	F1	AUC
ABMIL		86.3 ± 1.1	85.0±1.0	86.0 ± 0.5	85.6 ± 0.9	84.2 ± 1.3	86.6 ± 1.4	87.5 ± 0.8	87.5 ± 0.8	92.4 ± 0.5	87.9 ± 0.8	88.1 ± 0.6	93.8 ± 0.8
	+MIL Dropout	87.2 ± 1.0	86.4±0.8	90.1± 0.8	88.6 ± 1.1	87.4 ± 1.0	88.3 ± 1.2	91.1 ± 1.3	91.1 ± 1.3	95.6 ± 0.4	91.4 ± 0.6	91.5 ± 0.5	95.9 ± 0.1
	Δ	**+0.9**	**+1.4**	**+4.1**	**+3.0**	**+3.2**	**+1.7**	**+3.6**	**+3.6**	**+3.2**	**+3.5**	**+3.4**	**+2.1**
ABMIL-Gated		86.9 ± 1.1	85.7±1.2	86.2 ± 1.2	84.3 ± 1.1	83.4 ± 1.0	85.9 ± 1.6	87.9 ± 0.9	87.9± 0.9	92.8 ± 0.9	89.0 ± 1.2	89.0 ± 1.2	94.4 ± 0.7
	+MIL Dropout	90.4 ± 1.3	89.6±1.2	90.7 ± 0.9	87.7 ± 1.3	86.7 ± 1.3	87.4 ± 0.9	90.0 ± 0.6	90.0 ± 0.6	95.3 ± 0.3	90.8 ± 0.7	90.8 ± 0.7	95.8 ± 0.2
	Δ	**+3.5**	**+3.9**	**+4.6**	**+3.4**	**+4.2**	**+1.5**	**+2.1**	**+2.1**	**+2.5**	**+1.8**	**+1.8**	**+1.4**
DSMIL		85.5± 0.8	84.3± 1.1	89.0± 1.8	83.3 ± 1.0	82.0 ± 1.4	85.9 ± 1.6	89.3 ± 0.7	89.4± 0.7	94.2 ± 0.3	84.1 ± 1.8	86.2 ± 1.5	92.0 ± 1.6
	+MIL Dropout	87.9± 1.5	86.8±1.6	90.6± 1.2	85.6 ± 0.9	84.8 ± 0.5	87.6 ± 0.8	89.9 ± 0.6	90.0± 0.5	95.3 ± 0.6	86.9 ± 0.4	88.3 ± 0.2	93.9 ± 0.3
	Δ	**+2.4**	**+2.6**	**+1.6**	**+2.3**	**+2.8**	**+1.7**	**+0.6**	**+0.6**	**+1.1**	**+2.8**	**+2.1**	**+1.9**
TransMIL		84.7 ± 2.1	83.3±2.9	86.5 ± 2.4	86.8 ± 1.0	85.9 ± 1.2	89.7 ± 0.6	86.9 ± 0.6	87.0 ± 0.6	93.3 ± 0.7	88.2 ± 2.1	88.3 ± 2.1	94.6 ± 1.1
	+ MIL Dropout	86.0 ± 1.5	84.9±1.5	89.4 ± 0.9	89.7 ± 1.3	88.7 ± 1.4	90.3 ± 1.2	88.0 ± 0.5	88.5 ± 1.1	94.3 ± 0.4	91.6 ± 0.9	92.0 ± 0.7	96.2 ± 0.6
	Δ	**+1.3**	**+1.3**	**+2.9**	**+2.9**	**+2.8**	**+0.6**	**+1.1**	**+1.6**	**+1.0**	**+2.8**	**+3.7**	**+1.6**
DTFD-MIL(AFS)		84.1 ± 0.6	75.5 ± 0.6	88.2 ± 0.3	87.4 ± 0.9	81.8 ± 1.2	89.6 ± 0.9	88.5 ± 0.5	88.0 ± 0.3	94.4 ± 0.2	87.6 ± 0.3	87.8 ± 0.4	93.1 ± 0.2
	+MIL Dropout	85.7 ± 1.4	79.1 ± 2.2	89.9 ± 0.6	88.5 ± 0.7	84.2 ± 0.6	92.5 ± 1.0	90.3 ± 0.4	90.0 ± 0.4	94.8 ± 0.1	91.5 ± 0.4	91.8 ± 0.4	96.1 ± 0.2
	Δ	**+1.6**	**+3.6**	**+1.6**	**+1.5**	**+2.4**	**+2.9**	**+1.8**	**+2.0**	**+0.4**	**+3.9**	**+4.0**	**+3.0**
DTFD-MIL(MaxS)		84.7 ± 1.8	78.3 ± 2.4	87.8 ± 0.8	87.7 ± 1.5	82.0 ± 2.3	88.4 ± 0.9	87.4 ± 1.0	87.3± 1.0	93.8 ± 0.1	85.1 ± 1.2	84.9 ± 2.2	91.0 ± 1.0
	+MIL Dropout	86.5 ± 0.9	81.0 ± 1.3	89.8 ± 0.9	89.5 ± 0.4	84.4 ± 0.4	91.6 ± 0.5	88.8 ± 0.5	88.4 ± 0.5	95.0 ± 0.4	87.5 ± 2.5	88.2 ± 2.2	93.2 ± 1.0
	Δ	**+1.8**	**+2.7**	**+2.0**	**+1.8**	**+2.4**	**+3.2**	**+1.4**	**+1.1**	**+1.2**	**+2.4**	**+3.3**	**+2.2**

TCGA-NSCLC datasets, using features extracted by ResNet-50 pretrained on ImageNet. The optimal values for K varies from dataset to dataset, with $K = 10$ on CAMELYON16 and $K = 30$ on TCGA-NSCLC (Fig. 5(a) and (b)). This suggests that TCGA-NSCLC may have large variability in instances. Additionally, a too-large K and G may disrupt MIL tasks by discarding too much information, whereas a too-small K and G may not be sufficient enough to drop easily identified positive instances and encourage MIL to discover richer features. As a result, we seek to find an equilibrium point that balances these effects.

6.4. Lesion localization.

As shown in Fig. 5 (c), ABMIL with MIL-Dropout provides better localization with a higher attention score to the positive instances compared to ABMIL without MIL-Dropout. In contrast, the attention map of vanilla ABMIL may either miss some true positive regions or show a lower importance. These results show evidence that MIL-Dropout can potentially help uncover more rich and generalizable lesion features,instead of relying on a single group of surely identified positive instances to make decisions. This aligns with the underlying rationale and theoretical guarantees behind the success of MIL-Dropout.

7. Conclusion

In this paper, we thoroughly analyze the role of dropout-like methods in MIL. Our observations empirically demonstrate that simply dropping certain instances can significantly enhance the performance and generalization of existing MIL methods. We further propose an MIL-Dropout to systematically determine the instances to drop by leveraging an attention-based mechanism to weigh instance importance and a query-based mechanism to select instance sets. Experi-

ments on several MIL datasets demonstrate the effectiveness of our MIL-Dropout. We hope our findings can guide the design of MIL models to learn more rich and generalizable features.

Acknowledgment

This material is based upon the work supported by the National Science Foundation under Grant Number CNS-2204721. It is also supported by our collaborative project with MIT Lincoln Lab under Grant Numbers 2015887 and 7000612889. Additionally, this work was partially supported by National Institutes of Health, United States (R01EY032125 and R21AG065942) and the State of Arizona via the Arizona Alzheimer Consortium.

Impact Statement

This paper proposes a novel dropout regularization method for multiple instance learning, derived from an empirical and theoretical study. The potential societal implications of MIL-Dropout can be categorized into three main areas:

(i) Theoretical View. Starting from a generalization perspective, we investigate two commonly used dropout methods in MIL—dropout and drop instance. We find that drop instance not only provides stronger generalization but also improves performance. Building on this, we analyze different instance-dropping strategies from an optimization standpoint and conclude that dropping the top-k instances achieves a lower gradient-direction error and reaches a flatter minimum with better generalization. Our findings and theoretical insights also fill an important gap for subsequent research on regularization in MIL.

(ii) Applicability. Our dropout method can be seamlessly integrated with existing MIL approaches and applied to all MIL algorithms, whether pathology or other weakly supervised classification tasks in natural images.

(iii) Interpretability. From Interpretability perspective, our method can enhance the localization capability in MIL-based weak supervision and help ensure model robustness. This, in turn, facilitates the identification of vulnerabilities and reduces the risk of adversarial attacks.

Limitation. At present, we have not extended this method to a broader range of tasks, such as video frame detection or 3D point cloud classification. Moreover, the current MIL dropout approach does not account for instance uncertainty, which could be a promising direction for guiding instance-drop strategies. We plan to explore these aspects in future work.

References

Andrews, S., Tsochantaridis, I., and Hofmann, T. Support vector machines for multiple-instance learning. *Advances in neural information processing systems*, 15, 2002.

Babenko, B., Yang, M.-H., and Belongie, S. Robust object tracking with online multiple instance learning. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1619–1632, 2010.

Chen, R. J., Ding, T., Lu, M. Y., Williamson, D. F., Jaume, G., Chen, B., Zhang, A., Shao, D., Song, A. H., Shaban, M., et al. Towards a general-purpose foundation model for computational pathology. *Nature Medicine*, 2024a.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Chen, X., Qiu, P., Zhu, W., Li, H., Wang, H., Sotiras, A., Wang, Y., and Razi, A. Timemil: advancing multivariate time series classification via a time-aware multiple instance learning. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 7190–7206, 2024b.

Choe, J. and Shim, H. Attention-based dropout layer for weakly supervised object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2219–2228, 2019.

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

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pp. 1019–1028. PMLR, 2017.

Early, J., Cheung, G. K., Cutajar, K., Xie, H., Kandola, J., and Twomey, N. Inherently interpretable time series classification via multiple instance learning. *arXiv preprint arXiv:2311.10049*, 2023.

Feng, J. and Zhou, Z.-H. Deep miml network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=6Tm1mposlrM.

Ghiasi, G., Lin, T.-Y., and Le, Q. V. Dropblock: A regularization method for convolutional networks. *Advances in neural information processing systems*, 31, 2018.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Hou, L., Samaras, D., Kurc, T. M., Gao, Y., Davis, J. E., and Saltz, J. H. Patch-based convolutional neural network for whole slide tissue image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2424–2433, 2016.

Huang, Y., Zhao, W., Fu, Y., Zhu, L., and Yu, L. Unleash the power of state space model for whole slide image with local aware scanning and importance resampling. *IEEE Transactions on Medical Imaging*, 2024.

Ilse, M., Tomczak, J., and Welling, M. Attention-based deep multiple instance learning. In *International conference on machine learning*, pp. 2127–2136. PMLR, 2018.

Jastrzebski, S., Szymczak, M., Fort, S., Arpit, D., Tabor, J., Cho, K., and Geras, K. The break-even point on optimization trajectories of deep neural networks. *arXiv preprint arXiv:2002.09572*, 2020.

Jaume, G., Vaidya, A., Chen, R. J., Williamson, D. F., Liang, P. P., and Mahmood, F. Modeling dense multimodal interactions between biological pathways and histology for survival prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11579–11590, 2024.

Kavis, A., Skoulakis, S., Antonakopoulos, K., Dadi, L. T., and Cevher, V. Adaptive stochastic variance reduction for non-convex finite-sum minimization. *Advances in Neural Information Processing Systems*, 35:23524–23538, 2022.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Li, B., Li, Y., and Eliceiri, K. W. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14318–14328, 2021.

Liu, Z., Xu, Z., Jin, J., Shen, Z., and Darrell, T. Dropout reduces underfitting. In *International Conference on Machine Learning*, pp. 22233–22248. PMLR, 2023.

Park, J., Woo, S., Lee, J.-Y., and Kweon, I. S. Bam: Bottleneck attention module. *arXiv preprint arXiv:1807.06514*, 2018.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Qiu, P., Xiao, P., Zhu, W., Wang, Y., and Sotiras, A. Sc-mil: Sparsely coded multiple instance learning for whole slide image classification. *arXiv preprint arXiv:2311.00048*, 2023.

Qu, L., Yang, Z., Duan, M., Ma, Y., Wang, S., Wang, M., and Song, Z. Boosting whole slide image classification from the perspectives of distribution, correlation and magnification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21463–21473, 2023.

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

Shao, Z., Bian, H., Chen, Y., Wang, Y., Zhang, J., Ji, X., et al. Transmil: Transformer based correlated multiple instance learning for whole slide image classification. *Advances in neural information processing systems*, 34:2136–2147, 2021.

Song, A. H., Chen, R. J., Jaume, G., Vaidya, A. J., Baras, A. S., and Mahmood, F. Multimodal prototyping for cancer survival prediction. *arXiv preprint arXiv:2407.00224*, 2024.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

Tang, W., Huang, S., Zhang, X., Zhou, F., Zhang, Y., and Liu, B. Multiple instance learning framework with masked hard instance mining for whole slide image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4078–4087, 2023.

Tompson, J., Goroshin, R., Jain, A., LeCun, Y., and Bregler, C. Efficient object localization using convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 648–656, 2015.

Tu, M., Huang, J., He, X., and Zhou, B. Multiple instance learning with graph neural networks. *arXiv preprint arXiv:1906.04881*, 2019a.

Tu, M., Huang, J., He, X., and Zhou, B. Multiple instance learning with graph neural networks. *arXiv preprint arXiv:1906.04881*, 2019b.

Wang, X., Girshick, R., Gupta, A., and He, K. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7794–7803, 2018a.

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

Woo, S., Park, J., Lee, J.-Y., and Kweon, I. S. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

Xiang, J. and Zhang, J. Exploring low-rank property in multiple instance learning for whole slide image classification. In *The Eleventh International Conference on Learning Representations*, 2023.

Xu, G., Song, Z., Sun, Z., Ku, C., Yang, Z., Liu, C., Wang, S., Ma, J., and Xu, W. Camel: A weakly supervised learning framework for histopathology image segmentation. In *Proceedings of the IEEE/CVF International Conference on computer vision*, pp. 10682–10691, 2019.

Yan, Y., Wang, X., Guo, X., Fang, J., Liu, W., and Huang, J. Deep multi-instance learning with dynamic pooling. In *Asian Conference on Machine Learning*, pp. 662–677. PMLR, 2018.

Zhang, H., Meng, Y., Zhao, Y., Qiao, Y., Yang, X., Coupland, S. E., and Zheng, Y. Dtfd-mil: Double-tier feature distillation multiple instance learning for histopathology whole slide image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18802–18812, 2022.

Zhang, Z. and Xu, Z.-Q. J. Implicit regularization of dropout. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Zheng, T., Jiang, K., and Yao, H. Dynamic policy-driven adaptive multi-instance learning for whole slide image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8028–8037, 2024.

Zhu, W., Qiu, P., Chen, X., Dumitrascu, O. M., and Wang, Y. Pdl: Regularizing multiple instance learning with progressive dropout layers. *arXiv preprint arXiv:2308.10112*, 2023a.

Zhu, W., Qiu, P., Lepore, N., Dumitrascu, O. M., and Wang, Y. Self-supervised equivariant regularization reconciles multiple-instance learning: Joint referable diabetic retinopathy classification and lesion segmentation. In *18th International Symposium on Medical Information Processing and Analysis*, volume 12567, pp. 100–107. SPIE, 2023b.

Zhu, W., Chen, X., Qiu, P., Sotiras, A., Razi, A., and Wang, Y. Dgr-mil: Exploring diverse global representation in multiple instance learning for whole slide image classification. In *European Conference on Computer Vision*, pp. 333–351. Springer, 2024.

Zhu, Z., Xu, M., Bai, S., Huang, T., and Bai, X. Asymmetric non-local neural networks for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 593–602, 2019.

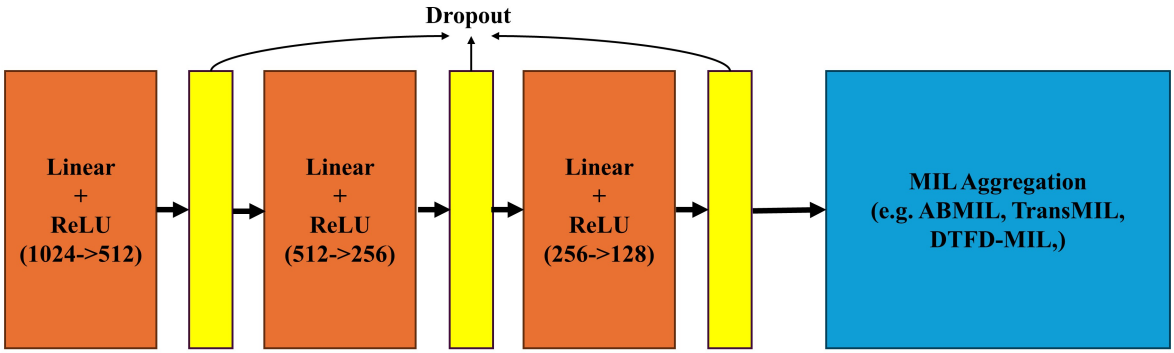

Figure 6. ABMIL aggregation with Dropout setting within investigation experiments .

A. Investigation Experiment Design

We implementation all experiments on a node of cluster with NVIDIA V100 (32GB). We use Pytorch Library (Paszke et al., 2019) with version of 1.13. The investigation experiments employ the same set of CAMELYON16 extractor features; for the pre-processing stage, we utilized Otsu's thresholding method to localize tissue regions within each slide image (WSI). Subsequently, non-overlapping patches, each measuring 256 × 256 pixels at 20X magnification, were extracted from these localized tissue regions. In total, 3.7 million patches were obtained from the CAMELYON-16 dataset (Zhang et al., 2022). Following the previous work (Zhang et al., 2022), The pre-trained Resnet-50 on ImageNet serves as the feature extractor for the embedding patch to a 1024-dimensional vector. The investigation experiments use the training/testing set provided by CAMELYON16 officials.

For these two experiments, we uniformly adopted the ABMIL (Ilse et al., 2018) with cross-entropy loss, and the Lookahead optimizer was employed with a learning rate of 1e-4 and weight decay of 1e-4. To integrate Dropout into ABMIL, we employed three fully connected layers and ReLU before ABMIL aggregation(1024→512→256→128). After applying the ReLU activation function to each layer, dropout regularization is introduced,as shown in Figure 6.

A.1. Detail of Generalization experiment

We conducted this experiment to investigate the CAMELYON16 dataset. We train the model on the original training set and test its generalization on the test data with artifact noise to mimic the domain shift between training data and the unseen test set in real-world scenarios. We consider some common types of noise in the imagining process. Specifically, we add five levels for each type of noise: i) for Gaussian noise $\sigma \in \{0.01, 0.05, 0.1, 0.2, 0.3\}$, ii) for Poisson noise, we set its scale to $\{1., 5., 10., 20., 25.\}$, and iii) for Salt and Pepper noise, we set the probability of both salt and pepper to $\{0.01, 0.03, 0.05, 0.07, 0.09\}$. Therefore, we have 15 different noisy test sets now. Fig.7 shows the exemplary patch with different noises. Apparently, a model with better generalization should be able to perform better on most of these datasets.

Now, let us elaborate on how to derive Eq. 5 to Eq. 6. The Taylor expansion for $L(\theta')$ presented in Eq. 5 can be expressed as,

$$L(\theta') \approx L(\theta) + \nabla L(\theta)^\top (\theta' - \theta) + + \frac{1}{2}(\theta' - \theta)^\top \nabla^2 L(\theta)(\theta' - \theta) + \mathcal{O}(\theta' - \theta).$$

Due to θ is a local minimal, $\nabla L(\theta) = 0$, thereby,

$$L(\theta') - L(\theta) = \frac{1}{2}(\theta' - \theta)^\top \nabla^2 L(\theta)(\theta' - \theta). \tag{10}$$

The maximum of a quadratic form occurs when $(\theta' - \theta)$ aligns with the eigenvector corresponding to the largest eigenvalue of $\nabla^2 L(\theta)$, therefore,

$$\frac{\max_{\theta' \in B_2(\epsilon,\theta)} \left(L\left(\theta'\right) - L(\theta)\right)}{1 + L(\theta)} = \frac{\epsilon^2 \|\nabla^2 L(\theta)\|_2}{2(1 + L(\theta))},$$

where $\|\nabla^2 L(\theta)\|_2$ denotes the spectral norm (largest eigenvalue).

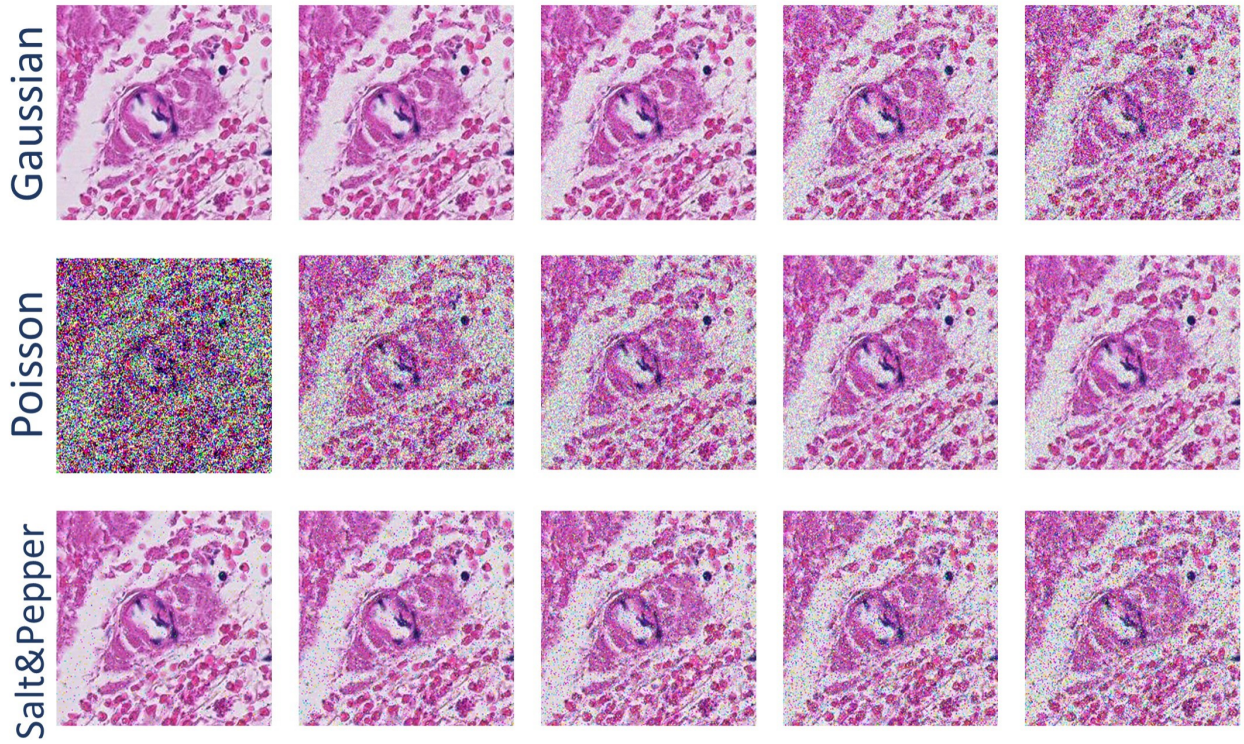

Figure 7. The exemplary patch with different noise types and strengths.

Table 3. Parameters setting for baseline MIL methods. Here employed the AdamW, RAdam, CosineAnnealingLR to follow these MIL methods papers setting.

	ABMIL	ABMIL-Gate	DSMIL	TransMIL	DTFD-MIL
Optimizer	AdamW	AdamW	AdamW	RAdam	Adam
Learning rate	$1e^{-4}$	$1e^{-4}$	$1e^{-4}$	$1e^{-4}$	$2e^{-4}$
Weight decay	$1e^{-4}$	$1e^{-4}$	$5e^{-3}$	$5e^{-3}$	$2e^{-3}$
Optimizer scheduler	LookAhead	LookAhead	CosineAnnealingLR	LookAhead	MultiStepLR
Loss function	CrossEntropy	CrossEntropy	CrossEntropy	CrossEntropy	CrossEntropy + distille loss

A.1.1. GRDIENT MEASURE EXPERIMENT

We conducted this experiment based on the CAMELYON16 training and testing set. The primary objective of fitting the training data is to minimize the loss across the entire training set, rather than focusing on any individual mini-batch. Following in (Liu et al., 2023), We compute the gradient for a given model over the entire training set, setting DropInstance to inference mode to capture the full model's gradient. Subsequently, we evaluate the deviation of the actual mini-batch gradient g_{step} from this whole-dataset "ground-truth" gradient $\hat{g}$. We define the average cosine distance from all g_{step} to $\hat{g}$ as the gradient direction error (GDE):

$$\text{GDE} = \frac{1}{|G|} \sum_{g_{\text{step}} \in G} \frac{1}{2} \left(1 - \frac{\langle g_{\text{step}}, \hat{g} \rangle}{\|g_{\text{step}}\|_2 \cdot \|\hat{g}\|_2} \right),$$

Here, we employ the same architecture as shown in Fig 6 for integrating Dropout. In this manuscript, we investigate four different DropInstance modes: top-k, bottom-k, random DropInstance, and without DropInstance. For the CAMELYON16 dataset, the value of k is consistently set to 20. The terms "top" and "bottom" refer to the ABMIL aggregation attention map

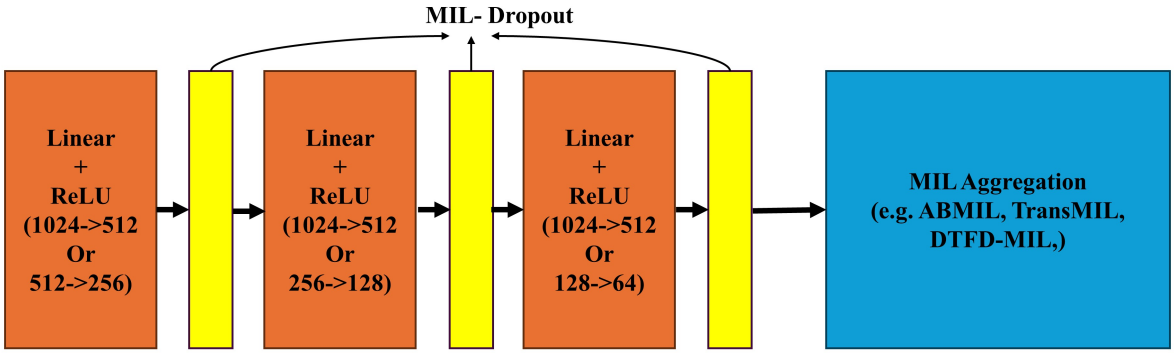

Figure 8. Different aggregation methods with MIL-Dropout setting within main experiments .

obtained from the previous iteration:

$$\rho_\psi(\{\boldsymbol{v}_n \mid n = 1, \cdots, N\}) = \sum_{n=1}^{N} \alpha_n \boldsymbol{v}_n, \quad \text{with } \alpha_n = \text{softmax}(\boldsymbol{w}_1^T \tanh(\boldsymbol{w}_2 \boldsymbol{v}_n^T)), \tag{11}$$

where α_n implies the importance of the n-th instance. In this context, the top and bottom refer to the ranking of attention scores in descending order, with the top corresponding to the highest attention scores and the bottom to the lowest. In addition, we conducted a performance comparison on the testing set. We observed that these models converge after approximately 10,000 iterations and achieve optimal performance prior to this point. In the manuscript, we present the highest AUC achieved before 10,000 iterations based on the four different DropInstance modes.

B. Main Experiment Details

We provide details of integrating MIL-Dropout into existing MIL methods, including ABMIL (Ilse et al., 2018), DTFD-MIL (Zhang et al., 2022), TransMIL (Shao et al., 2021), DSMIL (Li et al., 2021).

B.1. MIL Benchmark experiments setting

B.1.1. Embedding features

For the MUSK1 and MUSK2, A bag is constructed for each molecule, as instances. Each instance is represented with a 166-dimensional embedding feature. The FOX, TIGER, and ELEPHANT datasets contain 200 bags of instance features. Each instance is associated with a 230-dimensional embedding feature.

B.1.2. Detail of integrating MIL Dropout into existing MIL architectures

We employed three fully connected layers in shallow feature extractor f_θ, each comprising 256, 128, and 64 hidden units. Each layer was equipped with a ReLU activation function and subsequently appended by a MIL Dropout layer (refer to Fig 8). Following the passage of features through the f_θ, the ABMIL, ABMIL-Gate, and DSMIL aggregators processed the 64-dimensional embedding feature as their input. To compare original-based models, all aggregators adhere to the configuration outlined in paper (Li et al., 2021), with distinct embedding features of dimensions 230 and 166 directly employed as inputs. All experiments used the Adam optimizer with $2e^{-4}$ learning rates and $5e^{-3}$ weight decay and trained on cross-entropy loss for 40 epochs.

B.2. WSIs experimental setting

B.2.1. Preprocessing WSI

Following the threshold (Li et al., 2021) and OTSU (Zhang et al., 2022) preprocessing methods, each WSI was divided into non-overlapping 224×224 patches at a magnification of $\times 20$. This results in a total of 3.7 million patches ($\sim$ 9000 per

bag) from the CAMELYON16 dataset and 8.3 million patches ($\sim$ 8000 per bag) from the TCGA Lung Cancer dataset.

B.2.2. Embedding Network pretrained

Two sets of patches were extracted using different frameworks: `(i) ImageNet Pretrained`: ResNet-50 from DTFD-MIL, yielding 1024-dimensional vectors per patch, and `(i) SimCLR Pretrained`: the SimCL contrastive learning framework from DSMIL, yielding 512-dimensional vectors per patch. The self-supervised SimCLR manner employed the contrastive learning framework (Chen et al., 2020) to pretrain the projector based on the training set, wherein contrastive loss training was implemented between extracted patches and corresponding two random data data-augmentation counterparts (Li et al., 2021).

B.2.3. Detail of integrating MIL Dropout into existing MIL architectures

We followed the parameter settings outlined in the original literature for the baseline experiments on the two WSI datasets, as shown in Table 3. To integrate MIL Dropout into these MIL methods, we employed three fully connected layers in shallow feature extractor f_θ, each comprising 512 / 256, 256 / 128, and 128 / 64 hidden units. Each layer was equipped with a ReLU activation function and subsequently appended by a MIL Dropout layer as shown in Fig 8. The input dimensions of these networks were adjusted to 64 while keeping the other parameters unchanged. All the experiments were trained on 200 epochs. The MIL Dropout parameters were identical to the paper description (section Implementation details). The experiments integrated with MIL Dropout also followed the parameters in Table 3.

C. Rebuttal Extra Experiment

C.1. UNI feature (Chen et al., 2024a)

Consistent with requirements from other reviewers, we have replicated the relevant experiments here for clarity. Because UNI was pre-trained on TCGA and Camelyon data—raising the possibility of data leakage—we conducted additional experiments using the independent EBRAINS dataset. The results of these experiments are shown below.

Model	Accuracy	F1	Δ Accuracy	Δ F1
ABMIL	65.4	68.7	—	—
+MIL Dropout	70.4	73.2	+5.0	+4.5
TransMIL	67.4	74.4	—	—
+MIL Dropout	71.3	79.4	+3.9	+5.0
DSMIL	67.4	74.4	—	—
+MIL Dropout	69.3	76.0	+1.9	+1.6
DTFD	53.4	63.6	—	—
+MIL Dropout	64.8	69.8	+11.4	+6.2

Although our MIL dropout can still offer improvements when the better patch features are implemented, the performance gains may be less substantial (about 1.5 to 0.5). Nevertheless, as demonstrated by the experiments, foundation model features often do not perform optimally on private datasets, making our approach particularly suitable for such scenarios.

C.2. Additional Experiments

Currently, our experiments primarily utilize shallow feature extractors. We also attempted to integrate MIL dropout into the classifier (add more linear layers for integrating MIL Dropout) following the MIL module (actually the same thing we felt), and the performance remained largely unchanged. Furthermore, integrating MIL dropout into a transformer architecture was unsuccessful—likely because MIL dropout interferes with layer normalization.

MIL dropout fails to converge for integration into transformer blocks.

C.3. Extra Comparison

Our MIL dropout can be seamlessly integrated with other MIL methods, as it is orthogonal to them. We conducted experiments using our patch features and observed performance improvements, further demonstrating the flexibility and

Table 4. Model (running 5 times on CAMELYON16 Imagenet)

Model	Accuracy	F1	AUC
TransMIL	84.7	83.3	86.5
+MIL Dropout in shallow extractor	86.0	84.9	89.4
+MIL Dropout in classifier	87.6	82.5	88.7
+MIL Dropout in Transformer	64.5	52.5	52.5

effectiveness of our dropout.

Table 5. Model (running 3 times on CAMELYON16 Imagenet), [(Zheng et al., 2024; Qu et al., 2023)]The full open-source code has not been released, and we believe our MIL dropout can be integrated into these methods.

Model	Accuracy	F1	AUC	Δ Accuracy	Δ AUC
PAM (Huang et al., 2024)	85.0	83.2	86.7	—	—
+MIL Dropout	86.2	84.3	87.7	+1.2	+1.0
DPSF (Zheng et al., 2024)	—	—	—	—	—
(Qu et al., 2023)	—	—	—	—	—

C.4. Questions/Suggestions

C.4.1. Experimental setup:

Figure 5(c) reports ablation results on two datasets using ImageNet-pretrained features. All experiments in Table 2 use the optimal values of K and G identified in Figure 5(c).

C.4.2. Performance degradation with high drop rate:

- Dropping a large proportion of instances inevitably degrades performance — not only for our proposed MIL-Dropout but also for standard/vanilla dropout.
- When most features or instances are zeroed out, the model struggles to distinguish positive from negative instances (ReLU outputs zero for both), while the bag-level label remains positive. This prevents effective learning.
- In our Camelyon16 experiments, aggressive dropout led to the failure of convergence for all methods; See the following results.

Table 6. Dropout Comparison on Camelyon16

Model	Accuracy
ABMIL	86.3
+Dropout(p=0.4)	66.2
+Dropout(p=0.8)	53.4
+Dropout1(D=0.4)	63.1
+Dropout1(D=0.8)	51.3
+Our MIL-Dropout (k = 100, G = 5)	59.9
+Our MIL-Dropout (k = 400, G = 5)	55.7

C.4.3. Optimal dropout threshold:

- By restricting dropout to at most 10% of the average instance count (either via Top-K selection or a dropout probability $p = 0.1$), we observe consistent performance improvements across datasets.

C.5. Extra WSI task/datasets Evaluation:

TASK & DATASETS:

We evaluated our method on survival prediction using two TCGA datasets (LUAD and BRCA), following the protocols of (Jaume et al., 2024; Song et al., 2024).

C.5.1. EVALUATION METRIC:

Concordance index (C-index) with %. The K = 10, G = 10 for MIL-Dropout.

C.5.2. RESULTS:

Table 7. Survival Prediction Results on TCGA-LUAD and TCGA-BRCA

Model	**TCGA-LUAD**	**+MIL-Dropout**	**Δ**	**TCGA-BRCA**	**+MIL-Dropout**	**Δ**
ABMIL	65.7	67.7	+2.0	72.8	75.2	+2.4
DSMIL	61.4	63.4	+2.0	68.8	72.1	+3.3
TransMIL	64.3	67.2	+2.9	72.1	74.7	+2.6
DTFD (AFS)	62.0	65.3	+3.3	71.6	74.6	+2.5
DTFD (MaxS)	65.9	68.8	+2.4	72.8	75.7	+2.9

These results demonstrate that our MIL Dropout consistently improves all baseline methods on WSI-based survival prediction.