# Attention-Challenging Multiple Instance Learning for Whole Slide Image Classification

Yunlong Zhang[1,2] Honglin Li[1,2] Yunxuan Sun[1,2] Sunyi Zheng[2] Chenglu Zhu[2] Lin Yang[2*]
[1] Zhejiang University [2] Westlake University
{zhangyunlong, yanglin}@westlake.edu.cn

## Abstract

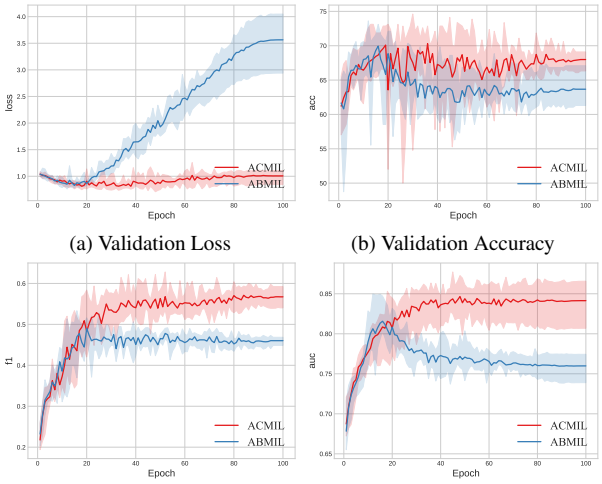
*Overfitting remains a significant challenge in the application of Multiple Instance Learning (MIL) methods for Whole Slide Image (WSI) analysis. Visualizing heatmaps reveals that current MIL methods focus on a subset of predictive instances, hindering effective model generalization. To tackle this, we propose Attention-Challenging MIL (ACMIL), aimed at forcing the attention mechanism to capture more challenging predictive instances. ACMIL incorporates two techniques, Multiple Branch Attention (MBA) to capture richer predictive instances and Stochastic Top-K Instance Masking (STKIM) to suppress simple predictive instances. Evaluation on three WSI datasets outperforms state-of-the-art methods. Additionally, through heatmap visualization, UMAP visualization, and attention value statistics, this paper comprehensively illustrates ACMIL's effectiveness in overcoming the overfitting challenge. The source code is available at https://github.com/dazhangyu123/ACMIL.*

## 1. Introduction

Whole slide image (WSI) analysis is a critical undertaking in digital pathology, aiming to extract valuable information from high-resolution scanned images for precise diagnosis [21,31,48,51], prognosis [9,30,54,60], and treatment planning [11, 33, 35, 38, 39] of diseases. In recent years, multiple instance learning (MIL) [1, 16, 36] has emerged as a promising approach for WSI analysis, treating each WSI as a "bag" and its extracted small patches as "instances" within the bag, thus enabling efficient classification of WSIs through assigning a single label to the entire slide.

Overfitting is a significant challenge in utilizing MIL methods for WSI analysis [32,45,56]. Firstly, common WSI datasets exhibit the intrinsic characteristics of limited data scale, ultra-high resolutions, and severe class imbalance, which makes overfitting more likely [2]. These datasets often consist of a relatively small number of slides, typically in the hundreds, each containing numerous patches, ranging from thousands to tens of thousands, with a relatively low proportion of positive cases. Secondly, these datasets are susceptible to data bias caused by variations in tissue preparations, staining protocols, and digital scanning methods [32, 57]. This bias can mislead machine learning models into making classifications based on spurious features rather than the underlying biological characteristics [17, 18]. As illustrated in Fig. 1, ABMIL [25], one of the most famous MIL methods, shows severe overfitting since loss drastically increases and validation metrics significantly decreases as the training processes.

Numerous efforts [14, 23, 24, 29, 34, 40, 43, 45, 46, 55] have been made to mitigate the overfitting challenge. These approaches have proven effective in improving evalua-




(a) Validation Loss (b) Validation Accuracy

(c) Validation F1-score (d) Validation AUROC

Figure 1. Comparison of performance between ABMIL [25] and our ACMIL on the LBC validation set throughout the training process. ABMIL displays pronounced signs of overfitting, as indicated by a significant increase in validation loss and a decline in the other three evaluation metrics. Conversely, ACMIL effectively mitigates the overfitting issue.

*Corresponding Author.

tion metrics on standard benchmarks. On the other hand, heatmap visualization, a valuable tool, is commonly used to enhance model interpretability [25]. In fact, heatmaps also play a critical role in aggregating instance features into bag features, which directly impact the final predictions. Existing studies have primarily focused on the interpretability aspect of heatmaps and have rarely explored their connection to the problem of overfitting.

In this paper, we study the overfitting challenge resorting to the heatmap. Heatmap shows that existing attention mechanisms predominantly concentrate on a subset of predictive instances while disregarding the remaining predictive ones (refer Fig. 5). However, recent studies [14, 24, 46] have shown that models trained solely on simple predictive features may struggle to generalize to out-of-distribution data. In conclusion, *the excessive concentration of attention values in heatmap is closely link to the overfitting.*

To mitigate the excessive concentration of attention values, we start with two additional analyses for the heatmap. Firstly, *there are various patterns among predictive instances, and existing attention mechanisms only can capture a part of them.* To solve this, we introduce the Multiple Branch Attention (MBA) method. MBA utilizes multiple attention branches, where each branch is responsible for capturing instances with the specific pattern, ensuring that richer predictive instances contribute to the final prediction. Secondly, in the existing attention mechanism, *a few of instances will occupy majority attention.* To mitigate it, we propose Stochastic Top-K Instance Masking (STKIM). STKIM randomly masks a portion of instances with top attention values at each iteration and then assigning their attention values to remaining instances. Combining both of them, we propose the Attention-Challenging MIL (ACMIL) framework, aiming to force attention mechanisms to capture more challenging predictive instances.

We conduct experiments on three WSI datasets, i.e., Camelyon16, BRACS, and our in-house LBC dataset. Experimental results demonstrate that our ACMIL significantly outperforms existing SOTA methods. We also present substantial experimental results, including heatmap visualization, umap visualization, and attention value statistics, to demonstrate the effectiveness of ACMIL in combatting overfitting.

## 2. Related Work

### 2.1. Combating Overfitting in WSI Analysis

In the domain of WSI analysis, combating the challenge of overfitting has received substantial attention. Several approaches have emerged, each with distinct focuses on improvement.

Some efforts have concentrated on enhancing the quality of feature representations. Early studies (e.g., [6, 25, 43]) relied on backbones pretrained on the ImageNet dataset. However, the substantial domain gap between natural and pathological images hindered representation quality. Recent works (e.g., [13, 22, 34, 50]) have addressed this by emphasizing Self-Supervised Learning (SSL) to learn patch-level feature representations. In addition, efforts such as the work by Chen et al. [8] leveraged hierarchical SSL for high-resolution image representations. Further, studies by Li et al. [29] and Wang et al. [49] demonstrated that fine-tuning the pretrained encoder is essential for acquiring task-specific information.

Other efforts have centered on improving attention mechanisms. For instance, ABMIL [25] introduced gated attention for predicting attention scores. DSMIL [28] used instance-to-instance distances to calculate attention scores. Moreover, some studies explored recurrent neural networks [6], self-attention layers [41, 43, 53], and graph neural networks [19, 58] to model inter-instance relationships.

Further strategies have aimed at refining the training process. For example, DTFD-MIL [56] introduced pseudo-bags to expand bag counts and employed a double-tier MIL framework. IBMIL [32] leveraged interventional training via backdoor adjustment to mitigate bias from contextual priors. IPS [4], Zoom-In Network [27], and RankMix [10] focused on generated bag representations by aggregating the representations of most salient patches. MHIM-MIL [45] masked instances with high attention scores, emphasizing the significance of hard instances during model training. Our work may share the similar motivation with these work. Nevertheless, our solution is based on the observation and analysis of heatmap, whereas the existing methods more rely on the intuition.

Our work is done independently and concurrently with MHIM-MIL [45].

### 2.2. Heatmap Visualization in WSI analysis

Heatmaps are valuable tools for assessing the interpretability of MIL methods [25]. Many existing methods [13, 25, 28, 29, 34, 43, 45, 52, 55] rely on heatmap visualizations to showcase their enhanced interpretability. It's important to recognize that heatmaps play a central role in aggregating instance features into bag-level features, significantly impacting the model's generalization capacity. This paper stands out by pioneering the use of heatmap as a tool for analyzing the overfitting challenge.

## 3. Method

Based on the ABMIL (detailed in Sec. 3.1), we present ACMIL to alleviate the overfitting problem, which is built on two components: Mutiple Branch Attention (MBA) and Stochastic Top-K Instance Masking (STKIM). We describe the details of two components in the Sec. 3.2 and 3.3, respectively.

### 3.1. ABMIL for WSI Analysis

In the binary MIL classification problem [6], a bag of instance, $\boldsymbol{X} = \{\boldsymbol{x}_n\}_{n=1}^N$, is associated with a single bag label, $\boldsymbol{Y}$. Each instance, $x_n$, is associated with a single binary label, $y_n$, which remains unknown during training. The assumption behind the MIL can be written as:

$$\boldsymbol{Y} = \begin{cases} 0, & \text{iff } \sum_{n=1}^N y_n = 0 \\ 1, & \text{otherwise} \end{cases} \tag{1}$$

In the ABMIL [25], the multiple instance learning is modeled by a three step process. i) Instance transformation into a low-dimensional embedding through neural networks: $\boldsymbol{h}_n = f(\boldsymbol{x}_n)$. ii) Aggregation of all instance embeddings into the bag-level representation using an attention operator. Specifically, this operation is defined as:

$$\boldsymbol{z} = \sum_{n=1}^{N} a_n \boldsymbol{h}_n, \tag{2}$$

Here, $a_n = \sigma(\boldsymbol{h}_n)$ represents the attention values for $n$-th instance, $\boldsymbol{h}_n$. In the case of ABMIL, a gated attention (GA) mechanism [12] is adopted:

$$\sigma(\boldsymbol{h}_n) = \frac{\exp\{\boldsymbol{w}^{\mathrm{T}}(\tanh(\boldsymbol{V}_1\boldsymbol{h}_n) \odot \mathrm{sigm}(\boldsymbol{V}_2\boldsymbol{h}_n))\}}{\sum_{j=1}^N \exp\{\boldsymbol{w}^{\mathrm{T}}(\tanh(\boldsymbol{V}_1\boldsymbol{h}_j) \odot \mathrm{sigm}(\boldsymbol{V}_2\boldsymbol{h}_j))\}}, \tag{3}$$

iii) The bag prediction is generated based on the aggregated bag embedding: $\hat{\boldsymbol{Y}} = g(\boldsymbol{z})$.

### 3.2. Mutiple Branch Attention

**Motivation.** It is challenging to capture all predictive instances using a single attention operator (see Fig. 2). Firstly, there are variations in patterns among predictive patches due to the texture difference. Then, DNNs exhibit laziness in finding more challenging features when capturing the simple predictive patterns is enough to minimize the training loss [17, 18]. To capture more predictive instances, we design the MBA that consists of multiple attention branches. Each branch is responsible for capturing instances with the specific pattern, ensuring that more predictive patterns contribute to the final prediction.

As depicted in Fig. 3 top view, the MBA firstly captures $M$ predictive patterns and then aggregates their embeddings to make prediction. Each pattern is captured by an attention branch. To ensure the predictive semantic and semantic diversity between patterns, the semantic regularization and diversity regularization are proposed, respectively. Firstly, to ensure capturing predictive patterns, the semantic regularization is accomplished by hanging a MLP layer behind each pattern embedding, equipping with the cross entropy loss function:

$$\mathcal{L}_p = -\frac{1}{M}\sum_{i=1}^{M} \boldsymbol{Y}\log\hat{\boldsymbol{Y}}_i + (1-\boldsymbol{Y})\log\left(1-\hat{\boldsymbol{Y}}_i\right) \tag{4}$$

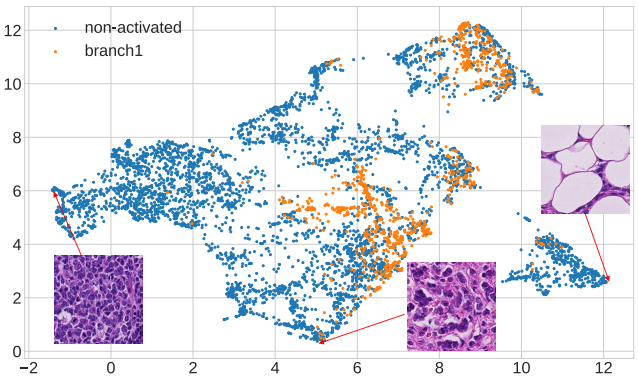



Figure 2. Motivation of MBA. UMAP visualization [37] of instance features in the tumor region of Camelyon16 'test_113' case. *There are various patterns/clusters among predictive instances, and relying on one single branch can only capture a part of clusters.* Three instances are selected to exhibits their texture differences.

where $\hat{\boldsymbol{Y}}_i = g_i(\boldsymbol{z}_i)$ is the prediction based on $i$-th pattern embedding, $\boldsymbol{z}_i$. However, only equipping with cross entropy loss may learn the similar patterns and cannot dig out more predictive information. To tackle this issue, we further introduce a diversity loss as follows:

$$\mathcal{L}_d = \frac{2}{M(M-1)}\sum_{i=1}^{M}\sum_{j=i+1}^{M}\cos(\boldsymbol{a}_i, \boldsymbol{a}_j) \tag{5}$$

where $\boldsymbol{a}_i$ consists of all attention values of $i$-th pattern, $\boldsymbol{a}_i = \{a_{i1}, \cdots, a_{iN}\}$, also named heatmap as custom. The $\cos(\cdot)$ function is used to measure the similarity of the heatmaps between branches. By diversifying the heatmaps, the embedding of each branches can concentrate on different predictive patterns.

To aggregate the captured patterns to make prediction, the average of heatmaps is utilized as the heatmap of the whole bag:

$$\boldsymbol{a} = \frac{1}{M}\sum_{i=1}^{M}\boldsymbol{a}_i \tag{6}$$

where $\boldsymbol{a}$ is the heatmap of the whole bag, with dimension of $N$. Then, the bag embedding can be got by aggregating the instance features using averaged heatmap $\boldsymbol{a}$. Moreover, since $\sum_{n=1}^N(\frac{1}{M}\sum_{i=1}^M a_{in})\boldsymbol{h}_n = \frac{1}{M}\sum_{i=1}^M(\sum_{n=1}^N a_{in}\boldsymbol{h}_n)$, the bag embedding also can be formulated by applying mean pooling operator to pattern embeddings. The top view of Fig. 3 adopts the latter formulation for brevity. The loss function for the bag classifier is defined as:

$$\mathcal{L}_b = -\boldsymbol{Y}\log\hat{\boldsymbol{Y}} + (1-\boldsymbol{Y})\log\left(1-\hat{\boldsymbol{Y}}\right) \tag{7}$$

Finally, the overall loss function for the ACMIL can be written as the combination of three loss terms defined in Eq. 4, 5 and 7,

$$\mathcal{L} = \mathcal{L}_b + \mathcal{L}_p + \mathcal{L}_d \tag{8}$$

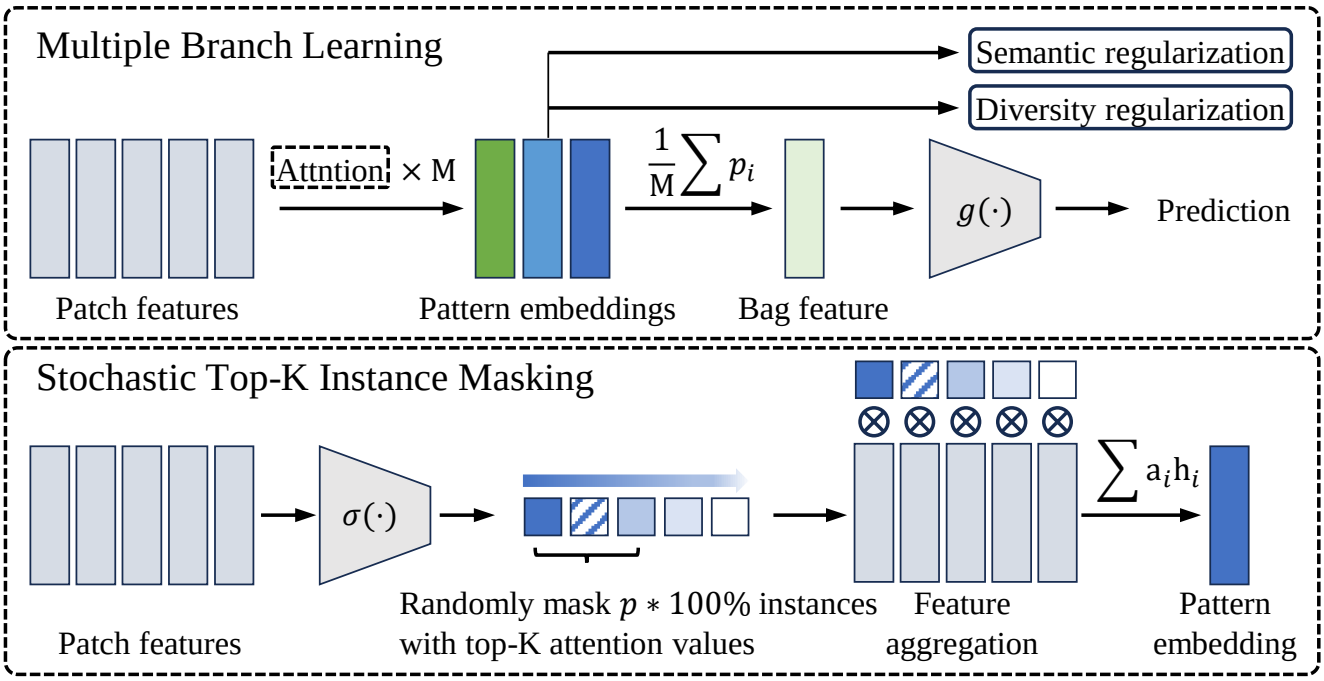



Figure 3. Overview of the proposed MBA (top view) and STKIM (bottom view). In the MBA, $M$ predictive patterns are extracted from patch features using the attention operator regularized by semantic and diversity regularization terms. Then, a mean operator is performed to generate the bag feature, used for bag-level prediction. In the STKIM, $p \times 100\%$ instances with top-K attention values are masked randomly in the attention operator.

**Discussion.** It's worth noting that when $M$ is set to 1, the MBA essentially equals to the feature aggregation process of ABMIL, which can only discern one single pattern. Thus, the MBA can be viewed as an extension for ABMIL used for capturing more diverse predictive patterns. Otherwise, We emphasize the different goals of using parallel attention modules in our MBA and the recent work, DTFD-MIL [56]. DTFD-MIL aims to augment the bag by randomly split the bag into several sub-bags, and parallel attention modules are used to capture the discriminative instances in each sub-bags. For the MBA, the parallel attention modules are used to capture distinct predictive patterns from the whole bag.

### 3.3. Stochastic Top-K Instance Masking

**Motivation.** As depicted in Fig. 4, we find that a few instances may occupy majority attention in ABMIL. For example, The sum of top-10 attention values is larger than 0.85 on all three datasets. To alleviate this issue, we present STKIM that randomly masks a portion of instances with top-K attention values and then assigning their attention values to remaining instances.

As depicted in Fig. 3 bottom view, STKIM introduces a masking operation into the attention mechanism, before

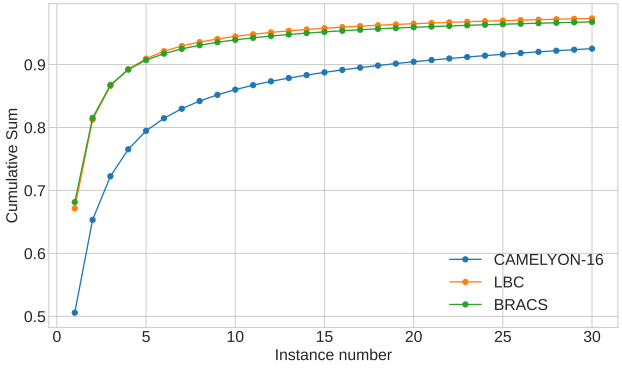



Figure 4. Motivation of STKIM. Accumulation of top-k attention values. *Instances with top-k attention values occupy majority attention.* Results are derived from features extracted through supervised pretraining.

feature aggregation and after attention values generation. The primary objective is to mask a few of the most predictive instances, redirecting more attention toward subordinate instances. A straightforward solution to achieve this is to mask the top-K instances. However, this method poses certain challenges. It can result in the loss of information associated with key instances, which are crucial for dis-

crimination. Furthermore, it might lead to a statistical mismatch between the feature representations before and after discarding these key instances. To address these issues, we draw inspiration from dropout techniques [44] commonly used in neural networks. Our proposed solution employs stochastic masking for instance features with top-K attention values.

Specifically, we begin by sorting all attention values from highest to lowest. Subsequently, we randomly set the attention values of the top-K instances to 0, with a probability of $p$. This process can be formulated as:

$$a_n = \begin{cases} 0, & \text{with probability } p \text{ and within top-K values} \\ a_n, & \text{otherwise} \end{cases} \tag{9}$$

where $p$ and $K$ are two hyperparameters that control the intensity of masking. Notably, following the STKIM, we will rescale the attention values by $a_n \rightarrow \frac{1}{\sum_{n=1}^{N} a_n} a_n$ to ensure $\sum_{n=1}^{N} a_n = 1$.

**Discussion.** There are notable technical distinctions between STKIM and MHIM-MIL [45]. Firstly, MHIM-MIL adopts a two-stage training procedure where the model trained in the second stage is initialized using the best checkpoint obtained in the first stage. In contrast, STKIM is a one-stage framework that doesn't require a pre-trained checkpoint beforehand, providing better scalability. Secondly, MHIM-MIL employs instance masking on a momentum teacher model, using the masked instances to train a student model. This involves two forward propagations for calculating attention values and producing bag predictions. On the other hand, STKIM utilizes a single model and necessitates only one forward propagation, resulting in faster execution compared to MHIM-MIL. Thirdly, MHIM-MIL incorporates three masking strategies and introduces five masking hyperparameters, which can be a complex and time-consuming trial-and-error process for reaching the optimal performance. In contrast, STKIM primarily involves two hyperparameters, $p$ and $K$. Moreover, our ablation study in Appendix Sec. D.1 demonstrates that setting $p = 0.6$ and $K = 10$ achieves near-optimal performance across all datasets, significantly reducing the effort and time associated with trial-and-error tuning. Overall, *STKIM has better scalability, faster execution, and reduced trial-and-error costs.* In Appendix Sec. D.5, experimental results verify that our STKIM has the faster execution, without compromising on performance.

# 4. Experiments

## 4.1. Experimental Details

**Datasets and Evaluation Metrics.** The performance of ACMIL is evaluated on two public WSI datasets, i.e., Camelyon16 [3] and BRACS [5], and one private benchmark, LBC. Camelyon16 dataset consists of 400 WSIs in total, including 270 for training and 130 for testing. Following [28, 56], we further randomly split the training and validation sets from the the official training set with a ratio of 9:1. We do not resplit BRACS dataset as it has been officially split to 395 of training set, 65 of validation set, and 87 of test set. We follow the challenge for a 3-class WSI classification: benign tumor, atypical tumor, and malignant. The liquid-based cytology (LBC) dataset collected 1,989 WSIs and include 4 classes, i.e., Negative, ASC-US, LSIL, and ASC-H/HSIL. We randomly split the whole dataset to training, validation, and test sets with the ratio of 6:2:2. Following [29], macro-AUC and macro-F1 scores are reported since all the three datasets are class imbalanced. Each of the main experiments is performed five times with random parameter initializations, and the average classification performance and standard deviation are reported.

**Baselines.** We systematically assess the efficacy of our ACMIL approach by benchmarking it against conventional MIL pooling strategies, Max-pooling and Mean-pooling, as well as contemporary attention-based techniques such as ABMIL [25], DSMIL [28], TransMIL [43], CLAM-SB [34], DTFD-MIL [56], MHIM-MIL [45], and IBMIL [32]. In pursuit of a comprehensive comparison across diverse aggregation operators, we utilize two distinct sets of features derived from ResNet-18 pretrained on the ImageNet dataset [20] and ViT-S/16 pretrained using DINO [7] on a substantial collection of 36,666 WSIs [26]. The results of all other methods are reproduced using the official code they provide under the same settings.

**Implementation Details.** Implementation Details are placed in Appendix Sec. C.

## 4.2. Performance Evaluation against SOTA

Table 1 provides a thorough comparison of performance between ACMIL and existing MIL methods. This evaluation spans three diverse datasets, involves two different choices for pretraining methods, and employs two crucial evaluation metrics, resulting in a comprehensive assessment with a total of 12 terms.

Considering the overall performance, ACMIL consistently outshines existing methods. It secures the top position in 10 out of the 12 metrics and holds the second position in the remaining 2 metrics. Specifically, for the Camelyon16, ACMIL achieves outstanding results using ResNet-18 pretrained on ImageNet embeddings, surpassing the runner-up by 2.1% and 2.6% in terms of F1-score and AUC, respectively. On the other hand, with ViT-S/16 SSL pretrained embeddings, existing attention-based MIL methods exhibit remarkable performance, boasting F1-scores and AUC values exceeding 0.9. Notably, ACMIL achieves comparable performance with the former best-performing method, DTFD-MIL, in this setup.

For the BRACS, ACMIL demonstrates a substantial lead when utilizing ViT-S/16 SSL pretrained embeddings, sur-

Table 1. The performance of different MIL approaches across three datasets, two pretrained methods, and two evaluation metrics. The most superior performance is highlighted in **bold**, while the second-best performance is indicated by underlining.

| | Method \ Performance | CAMELYON-16 | | BRACS | | LBC | |
|---|---|---|---|---|---|---|---|
| | | F1-score | AUC | F1-score | AUC | F1-score | AUC |
| ResNet-18 ImageNet pretrained | Max-pooling | 0.582±0.170 | 0.620±0.155 | 0.489±0.047 | 0.738±0.014 | 0.476±0.033 | 0.775±0.010 |
| | Mean-pooling | 0.592±0.026 | 0.597±0.033 | 0.484±0.029 | 0.685±0.011 | 0.511±0.022 | 0.797±0.011 |
| | Clam-SB | 0.742±0.024 | 0.763±0.049 | 0.521±0.046 | 0.750±0.039 | 0.514±0.024 | 0.805±0.017 |
| | TransMIL | 0.643±0.088 | 0.706±0.076 | 0.444±0.040 | 0.732±0.043 | 0.385±0.013 | 0.693±0.027 |
| | DSMIL | 0.736±0.028 | 0.773±0.034 | 0.511±0.052 | 0.751±0.028 | 0.458±0.029 | 0.766±0.023 |
| | DTFD-MIL | 0.758±0.051 | 0.815±0.063 | 0.469±0.016 | 0.717±0.032 | 0.473±0.021 | 0.776±0.021 |
| | IBMIL | 0.777±0.009 | 0.799±0.050 | 0.510±0.043 | 0.726±0.034 | 0.489±0.017 | 0.791±0.021 |
| | MHIM-MIL | 0.752±0.034 | 0.772±0.026 | 0.511±0.022 | **0.774±0.021** | 0.543±0.037 | 0.816±0.009 |
| | ABMIL | 0.757±0.020 | 0.790±0.027 | 0.523±0.028 | 0.723±0.035 | 0.465±0.040 | 0.798±0.013 |
| | ACMIL(ours) | **0.798±0.029** | **0.841±0.030** | **0.552±0.048** | 0.754±0.008 | **0.546±0.028** | **0.821±0.015** |
| ViT-S/16 SSL pretrained | Max-pooling | 0.903±0.054 | 0.956±0.029 | 0.596±0.029 | 0.823±0.033 | 0.590±0.043 | 0.829±0.023 |
| | Mean-pooling | 0.577±0.057 | 0.569±0.081 | 0.522±0.038 | 0.739±0.007 | 0.559±0.024 | 0.827±0.012 |
| | Clam-SB | 0.925±0.035 | 0.969±0.024 | 0.631±0.034 | 0.863±0.005 | 0.617±0.022 | 0.865±0.018 |
| | TransMIL | 0.922±0.019 | 0.943±0.009 | 0.631±0.030 | 0.841±0.006 | 0.539±0.028 | 0.805±0.010 |
| | DSMIL | 0.943±0.007 | 0.966±0.009 | 0.577±0.028 | 0.816±0.028 | 0.562±0.028 | 0.820±0.033 |
| | DTFD-MIL | 0.948±0.007 | **0.980±0.011** | 0.612±0.080 | 0.870±0.022 | 0.612±0.034 | 0.842±0.010 |
| | IBMIL | 0.912±0.034 | 0.954±0.022 | 0.645±0.041 | 0.871±0.014 | 0.604±0.032 | 0.834±0.014 |
| | MHIM-MIL | 0.932±0.024 | 0.970±0.037 | 0.625±0.060 | 0.865±0.017 | 0.658±0.041 | 0.872±0.022 |
| | ABMIL | 0.914±0.031 | 0.945±0.027 | 0.680±0.051 | 0.866±0.029 | 0.595±0.036 | 0.831±0.022 |
| | ACMIL(ours) | **0.954±0.012** | 0.974±0.012 | **0.722±0.030** | **0.888±0.010** | **0.662±0.043** | **0.901±0.011** |

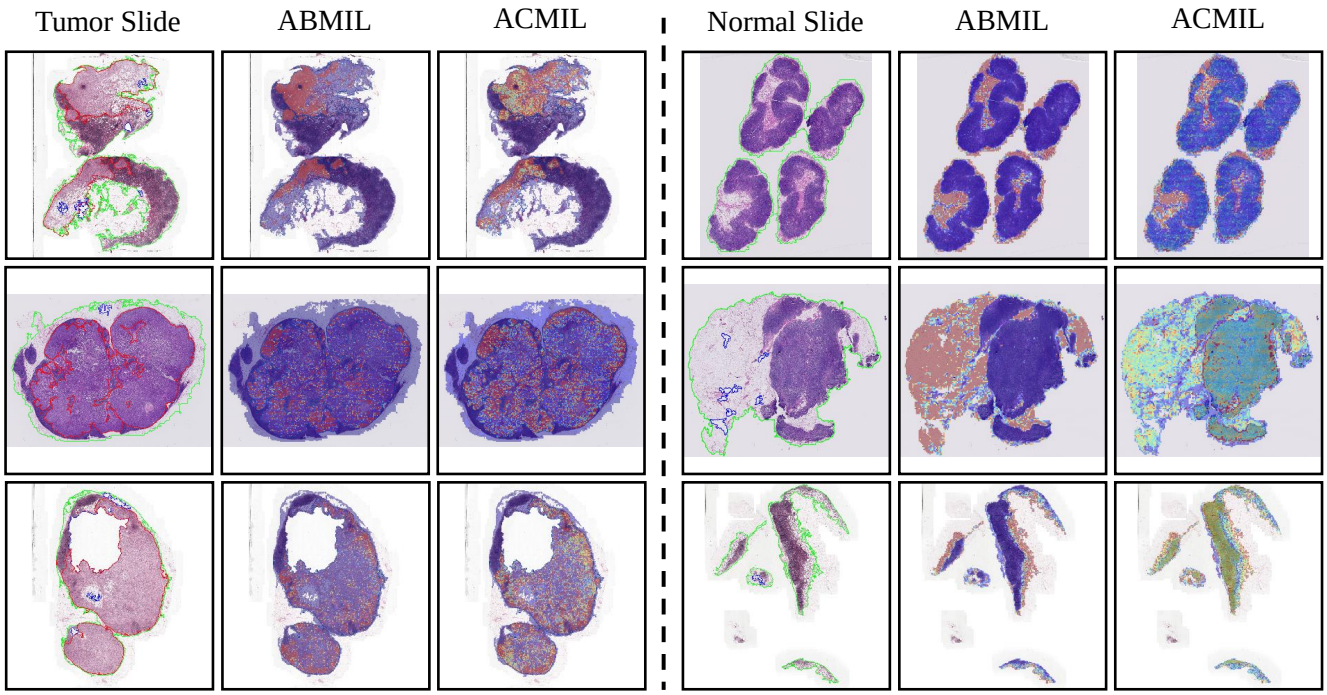
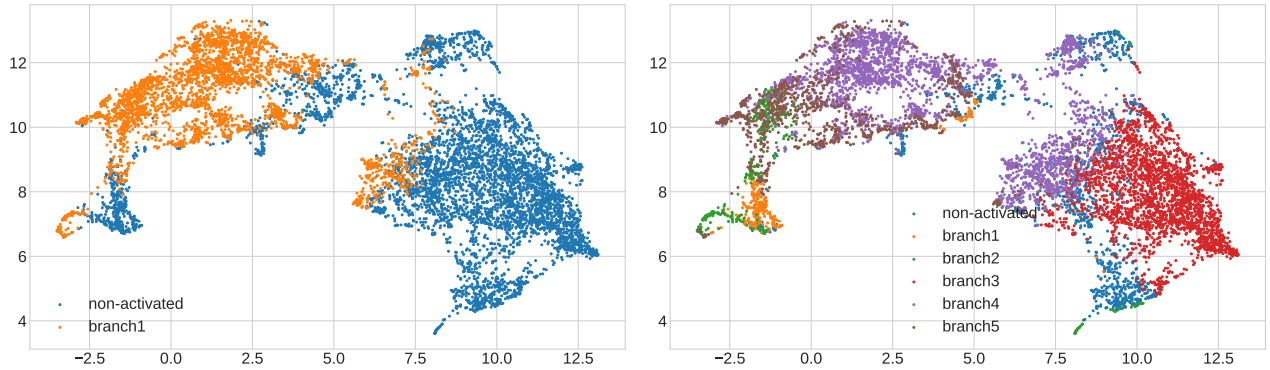



Figure 5. Heatmap visualization of WSI examples produced by ABMIL [25] (baseline) and our ACMIL. The left part shows three tumor WSIs come from Camelyon16 dataset, and its tumor region is delineated by the red line. ACMIL generates attention values that cover a more extensive portion of the tumor region compared to ABMIL. The right part shows three normal WSIs come from Camelyon16 dataset. ABMIL primarily focuses on a part of tissue such as adipose, while ACMIL extends its attention to the more normal tissues.




(a) ABMIL (b) ABMIL with MBA

Figure 6. UMAP visualization of instance features in the tumor region of the Camelyon16 'test_090' case. The tumor instances display distinct patterns, posing a challenge for a single branch to capture all of them. As a result, ABMIL overlooks the right pattern/cluster. In contrast, The varying branch in MBA capture patterns separately, and combining them enable the activation of more patterns. An instance is considered active when its attention value surpasses $\frac{1}{N}$.

passing the second-best performance by margins of 7.7% and 1.7% in F1-score and AUC, respectively. Moreover, when employing ResNet-18 pretrained on ImageNet embeddings, ACMIL achieves comparable performance with the previously top-performing method, MHIM-MIL. For the LBC, ACMIL stands out significantly among the other methods across all four metrics.

### 4.3. Heatmap Visualization

Fig. 5 presents heatmap visualizations illustrating examples of our approach's performance in comparison to the baseline method, ABMIL [25]. Three tumor slides (left part) and three normal slides (right part) are selected to showcase the heatmap differences. Otherwise, due to the space limitation, we present more visualization at Appendix for further insights.

For the tumor slides, ABMIL tends to concentrate its attention on only a fraction of the tumor regions, potentially overlooking other significant areas. In contrast, ACMIL allocates attention across a wider spectrum of tumor regions, resulting in better alignment with expert annotations. For the normal slides, ABMIL predominantly focuses on specific tissue types, such as adipose tissue. This will lead to misinterpretation that only the adipose tissue is the normal tissue and other normal regions are uncorrelated to the WSI label. On the other hand, ACMIL effectively distributes attention values to encompass all normal regions, ensuring all regions are correlated for the WSI label. This approach closely mimics human intuition and satisfies the definition of the MIL formulation.

### 4.4. Further Analysis

**MBA can capture diverse predictive patterns.** We employed UMAP [37] to visualize instance features within the tumor region of the Camelyon16 'test_090' case. In Fig. 6a, it's evident that the tumor instances exhibit two primary patterns. However, ABMIL primarily activates the left pattern (colored orange) and neglects the right one. On the other hand, as demonstrated in Fig. 6b, MBA's various branches (branch1, branch2, branch3, and branch5) collectively capture the substructures of the left pattern, while branch4 specifically captures the right pattern. Combining all branches can capture more comprehensive patterns.

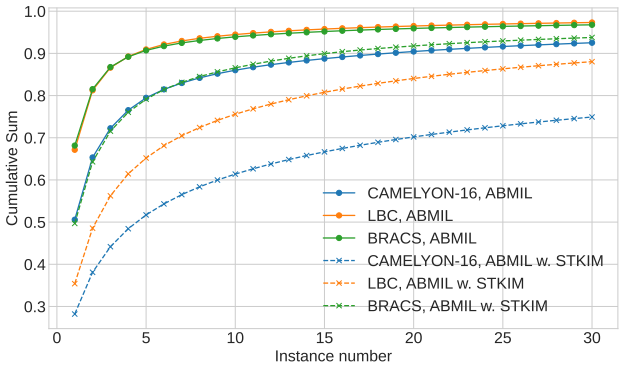
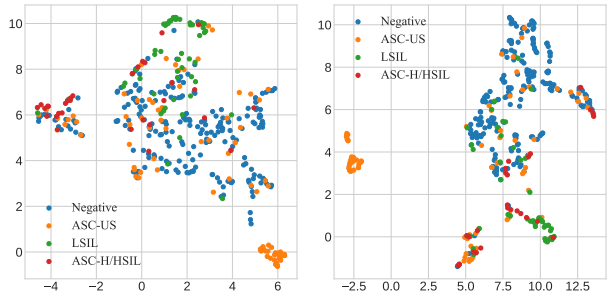



Figure 7. Comparison of accumulative sum of top-k attention values with STKIM and without STKIM. The use of STKIM helps alleviate the issue of excessive concentration of attention values within the top-K range..

**STKIM can suppress overly concentrated attention values.** Fig. 7 illustrates a comparison of the cumulative sum of top-K attention values with and without STKIM. The plot clearly demonstrates that the use of STKIM helps mitigate the scenario where top-K attention values excessively dominate in the attention mechanism. This effect is particularly pronounced for the Camelyon16 dataset, where the cumulative sum of the top-10 values decreases from 0.87 to 0.6.

**ACMIL can learn more discriminative bag features.** We employ UMAP [37] to visualize bag features from the LBC test set, as illustrated in Fig. 8a and 8b. This visualization demonstrates that our ACMIL is capable of learning more




(a) ABMIL(V-measure=0.224) (b) ACMIL(V-measure=0.316)

Figure 8. UMAP visualization [37] of bag features for LBC test set. ACMIL learns more discriminative features than ABMIL effectively separating 'LSIL' and 'ASC-H/HSIL' features from the 'Negative' class. This improved feature separation is corroborated by the V-measure score [42], a clustering metric that considers both the homogeneity and completeness of the clusters.

Table 2. Performance comparison between ACMIL with (w.) and without (w/o.) T-STKIM. The Gap column reports the performance difference between with and without T-STKIM. T-STKIM means using the STKIM at test phase. Using STKIM at test phase slightly reduces its performance.

ViT-S/16 SSL pretrained

| Dataset | Metric | w. T-STKIM | w/o. T-STKIM | Gap(%) |
|---|---|---|---|---|
| Camelyon | F1-score | 0.927±0.057 | 0.954±0.012 | +2.7 |
| | AUC | 0.967±0.017 | 0.974±0.012 | +0.7 |
| BRACS | F1-score | 0.697±0.033 | 0.722±0.030 | +2.5 |
| | AUC | 0.875±0.012 | 0.888±0.010 | +1.3 |
| LBC | F1-score | 0.637±0.034 | 0.662±0.043 | +2.5 |
| | AUC | 0.878±0.012 | 0.901±0.011 | +2.3 |

ResNet-18 Imagenet pretrained

| Dataset | Metric | w. T-STKIM | w/o. T-STKIM | Gap(%) |
|---|---|---|---|---|
| Camelyon | F1-score | 0.780±0.026 | 0.798±0.029 | +1.8 |
| | AUC | 0.837±0.028 | 0.841±0.030 | +0.4 |
| BRACS | F1-score | 0.566±0.054 | 0.552±0.048 | -1.4 |
| | AUC | 0.750±0.021 | 0.754±0.008 | +0.4 |
| LBC | F1-score | 0.535±0.027 | 0.546±0.028 | +1.1 |
| | AUC | 0.808±0.019 | 0.821±0.015 | +1.3 |

discriminative features compared to ABMIL. Specifically, it successfully separates the LSIL and ASC-H/HSIL clusters from the Negative cluster. To quantitatively assess this clustering performance, we employ the V-measure [42], following the methodology of Li et al. [29] and Diao et al. [15]. ACMIL achieves a V-measure score of 0.316, a significant improvement over ABMIL, which scores 0.224.

**Do we need STKIM at the test phase? Answer is No.** In Tab. 2, we present the outcomes of ACMIL with and without STKIM during the test phase, along with the performance differences between these settings. Across 11 out of 12 evaluation metrics, the version of ACMIL without STKIM during testing outperforms the version with STKIM slightly. This suggests that STKIM is not necessary during the test phase. Consequently, we can draw an analogy between the role of STKIM and masking data augmentation techniques such as cutout [14, 59].

Table 3. Performance comparison between ACMIL with (w.) and without (w/o.) $\mathcal{L}_d$. The Gap column reports the performance difference between without and with $\mathcal{L}_d$. ACMIL without $\mathcal{L}_d$ drastically reduces its performance.

ViT-S/16 SSL pretrained

| Dataset | Metric | w/o. $\mathcal{L}_d$ | w. $\mathcal{L}_d$ | Gap(%) |
|---|---|---|---|---|
| Camelyon | F1-score | 0.901±0.037 | 0.954±0.012 | +5.3 |
| | AUC | 0.943±0.027 | 0.974±0.012 | +3.1 |
| BRACS | F1-score | 0.642±0.046 | 0.722±0.030 | +8.0 |
| | AUC | 0.859±0.020 | 0.888±0.010 | +2.9 |
| LBC | F1-score | 0.603±0.023 | 0.662±0.043 | +5.9 |
| | AUC | 0.837±0.009 | 0.901±0.011 | +6.4 |

ResNet-18 Imagenet pretrained

| Dataset | Metric | w/o. $\mathcal{L}_d$ | w. $\mathcal{L}_d$ | Gap(%) |
|---|---|---|---|---|
| Camelyon | F1-score | 0.747±0.022 | 0.798±0.029 | +5.1 |
| | AUC | 0.796±0.032 | 0.841±0.030 | +5.5 |
| BRACS | F1-score | 0.500±0.031 | 0.552±0.048 | +5.2 |
| | AUC | 0.760±0.026 | 0.754±0.008 | -0.6 |
| LBC | F1-score | 0.532±0.019 | 0.546±0.028 | +1.4 |
| | AUC | 0.809±0.018 | 0.821±0.015 | +1.2 |

**Do we need diversity loss in MBA? Answer is Yes.** In Tab. 3, we present the outcomes of ACMIL with and without $\mathcal{L}_d$, along with the performance differences between these settings. Notably, the last column clearly indicates a significant performance drop for ACMIL without $\mathcal{L}_d$. This emphasizes the crucial role of $\mathcal{L}_d$ in encouraging different branches to acquire distinctive predictive knowledge within the MBA technique.

### 4.5. Ablation Study

Due to space limitation, we place the ablation study in the Appendix Sec. D.1, with the main focus on the effect of hyperparameter $K, p, M$. Note that the effect for ablating MBA and STKIM is also discussed by setting $M = 1$ and $K = 0$, respectively.

## 5. Conclusion

Due to intrinsic properties of WSI, MIL methods have often led to overfitting, limiting their applications. This paper revealed that the overly concentrated attention values in heatmap is closely related to the overfitting. To Address this, we proposes the ACMIL approach, which is underpinned by two novel techniques: MBA and STKIM. Our experimental results on three datasets demonstrate ACMIL significantly surpasses SOTA methods. Moreover, this paper provides comprehensive experiments confirming the effectiveness of ACMIL on alleviating overfitting problem. We hope the our work can inspire future exploration into leveraging heatmaps for comprehensive analysis, encompassing both interpretability and generalization aspects. We also hope that our ACMIL can find valuable applications in a broader spectrum of WSI analysis tasks.

## References


[1] Jaume Amores. Multiple instance classification: Review, taxonomy and comparative study. *Artificial intelligence*, 201:81–105, 2013. 1

[2] Mohammad Mahdi Bejani and Mehdi Ghatee. A systematic review on overfitting control in shallow and deep neural networks. *Artificial Intelligence Review*, pages 1–48, 2021. 1

[3] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes Van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen AWM Van Der Laak, Meyke Hermsen, Quirine F Manson, Maschenka Balkenhol, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*, 318(22):2199–2210, 2017. 5

[4] Benjamin Bergner, Christoph Lippert, and Aravindh Mahendran. Iterative patch selection for high-resolution image recognition. *arXiv preprint arXiv:2210.13007*, 2022. 2, 13

[5] Nadia Brancati, Anna Maria Anniciello, Pushpak Pati, Daniel Riccio, Giosuè Scognamiglio, Guillaume Jaume, Giuseppe De Pietro, Maurizio Di Bonito, Antonio Foncubierta, Gerardo Botti, et al. Bracs: A dataset for breast carcinoma subtyping in h&e histology images. *Database*, 2022:baac093, 2022. 5

[6] 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(8):1301–1309, 2019. 2, 3

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 5

[8] Richard J Chen, Chengkuan Chen, Yicong Li, Tiffany Y Chen, Andrew D Trister, Rahul G Krishnan, and Faisal Mahmood. Scaling vision transformers to gigapixel images via hierarchical self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16144–16155, 2022. 2, 13

[9] Richard J Chen, Ming Y Lu, Wei-Hung Weng, Tiffany Y Chen, Drew FK Williamson, Trevor Manz, Maha Shady, and Faisal Mahmood. Multimodal co-attention transformer for survival prediction in gigapixel whole slide images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4025, 2021. 1

[10] Yuan-Chih Chen and Chun-Shien Lu. Rankmix: Data augmentation for weakly supervised learning of classifying whole slide images with diverse sizes and imbalanced categories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23936–23945, 2023. 2

[11] Toby C Cornish, Ryan E Swapp, and Keith J Kaplan. Whole-slide imaging: routine pathologic diagnosis. *Advances in anatomic pathology*, 19(3):152–159, 2012. 1

[12] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941. PMLR, 2017. 3, 13

[13] Olivier Dehaene, Axel Camara, Olivier Moindrot, Axel de Lavergne, and Pierre Courtiol. Self-supervision closes the gap between weak and strong supervision in histology. *arXiv preprint arXiv:2012.03583*, 2020. 2

[14] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 1, 2, 8

[15] James A Diao, Jason K Wang, Wan Fung Chui, Victoria Mountain, Sai Chowdary Gullapally, Ramprakash Srinivasan, Richard N Mitchell, Benjamin Glass, Sara Hoffman, Sudha K Rao, et al. Human-interpretable image features derived from densely mapped cancer pathology slides predict diverse molecular phenotypes. *Nature communications*, 12(1):1613, 2021. 8

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

[17] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. 1, 3

[18] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. 1, 3

[19] Yonghang Guan, Jun Zhang, Kuan Tian, Sen Yang, Pei Dong, Jinxi Xiang, Wei Yang, Junzhou Huang, Yuyao Zhang, and Xiao Han. Node-aligned graph convolutional network for whole-slide image representation and classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18813–18823, 2022. 2

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[21] Lei He, L Rodney Long, Sameer Antani, and George R Thoma. Histology image analysis for carcinoma detection and grading. *Computer methods and programs in biomedicine*, 107(3):538–556, 2012. 1

[22] Simon Holdenried-Krafft, Peter Somers, Ivonne A Montes-Majarro, Diana Silimon, Cristina Tarín, Falko Fend, and Hendrik Lensch. Dual-query multiple instance learning for dynamic meta-embedding based tumor classification. *arXiv preprint arXiv:2307.07482*, 2023. 2

[23] Le Hou, Dimitris Samaras, Tahsin M Kurc, Yi Gao, James E Davis, and Joel H Saltz. Patch-based convolutional neural network for whole slide tissue image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2424–2433, 2016. 1

[24] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference,*

*Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 124–140. Springer, 2020. 1, 2

[25] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR, 2018. 1, 2, 3, 5, 6, 7, 11, 13

[26] Mingu Kang, Heon Song, Seonwook Park, Donggeun Yoo, and Sérgio Pereira. Benchmarking self-supervised learning on diverse pathology datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3344–3354, 2023. 5

[27] Fanjie Kong and Ricardo Henao. Efficient classification of very large images with tiny objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2384–2394, 2022. 2

[28] 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/CVF conference on computer vision and pattern recognition*, pages 14318–14328, 2021. 2, 5

[29] Honglin Li, Chenglu Zhu, Yunlong Zhang, Yuxuan Sun, Zhongyi Shui, Wenwei Kuang, Sunyi Zheng, and Lin Yang. Task-specific fine-tuning via variational information bottleneck for weakly-supervised pathology whole slide image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7454–7463, 2023. 1, 2, 5, 8

[30] Ruoyu Li, Jiawen Yao, Xinliang Zhu, Yeqing Li, and Junzhou Huang. Graph cnn for survival analysis on whole slide pathological images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 174–182. Springer, 2018. 1

[31] Yi Li and Wei Ping. Cancer metastasis detection with neural conditional random field. *arXiv preprint arXiv:1806.07064*, 2018. 1

[32] Tiancheng Lin, Zhimiao Yu, Hongyu Hu, Yi Xu, and Chang-Wen Chen. Interventional bag multi-instance learning on whole-slide pathological images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19830–19839, 2023. 1, 2, 5

[33] Geert Litjens, Clara I Sánchez, Nadya Timofeeva, Meyke Hermsen, Iris Nagtegaal, Iringo Kovacs, Christina Hulsbergen-Van De Kaa, Peter Bult, Bram Van Ginneken, and Jeroen Van Der Laak. Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis. *Scientific reports*, 6(1):26286, 2016. 1

[34] Ming Y Lu, Drew FK 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(6):555–570, 2021. 1, 2, 5, 11, 13

[35] Anant Madabhushi. Digital pathology image analysis: opportunities and challenges. *Imaging in medicine*, 1(1):7, 2009. 1

[36] Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. *Advances in neural information processing systems*, 10, 1997. 1

[37] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 3, 7, 8, 14, 15

[38] Liron Pantanowitz, Paul N Valenstein, Andrew J Evans, Keith J Kaplan, John D Pfeifer, David C Wilbur, Laura C Collins, and Terence J Colgan. Review of the current state of whole slide imaging in pathology. *Journal of pathology informatics*, 2(1):36, 2011. 1

[39] Hans Pinckaers, Bram Van Ginneken, and Geert Litjens. Streaming convolutional neural networks for end-to-end learning with multi-megapixel images. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1581–1590, 2020. 1

[40] Linhao Qu, Manning Wang, Zhijian Song, et al. Bi-directional weakly supervised knowledge distillation for whole slide image classification. *Advances in Neural Information Processing Systems*, 35:15368–15381, 2022. 1

[41] Linhao Qu, Zhiwei Yang, Minghong Duan, Yingfan Ma, Shuo Wang, Manning Wang, and Zhijian Song. Boosting whole slide image classification from the perspectives of distribution, correlation and magnification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21463–21473, 2023. 2

[42] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 410–420, 2007. 8

[43] Zhuchen Shao, Hao Bian, Yang Chen, Yifeng Wang, Jian Zhang, Xiangyang Ji, et al. Transmil: Transformer based correlated multiple instance learning for whole slide image classification. *Advances in neural information processing systems*, 34:2136–2147, 2021. 1, 2, 5, 13

[44] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 5

[45] Wenhao Tang, Sheng Huang, Xiaoxian Zhang, Fengtao Zhou, Yi Zhang, and Bo Liu. Multiple instance learning framework with masked hard instance mining for whole slide image classification. *arXiv preprint arXiv:2307.15254*, 2023. 1, 2, 5, 14, 15

[46] Rishabh Tiwari and Pradeep Shenoy. Overcoming simplicity bias in deep networks using a feature sieve. 2023. 1, 2

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 13

[48] Dayong Wang, Aditya Khosla, Rishab Gargeya, Humayun Irshad, and Andrew H Beck. Deep learning for identifying metastatic breast cancer. *arXiv preprint arXiv:1606.05718*, 2016. 1

[49] Hongyi Wang, Luyang Luo, Fang Wang, Ruofeng Tong, Yen-Wei Chen, Hongjie Hu, Lanfen Lin, and Hao Chen. Iteratively coupled multiple instance learning from instance

to bag classifier for whole slide image classification. *arXiv preprint arXiv:2303.15749*, 2023. 2

[50] Xiyue Wang, Jinxi Xiang, Jun Zhang, Sen Yang, Zhongyi Yang, Ming-Hui Wang, Jing Zhang, Wei Yang, Junzhou Huang, and Xiao Han. Scl-wc: Cross-slide contrastive learning for weakly-supervised whole-slide image classification. *Advances in neural information processing systems*, 35:18009–18021, 2022. 2

[51] Yinxi Wang, Kimmo Kartasalo, Philippe Weitz, Balazs Acs, Masi Valkonen, Christer Larsson, Pekka Ruusuvuori, Johan Hartman, and Mattias Rantalainen. Predicting molecular phenotypes from histopathology images: A transcriptome-wide expression–morphology analysis in breast cancer. *Cancer research*, 81(19):5115–5126, 2021. 1

[52] Jinxi Xiang and Jun Zhang. Exploring low-rank property in multiple instance learning for whole slide image classification. In *The Eleventh International Conference on Learning Representations*, 2022. 2

[53] Conghao Xiong, Hao Chen, Joseph Sung, and Irwin King. Diagnose like a pathologist: Transformer-enabled hierarchical attention-guided multiple instance learning for whole slide image classification. *arXiv preprint arXiv:2301.08125*, 2023. 2

[54] Jiawen Yao, Xinliang Zhu, Jitendra Jonnagaddala, Nicholas Hawkins, and Junzhou Huang. Whole slide images based cancer survival prediction using attention guided deep multiple instance learning networks. *Medical Image Analysis*, 65:101789, 2020. 1

[55] Cui Yufei, Ziquan Liu, Xiangyu Liu, Xue Liu, Cong Wang, Tei-Wei Kuo, Chun Jason Xue, and Antoni B Chan. Bayes-mil: A new probabilistic perspective on attention-based multiple instance learning for whole slide images. In *The Eleventh International Conference on Learning Representations*, 2022. 1, 2

[56] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 18802–18812, 2022. 1, 2, 4, 5, 13

[57] Yunlong Zhang, Yuxuan Sun, Honglin Li, Sunyi Zheng, Chenglu Zhu, and Lin Yang. Benchmarking the robustness of deep neural networks to common corruptions in digital pathology. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 242–252. Springer, 2022. 1

[58] Yu Zhao, Fan Yang, Yuqi Fang, Hailing Liu, Niyun Zhou, Jun Zhang, Jiarui Sun, Sen Yang, Bjoern Menze, Xinjuan Fan, et al. Predicting lymph node metastasis using histopathological images based on multiple instance learning with deep graph convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4837–4846, 2020. 2

[59] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020. 8

[60] Xinliang Zhu, Jiawen Yao, Feiyun Zhu, and Junzhou Huang. Wsisa: Making survival prediction from whole slide histopathological images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7234–7242, 2017. 1

## A. Overview

In the ACMIL appendix, we provide valuable resources and insights, including the source code (Sec. B), performance comparison against baseline, implementation details (Sec. C), additional experimental results (Sec. D), and a discussion on limitations (Sec. E).

## B. Source Code

The source code to train ACMIL is available at `https://github.com/dazhangyu123/ACMIL`. For further information on the environment setup and experiment execution, please refer to README.md. The implementation of ACMIL is based on the source code of ABMIL [25] and CLAM [34].

## C. Implementation Details

**Data Pre-processing.** We adopt the data pre-processing method from CLAM [34], which involves threshold segmentation and filtering to locate tissue regions in each whole-slide image (WSI). From these regions, we extract non-overlapping patches of size $256 \times 256$ at a magnification of $\times 20$ for Camelyon16 and LBC datasets, and at a magnification of $\times 10$ for BRACS.

**Feature Extraction.** Given that ACMIL freezes the feature extractor during training, we extract and save features with 512 dimensions for ResNet-18 and 384 dimensions for ViT-S/16 to conserve space and expedite computation.

**Model Architecture.** The learnable components of the model include one fully-connected layer to reduce features to 256 dimensions for ResNet-18 and 128 dimensions for ViT-S/16, a gated attention network, and a fully-connected layer for making predictions.

**Training.** All models are trained for 100 epochs using a cosine learning rate decay starting at 0.0001 for ViT-S/16 and 0.0002 for ResNet-18. We employ an Adam optimizer with a weight decay of 0.0001, and the batch size is set to 1.

**Hyperparameters.** For the setting of Camelyon16 and natural supervised pre-training, we set hyperparameters as $M = 2, K = 10, p = 0.6$. For the other situation, we set hyperparameters as $M = 5, K = 10, p = 0.6$.

## D. More Experimental Results

The additional experimental results include ablation study (Sec. D.1), more heatmap visualizations (Sec. D.3), instance feature analysis for normal slides (Sec. D.4), and discussion for the computational cost of MBA (Sec. D.5).

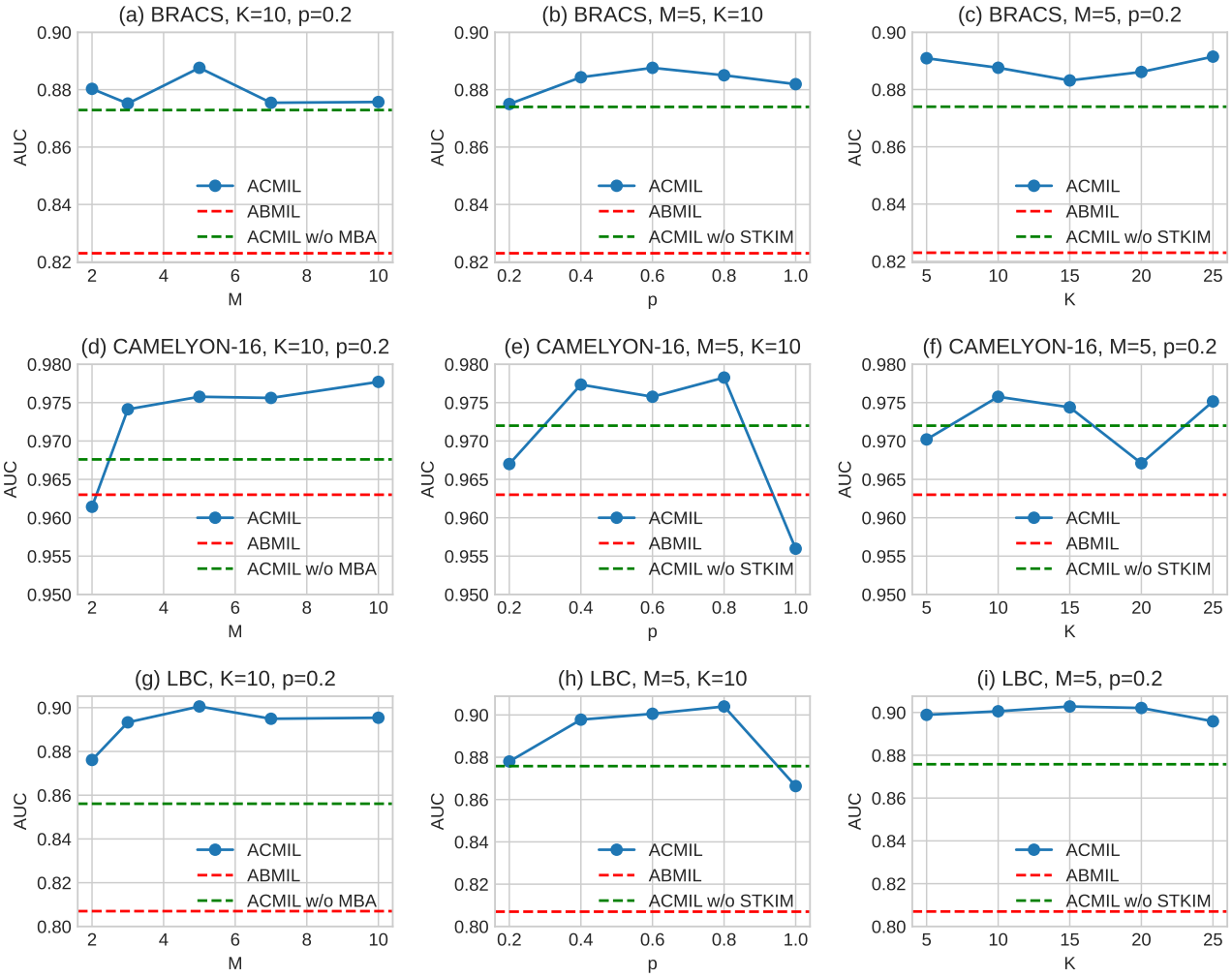



Figure 9. Hyperparameters sensitivity analysis on features extracted through the SSL pre-training. The effect of three hyperparameters, $K$, $p$, $M$, is investigated. Note that the red dot line denotes the performance of baseline, ABMIL, and the green dot line denotes the performance of ACMIL w/o MBA or STKIM. Five conclusions derived from the figure can be found in Sec. D.1.

## D.1. Ablation Study

Fig. 9 illustrates the AUC scores of ACMIL across three datasets when utilizing a ViT/B-16 feature extractor and varying hyperparameter settings. Several key observations emerge from these experiments:

**Solely relying on MBA or STKIM can improving performance**: Implementing either MBA or STKIM alone leads to significant performance improvements compared to ABMIL. The green dotted lines represent ACMIL's AUC performance without MBA or STKIM, outperforming the red dotted lines (ABMIL's performance) across all subfigures. Particularly noteworthy is the observation that MBA achieves better improvement than STKIM on all three datasets, with the green dot lines in the last two columns surpassing those in the first column, especially on the Camelyon and LBC datasets.

**Combining MBA with STKIM further enhances performance beyond what can be achieved with either MBA or STKIM alone**: Combining both MBA and STKIM yields performance improvements beyond what can be achieved with either MBA or STKIM individually. The blue dots represent ACMIL's performance under different hyperparameter combinations, with 39 out of 45 blue dots exceeding the green horizontal lines.

**Random Masking is important in STKIM**: Random masking is a crucial aspect of STKIM. The second column demonstrates that setting $p = 1.0$ leads to a performance deterioration across all three datasets. For LBC and Camelyon, a $p = 1.0$ setting even results in performance lower than the green horizontal lines, indicating the ineffectiveness of STKIM without random masking.

Table 4. The performance comparison between the baseline and our ACMIL across two attention mechanisms (i.e., gated attention (GA) and multiple head attention (MHA)), three datasets, and two pretrained methods.

| Method \ Performance | CAMELYON-16 | | BRACS | | LBC | | Average |
|---|---|---|---|---|---|---|---|
| | F1-score | AUC | F1-score | AUC | F1-score | AUC | |
| ResNet18 ImageNet pretrained | | | | | | | |
| GA | 0.757±0.020 | 0.790±0.027 | 0.523±0.028 | 0.723±0.035 | 0.465±0.040 | 0.798±0.013 | 0.676 |
| +ACMIL | 0.798±0.029 | 0.841±0.030 | 0.552±0.048 | 0.754±0.008 | 0.546±0.028 | 0.821±0.015 | 0.719 |
| Δ(%) | +4.1 | +5.1 | +2.9 | +3.1 | +8.1 | +2.3 | +4.3 |
| MHA | 0.752±0.030 | 0.775±0.027 | 0.502±0.039 | 0.738±0.019 | 0.531±0.025 | 0.817±0.011 | 0.686 |
| +ACMIL | 0.799±0.018 | 0.875±0.017 | 0.541±0.063 | 0.723±0.028 | 0.555±0.038 | 0.818±0.012 | 0.719 |
| Δ(%) | +4.7 | +10.0 | +3.9 | -1.5 | +2.4 | +0.1 | +3.3 |
| ViT-S/16 SSL pretrained | | | | | | | |
| GA | 0.914±0.031 | 0.945±0.027 | 0.680±0.051 | 0.866±0.029 | 0.595±0.036 | 0.831±0.022 | 0.805 |
| +ACMIL | 0.954±0.012 | 0.974±0.012 | 0.722±0.030 | 0.888±0.010 | 0.662±0.043 | 0.901±0.011 | 0.850 |
| Δ(%) | +4.0 | +2.9 | +4.2 | +2.2 | +6.7 | +7.0 | +4.5 |
| MHA | 0.931±0.032 | 0.961±0.017 | 0.656±0.030 | 0.850±0.030 | 0.619±0.032 | 0.864±0.013 | 0.813 |
| +ACMIL | 0.936±0.027 | 0.973±0.014 | 0.667±0.059 | 0.879±0.028 | 0.649±0.024 | 0.876±0.012 | 0.830 |
| Δ(%) | +0.5 | +1.2 | +1.1 | +2.9 | +3.0 | +1.2 | +1.7 |

**Keeping $p = 0.6$ or $p = 0.8$ achieves better performance**: The best performances across the three datasets are achieved at $p = 0.6$ and $p = 0.8$, as shown in the second column. Specifically, $p = 0.6$ achieves the best performance on the BRACS dataset, whereas $p = 0.8$ achieves the best performance on the other two datasets. In this paper, we set $p = 0.6$ as the default value.

**ACMIL is insensitive to the hyperparameter $K$**: The hyperparameter $K$ exhibits minimal sensitivity, where different $K$ values result in a performance difference of less than 1.0% AUC. In practice, setting $K$ to 10 is generally sufficient for achieving near-optimal performance.

## D.2. Performance Evaluation against Baseline

To assess the adaptability of our ACMIL to different attention mechanisms, we selected two prominent attention mechanisms as our baselines. The first is the gated attention (GA) mechanism [12], employed in approaches like ABMIL [25], CLAM [34], and DTFD-MIL [56]. The second is the multiple head attention (MHA) mechanism [47], utilized in methods such as TransMIL [43] and IPS transformer [4]. The results are presented in Table 4.

With GA as the baseline, ACMIL exhibits a substantial and comprehensive improvement in performance. All 12 performance metrics show enhancements, with an average gain of 4.4 points, a minimum increase of 2.2 points, and a maximum improvement of 8.1 points.

With MHA as the baseline, ACMIL also demonstrates performance improvements in the majority of terms (i.e., 11 out of 12 terms), achieving an average improvement of 2.5 points. In comparison to GA, MHA introduces parallel processing (i.e., heads). This modification enables the learning of different visual concepts across heads [8], contributing to a slight attenuation in the improvements brought by ACMIL.

## D.3. More heatmap visualization

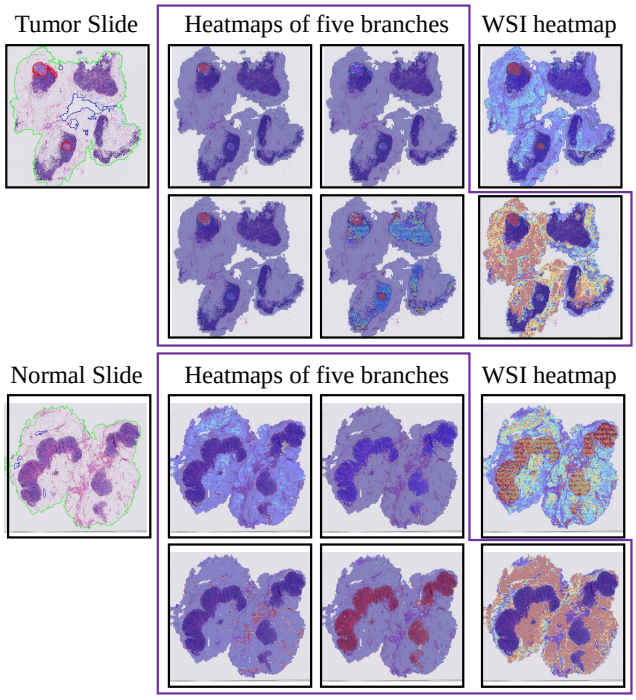



Figure 10. Heatmap visualizations for five attention branches. Different branches specialize in capturing specific features, contributing to the better interpretability for the bag (final) heatmap.

**Heatmap visualizations of five attention branches in MBA.** In Fig. 10, we present the heatmap visualizations for five attention branches and delve into the effects of these distinct branches. We've chosen two test slides in Camelyon16 for this analysis, including one tumor slide and one normal slide. For the tumor slide, we observe that all five branches capture the cancerous instances. Notably, the third and fifth branches successfully capture the entirety of the tumor regions, while the remaining three branches only manage to capture a subset of the tumor regions. Additionally, the third branch activates the adipose, and the fifth branch activates the lymphocytes regions. Overall, the averaged heatmap captures the whole tumor regions, along with slightly activating some normal regions. For the normal slide, the first two branches activate instances lying between adipose and lymphocytes regions. The third branch predominantly activates adipose tissue, the fourth branch emphasizes muscle regions, and the fifth branch highlights lymphocytes regions. Overall, the averaged heatmap activates all normal regions. This analysis illustrates how the different branches specialize in capturing specific features, contributing to a more comprehensive understanding of the data.

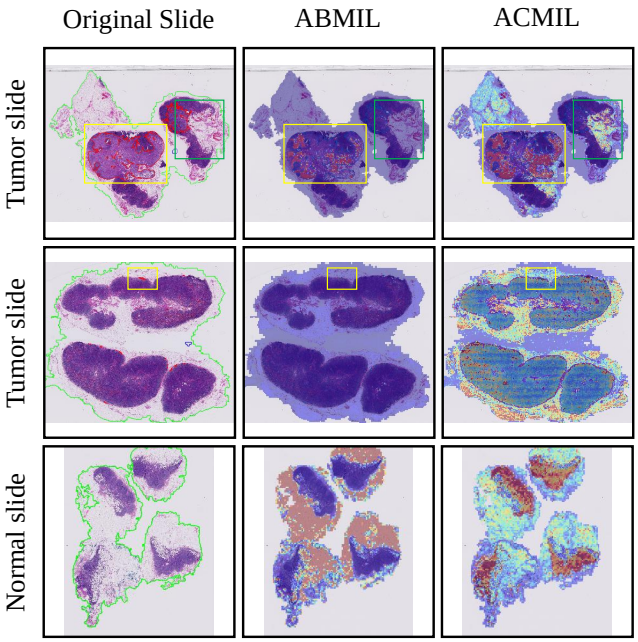



Figure 11. Heatmap visualizations with bad interpretability. Three cases indicates that ACMIL's approach of assigning broader attention values to a wide range of predictive instances doesn't consistently enhance interpretability.

**Heatmap visualizations with bad interpretability.** In Fig. 11, we present three cases (i.e., two tumor slides and one normal slide) with heatmap visualizations that exhibit poor interpretability. The first slide is a tumor slide. While ACMIL activates a greater number of cancerous instances than ABMIL (as indicated by the yellow box), it also activates some normal instances (visible in the green box). This mixed activation can potentially mislead experts during practical interpretability analysis. The second instance also concerns a tumor case but with small tumor regions. ABMIL accurately localizes the tumor regions (see yellow box). In contrast, ACMIL allocates more attention values to a broader range of predictive instances, which results in an inability to precisely locate the tumor regions. The third case pertains to a normal slide. In contrast to ABMIL, which provides misleading interpretability by predominantly focusing on adipose tissue, ACMIL assigns excessive attention values to lymphocytes regions. Consequently, the heatmap primarily highlights lymphocytes tissue instead of the expected comprehensive representation of normal tissue.

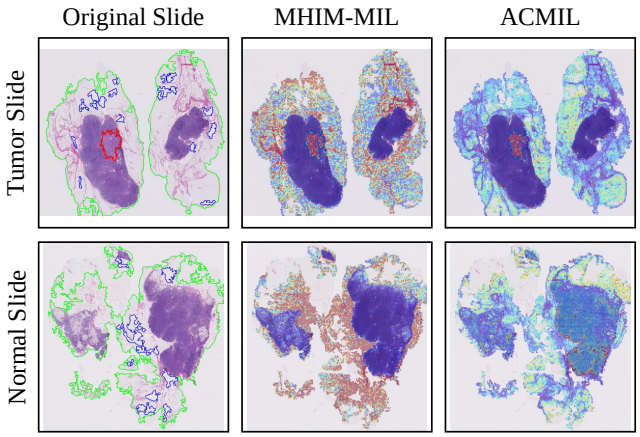
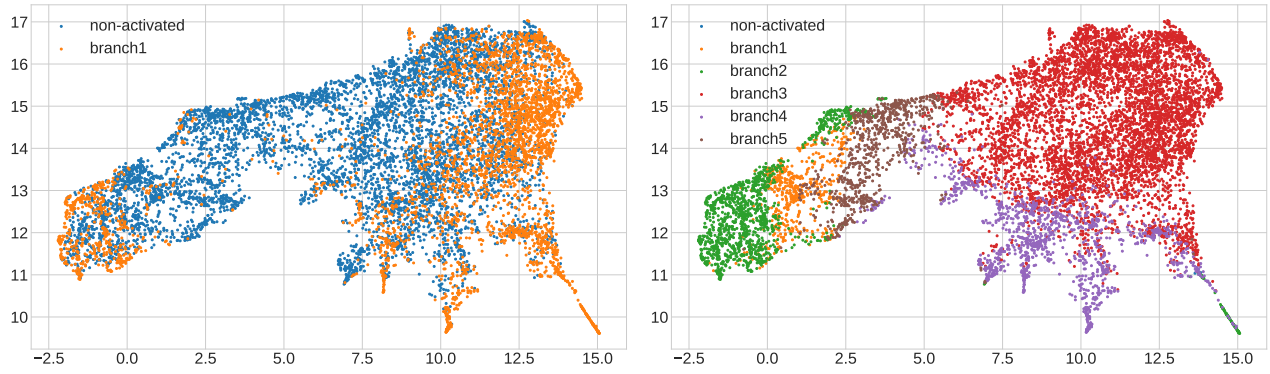



Figure 12. Comparison of heatmap visualizations between MHIM-MIL [45] and ACMIL (Zoom in for best view). ACMIL performs better on capturing comprehensive predictive instances.

**Comparison of heatmap visualization between MHIM-MIL [45] and ACMIL.** In Fig. 12, we present the heatmap visualizations of MHIM-MIL and ACMIL. For the tumor slide (first row), MHIM-MIL and ACMIL both capture all cancerous instances in the tumor region, but MHIM-MIL activates more normal instances than ACMIL. For the normal slide (second row), MHIM-MIL predominately activates adipose, whereas ACMIL activate all normal instances more uniformly.

### D.4. Instance feature analysis for normal slide

In Fig. 13, we present the UMAP visualization [37] of normal instance features in a typical Camelyon case, 'test_016'. The comparison between ABMIL and ACMIL is quite evident. ABMIL, with a single attention branch, activates only a fraction of normal instances. Conversely, ACMIL utilizes five branches, with each branch specializing in capturing specific patterns, resulting in the acti-




(a) ACMIL without MBA (b) ACMIL

Figure 13. UMAP visualization [37] of instance features in a normal case, Camelyon16 'test_016'. The normal instances exhibit distinct patterns, making it challenging for a single-branch model like ABMIL to capture them comprehensively. Consequently, ABMIL may overlook certain instances. In contrast, our ACMIL leverages multiple branches, each adept at capturing specific patterns, enabling ACMIL to activate a greater number of normal instances. Note that the instance is considered active when its attention value surpasses $\frac{1}{N}$.

Table 5. Comparison of performance and computational cost requirements between MHIM-MIL and STKIM. We report the auc, FLOPs, training time per epoch (Time), and peak memory usage (Mem.) on the CAMELYON-16 (C16) dataset. The flops are measured with the number of instances of a bag being 1024.

| Model | C16 | BRACS | LBC | FLOPs | Time | Mem. |
|---|---|---|---|---|---|---|
| ResNet18 ImageNet pretrained | | | | | | |
| ABMIL | **0.790** | 0.723 | 0.798 | **201M** | **8.0s** | **0.3G** |
| MHIM-MIL | 0.772 | 0.774 | 0.816 | **201M** | 20.8s | 1.9G |
| STKIM | 0.779 | **0.789** | **0.820** | **201M** | **8.0s** | **0.3G** |
| ViT-S/16 SSL pretrained | | | | | | |
| ABMIL | 0.945 | 0.866 | 0.831 | **84M** | **6.4s** | **0.2G** |
| MHIM-MIL | **0.970** | 0.865 | **0.872** | **84M** | 16.8s | 1.0G |
| STKIM | 0.968 | **0.873** | 0.856 | **84M** | 6.5s | **0.2G** |

Table 6. Comparison of performance, time and memory requirements between ABMIL and MBA. We report the auc, the FLOPs, the training time per epoch (Time), and the peak memory usage (Mem.) on the CAMELYON-16 dataset (C16). The flops are measured with the number of instances of a bag being 1024.

| Model | C16 | BRACS | LBC | FLOPs | Time | Mem. |
|---|---|---|---|---|---|---|
| ResNet18 ImageNet pretrained | | | | | | |
| ABMIL | 0.790 | 0.723 | 0.798 | **201M** | **8.0s** | **0.3G** |
| +MBA | **0.850** | **0.797** | **0.818** | 202M | 11.6s | **0.3G** |
| ViT-S/16 SSL pretrained | | | | | | |
| ABMIL | 0.945 | 0.866 | 0.831 | **84M** | **6.4s** | **0.2G** |
| +MBA | **0.973** | **0.878** | **0.875** | 85M | 9.3s | **0.2G** |

vation of nearly all normal instances. This observation demonstrates the superior ability of ACMIL to encompass a broader range of patterns in the data.

### D.5. Discussion Combining Performance and Computational Cost

**STKIM and MHIM-MIL [45].** We conducted a comprehensive comparison between STKIM and MHIM-MIL, focusing on computational cost and performance, as detailed in Tab. 5. For the computational cost, STKIM demonstrates nearly identical training time consumption and GPU memory usage as the baseline, ABMIL. This similarity arises because STKIM primarily integrates a sorting algorithm, which does not substantially increase resource requirements. On the other hand, MHIM-MIL introduces a teacher model while requiring two forward propagations, leading to significantly higher GPU memory usage and training time consumption. Due to the masking operator will be discarded in the evaluation, STKIM and MHIM-MIL keep the same evaluation cost (FLOPs) with the ABMIL. For the performance, STKIM delivers comparable results to MHIM-MIL across three datasets and with two pretrained backbone models. Notably, STKIM outperforms MHIM-MIL in four out of six performance metrics while lagging behind in the remaining two.

**MBA.** In Tab. 6, we present the comparison of performance and computational cost between ABMIL and MBA. Notably, MBA demonstrates a substantial performance improvement over ABMIL. Meanwhile, due to introducing a small number of parameters, the FLOPs and Memory cost increases marginally. Otherwise, the inclusion of the newly introduced diversity loss leads to a notable increase in time cost.

## E. Limitations

Even though our ACMIL is able to improve the model generalization ability and interprebility of MIL methods for WSI analysis, it still exist some limitations that are required

to further explore in the future. Firstly, our ACMIL also will produce heatmap visualization with bad interpretability. Intuitively, this also will hurt the model generalization. To solve this, future work can further elaborate attention mechanism. Secondly, we find that the better representations will weaken the requirements for better attention design. Although we verify the effectiveness of our ACMIL on one of the currently best feature extraction methods, SSL pre-training, we still cannot ensure its scalability to the better representations.