

Pseudo-Bag Mixup Augmentation for Multiple Instance Learning-Based Whole Slide Image Classification

Pei Liu^{ID}, Luping Ji^{ID}, Member, IEEE, Xinyu Zhang, and Feng Ye

Abstract—Given the special situation of modeling gigapixel images, multiple instance learning (MIL) has become one of the most important frameworks for Whole Slide Image (WSI) classification. In current practice, most MIL networks often face two unavoidable problems in training: i) insufficient WSI data and ii) the sample memorization inclination inherent in neural networks. These problems may hinder MIL models from adequate and efficient training, suppressing the continuous performance promotion of classification models on WSIs. Inspired by the basic idea of Mixup, this paper proposes a new Pseudo-bag Mixup (PseMix) data augmentation scheme to improve the training of MIL models. This scheme generalizes the Mixup strategy for general images to special WSIs via pseudo-bags so as to be applied in MIL-based WSI classification. Cooperated by pseudo-bags, our PseMix fulfills the critical size alignment and semantic alignment in Mixup strategy. Moreover, it is designed as an efficient and decoupled method, neither involving time-consuming operations nor relying on MIL model predictions. Comparative experiments and ablation studies are specially designed to evaluate the effectiveness and advantages of our PseMix. Experimental results show that PseMix could often assist state-of-the-art MIL networks to refresh their classification performance on WSIs. Besides, it could also boost the generalization performance of MIL models in special test scenarios, and promote their robustness to patch occlusion and label noise. Our source code is available at <https://github.com/liupe101/PseMix>.

Index Terms—Computational pathology, data augmentation, multiple instance learning, pseudo-bag Mixup, whole slide image classification.

I. INTRODUCTION

HISTOLOGICAL Whole-Slide Image (WSI) serves as the gold standard of pathology diagnosis and plays a vital role in cancer assessment and treatment [1], [2], [3]. Owing to

Manuscript received 2 November 2023; accepted 4 January 2024. Date of publication 9 January 2024; date of current version 2 May 2024. This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant ZYGX2022YGRH015 and in part by the National Natural Science Foundation of China (NSFC) under Grant 61972072. (Corresponding author: Luping Ji.)

Pei Liu, Luping Ji, and Xinyu Zhang are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: yuukip@163.com; jiluping@uestc.edu.cn; universestarrysky@126.com).

Feng Ye is with the Institute of Clinical Pathology, West China Hospital, Sichuan University, Chengdu 610041, China (e-mail: fengye@scu.edu.cn).

Digital Object Identifier 10.1109/TMI.2024.3351213

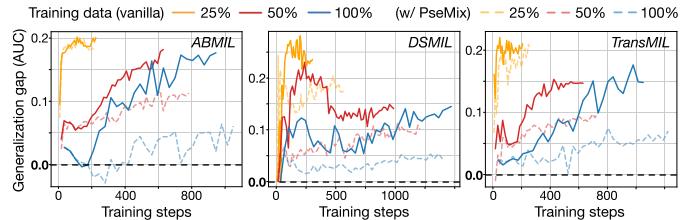


Fig. 1. Generalization gap of MIL models in training. It is measured by the model performance gap between training and test sets, to assess the generalization ability of models, following [20], [21]. Three state-of-the-art MIL models are trained on TCGA-BRCA WSIs. Vanilla models often show rapidly-growing gaps. The models trained with our PseMix could ease these without introducing extra complicated techniques.

the advances in deep learning, many clinical tasks with WSIs, such as lymph node metastasis detection, cancer subtyping, and cancer prognosis, have been made automatic and precise by specialized models [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]. These models show exciting progress towards overcoming the long-standing drawbacks of manual inspection [16], [17]. Across them, a weakly-supervised learning framework called multiple instance learning (MIL) is widely used. It treats a gigapixel WSI (*e.g.*, 40,000 × 40,000 pixels) as a bag of multiple instances, and learns a global feature representation via instance embedding and aggregation [18]. This framework enables neural network (NN) models to handle gigapixel images and no longer rely on fine-grained instance-level labels. For those reasons, MIL has become a prevalent paradigm for WSI classification, achieving considerable successes in computational pathology (CPATH) [19].

Nonetheless, our empirical study, as presented in Figure 1, reveals that some state-of-the-art MIL models [8], [9], [18] often perform poorly in generalization, even worse in probable small data scenarios. The literatures on learning theory and NNs [22], [23], [24] point out that such kind of problem is the consequence of *data memorization*, fundamentally caused by the nature of NNs and existing even in the presence of strong regularization. Due to internal data memorization characteristics, MIL networks could merely memorize, rather than efficiently learn or generalize from, given training samples, thus limiting the model performance on unseen test data and increasing generalization gaps. Besides, as an external cause, insufficient samples may hinder models from

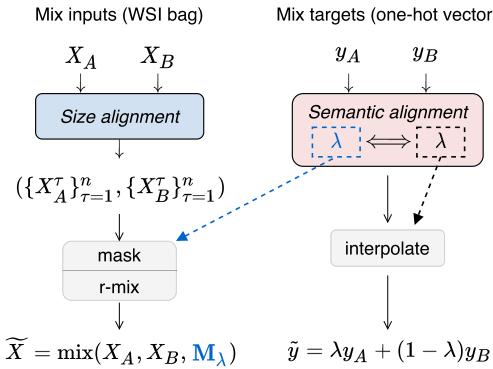


Fig. 2. A conceptual framework of PseMix for MIL-based WSI classification. Size alignment means that two input representations (*e.g.*, feature vectors and image matrices) are aligned in size at every dimension involving mixing. X_A^τ and X_B^τ denote pseudo-bags. M_λ is a binary mask. R-mix means random mixing. PseMix generalizes Mixup and fulfills its critical size alignment and semantic alignment via pseudo-bags.

inadequate training, further degrading generalization performance. In contrast to a multitude of studies on classical image classification for tackling the data memorization problem in models [20], [25], the current works on WSI classification rarely investigate this problem in MIL formally. Thereby, most modern MIL models still carry the risk of inadequate and inefficient training, suppressing their continuous performance promotion.

Mixup [26], as a special data augmentation method, shows great promise in mitigating the inherent data memorization behavior in NNs. Unlike most augmentation methods that only manipulate single input, Mixup operates on two inputs and generates the interpolation samples between them ($\tilde{x} = \lambda x_i + (1 - \lambda)x_j$, $\tilde{y} = \lambda y_i + (1 - \lambda)y_j$). These samples are also interpreted as neighbor or vicinity samples, leveraged as augmented data for training. In this particular way, NNs are never limited to memorizing finite training samples from the original distribution; instead, they are encouraged to learn from the vicinity distribution of training samples and obtain a decision boundary with better generalization [26]. Such a simple yet efficient method has been developed and witnessed remarkable success in its application to heterogeneous data, *e.g.*, natural image [27], [28], [29], graph [30], 3D point cloud, [31], visual-language [32], etc. Further adoption of Mixup for histological WSIs is strongly anticipated. However, the characteristic of WSIs poses several challenges to it. Unlike other data sources, i) histological WSIs are usually cast as bags for weakly-supervised MIL; ii) WSI bags are often irregular and not well-aligned in their representation space for interpolation (*i.e.*, $\tilde{x} = \lambda x_i + (1 - \lambda)x_j$); and iii) a single bag may contain tens of thousands of instances, each with a high-dimensional feature vector [7]. These characteristics suggest that a practical and suitable Mixup method for WSIs should be compatible with MIL, capable of mixing alignment, and efficient.

For the purposes above, this paper proposes a new Pseudo-bag Mixup (PseMix) data augmentation scheme, as illustrated in Figure 2. We address the challenges aforementioned from the following three aspects. (1) *MIL compatibility*: our scheme

takes WSI bags as input and outputs the mixed bags that still could be processed by MIL. (2) *Mixing alignment*: we divide each bag into n pseudo-bags for size alignment in bag mixing, and leverage the proportion of pseudo-bag mixing as λ to mix targets for semantic alignment. (3) *Efficiency*: our pseudo-bag division is designed as an algorithm with linear time and space complexity, and PseMix is decoupled from the stage of MIL so as to be plugin-and-play. In addition, our PseMix further introduces a random mixing mechanism (*i.e.*, r-mix in Figure 2) for more data diversity and efficient learning on vicinity samples. Comparative experiments confirm that our PseMix could serve as an effective Mixup data augmentation method. Apart from the improvements across three different classification tasks, the models trained with our PseMix could often surpass vanilla models by a large margin in various generalization and robustness tests.

The main contributions of this paper are as follows. (1) This paper proposes a new Pseudo-bag Mixup (PseMix) data augmentation scheme for MIL-based WSI classification, to help MIL models improve performance and obtain better generalization and robustness. It is an efficient and plugin-and-play one, neither involving time-consuming operations nor relying on the prediction of MIL models. (2) This paper generalizes the Mixup strategy for common images to special WSIs by pseudo-bag-based size and semantic alignment. Furthermore, it introduces a random mixing mechanism into standard Mixup for more data diversity and efficient learning on augmented mixing samples. (3) Comparative experiments, ablation study, and various scenarios, *e.g.*, regular WSI classification, special generalization tests, patch occlusion robustness, label noise robustness, etc, are specially designed to demonstrate the broad advantages of PseMix.

II. RELATED WORK

A. Multiple Instance Learning for WSI Classification

Multiple instance learning (MIL) [18] nowadays has been increasingly used in computational pathology for weakly-supervised WSI classification [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]. Its regular procedure [19] on gigapixel WSIs could be decomposed into two main stages: i) WSI preprocessing, each digital slide is transformed into a bag of instances through patching and patch feature extracting; and ii) weakly-supervised MIL, a bag is compressed into a global vector for slide-level prediction by a MIL network [18]. The first stage is computationally intensive because of the transformation from gigapixel images to massive feature vectors. Thus, this stage is usually fixed for saving computational costs, not involved in regular training [35], [36].

B. Mixup Data Augmentation

The basic idea of Mixup [26] could be formulated as follows:

$$\tilde{x} = \lambda x_i + (1 - \lambda)x_j, \quad \tilde{y} = \lambda y_i + (1 - \lambda)y_j, \quad (1)$$

where the two samples (x_i, y_i) and (x_j, y_j) are drawn from training data, and $\lambda \in [0, 1]$ follows a Beta distribution with a parameter $\alpha \in [0, +\infty]$. Despite its simplicity, Mixup has

TABLE I
RELATED DATA AUGMENTATION METHODS. THE MIX MANNER, R-MIX, MEANS RANDOM MIXING

Method	Data type	Input			Mix manner	Target	
		Mix	Size alignment			Mix	Semantic alignment
Mixup (2018) [26]	data-agnostic	✓	aligned by default	interpolate	✓	input interpolation scale	
TransMix (2022) [29]	natural image	✓	scaling or cropping	mask & mix	✓	target attention weights	
\mathcal{G} -Mixup (2022) [30]	graph	✓	graphon estimation	interpolate	✓	graphon interpolation scale	
PointPatchMix (2023) [31]	point cloud	✓	point patch	mask & mix	✓	patch attention scores	
Pseudo-bag (2021) [6]	WSI	✗	—	—	✗	—	
ReMix (2022) [33]	WSI	✓	bag prototype	mask & mix	✗	—	
RankMix (2023) [34]	WSI	✓	ranking and dropping	interpolate	✓	bag interpolation scale	
PseMix (ours)	WSI	✓	pseudo-bag	mask & r-mix	✓	pseudo-bag mixing ratio	

been proven, empirically and theoretically, to be an effective and practical data augmentation strategy across many different fields for improving the generalization and robustness of deep learning models [21], [27], [28], [29], [30], [31], [32], [37].

From an alignment perspective, we could observe the two critical factors in the Mixup formulation given by Equation 1. (1) Size alignment, *i.e.*, x_i and x_j must be aligned in size at every dimension involving mixing (*e.g.*, height and width dimensions for two input gray images). (2) Semantic alignment, *i.e.*, the mixed input \tilde{x} is derived from x_i and x_j , and its content is controlled by λ , so its corresponding label \tilde{y} should be associated with y_i and y_j , and its value should be determined by the same λ . Based on this perspective, the state-of-the-art Mixup variants for different data sources could be summarized in a unified form, as exhibited in the top half of Table I.

Apart from interpolation, masking is also one of the most important means for input mixing implementation. In the domain of image classification, it is reported [29] that masking-based Mixup variants often perform better than interpolation-based ones because interpolation operation treats background and foreground equally and often yield meaningless samples. However, these leading Mixup methods could not be directly or efficiently applied to gigapixel WSIs, as WSIs are yet another kind of heterogeneous data, usually cast as irregular bags containing many high-dimensional instances [7], [19].

C. Data Augmentation for Multiple Instance Learning

Traditional image-level data augmentation methods [38], such as flipping, rotation, blurring, etc, have been adopted in WSI analysis [39], [40]. They are utilized in the stage of WSI preprocessing to generate different instance features for the same patch image, aiming to increase the diversity of instance-level features. However, this manner is computationally expensive to be adopted in MIL training, because there are usually tens of thousands of patch images and each one needs multiple feature extraction for a single WSI [35], [36].

The other methods, specially designed for MIL-based WSI classification, could be roughly divided into two categories: i) embedding-level data augmentation and ii) instance-level or bag-level data augmentation. The former focuses on augmenting instance embeddings with bag prototypes [33], generative adversarial networks [35], or diffusion models [41]. The latter

mainly generates new subsets from bags via hierarchical [6], [42] or random sampling [11]. However, these methods have not yet explored the basic idea of Mixup.

As shown in the bottom half of Table I, although ReMix [33] proposes to mix two bag inputs, it restricts the mix within the same class, not falling into the range of Mixup [26]. Most recently, RankMix [34], as a Mixup variant, studies the basic idea of Mixup for WSI classification. It strictly follows the original interpolation way for bag mixing. However, it heavily relies on MIL networks to obtain instance attention scores for importance ranking and instance alignment. Unfortunately, there are often potential biases between attention scores and instance importance, as highlighted in [43]. Accordingly, there is still room for exploring Mixup on WSI data.

III. METHOD

In this section, we elaborate on how Mixup can be generalized to WSI classification via pseudo-bags, and introduce our random mixing mechanism. In the design of PseMix, we consider algorithm efficiency to make it practical and efficient to apply. Moreover, our PseMix is decoupled from the stage of MIL, so it is a plugin-and-play method for MIL models.

A. Preliminary

1) Notation and Convention: As shown in Figure 3(a), a single digital slide is sliced into image patches, and then is preprocessed into high-dimensional vectors by a parameter-fixed feature extractor. These preprocessed feature vectors are taken as a bag of multiple instances. Given N slides, we denote their processed data by

$$\mathcal{D} = \{(X_i \in \mathbb{R}^{m_i \times d}, y_i)\}_{i=1}^N, \quad (2)$$

where X_i is the i -th bag, m_i is instance number, d means the dimension of each feature vector, and y_i is a slide-level label. Let $x_{i,j} \in \mathbb{R}^d$ denote the j -th instance of the i -th bag, hence $X_i = \{x_{i,j}\}_{j=1}^{m_i}$. Let $\{X_i^\tau\}_{\tau=1}^n$ represent the n pseudo-bags of X_i , where τ denotes the index of pseudo-bag.

2) Multiple Instance Learning: In the absence of instance-level labels, most MIL framework usually takes WSI bags as input and leverages a neural network to learn bag-level representations by aggregating multiple instance embeddings,

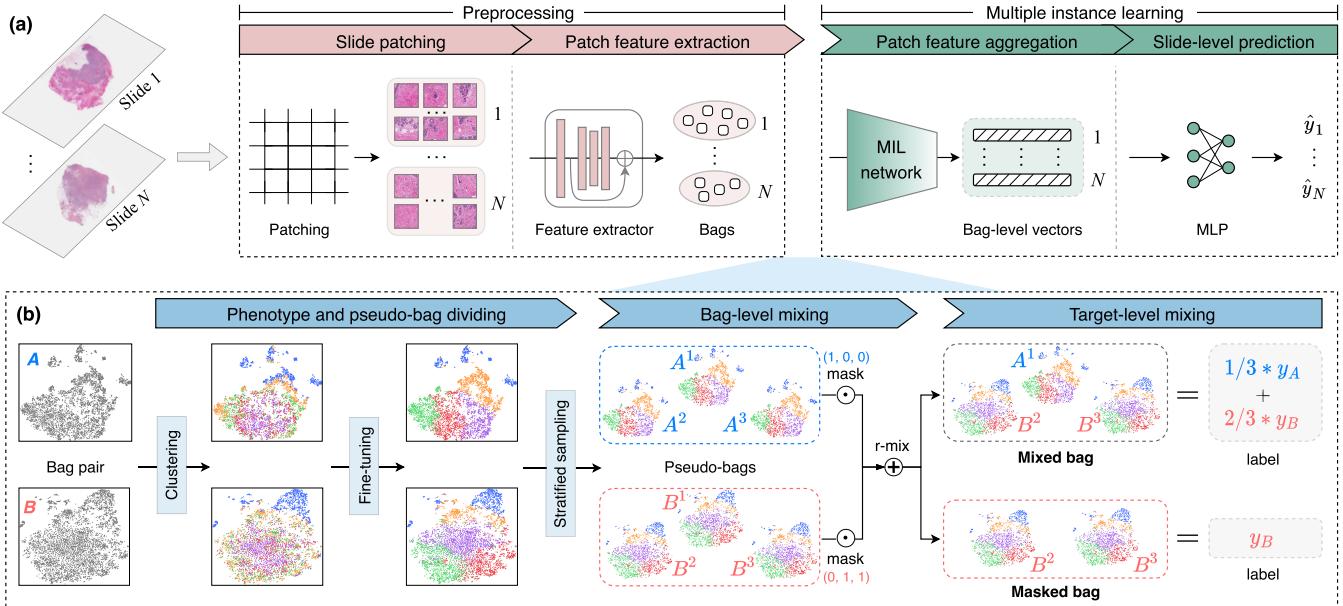


Fig. 3. PseMix data augmentation for MIL-based WSI classification. **(a)** Classical MIL paradigm for weakly-supervised WSI analysis. **(b)** Illustration of PseMix. Two WSI bags A and B are taken as examples. Solid rectangular boxes give the t-SNE visualization of instances, where scatter points represents instances and each color indicates a specific phenotype. A^1, A^2, \dots, B^3 are pseudo-bag notations. R-mix means random mixing.

followed by an MLP (multi-layer perceptron) to output slide-level predictions, $\hat{y}_1, \dots, \hat{y}_N$, as illustrated in Figure 3(a). This framework is also called embedding-level MIL [18].

State-of-the-art MIL networks contain ABMIL [18], DSMIL [8], and TransMIL [9]. They are frequently adopted in weakly-supervised WSI classification. This paper will mainly employ them as MIL networks for experiments. Note that the parameters of feature extractor are fixed and not updated with MIL training. For the feature extractor, we simply follow CLAM [7] to utilize a truncated ResNet-50 model [44] pre-trained on ImageNet [45]; and $d = 1024$.

B. Pseudo-Bag Mixup Augmentation

As shown in Figure 3(b), our scheme has three necessary steps: i) phenotype and pseudo-bag dividing, ii) bag-level mixing, and iii) target-level mixing. Next, we will describe their implementation details one by one.

1) Phenotype and Pseudo-Bag Division: To align bag inputs in size for mixing, we cluster the instances of each bag into l different phenotypes, and then adopt phenotype-stratified sampling to obtain n instance-disjoint pseudo-bags. Pseudo-bag division consists of two major steps. Their details are provided in Algorithm 1.

(1) **Phenotype clustering.** In this step, i) we first obtain l initial instance clusters, following a bag-prototype-based clustering method [42]; ii) then we fine-tune these clusters for k times, using an iterative cluster updating algorithm similar to that used in K-means. Specifically, for each WSI bag, its prototype is calculated using the mean of its instance features, as shown in Algorithm 1. Then cosine similarity is measured between each instance and bag prototype. Based on these similarity metrics, all instances are grouped into l initial clusters, where the c -th cluster contains the instances whose

similarity scores are in $[-1 + \frac{2(c-1)}{l}, -1 + \frac{2c}{l}]$. However, these initial clusters could not be desirable, as shown in Figure 3(b). One possible reason is that mean-feature-based calculation only approximately estimates the prototype of WSI bag and it thus may lead to the potential bias of instance similarity measurement. Therefore, we fine-tune those initial clusters to further improve the quality of instance clustering. Each instance cluster could be interpreted as a specific pathology phenotype [6].

We note that the attention-based prototype proposed in [42] may mitigate the bias of similarity measurement and obtain better results. However, it relies on the attention score produced by MIL models. This would increase computation costs and make our PseMix dependent on attention-based models and tightly coupled with MIL training. Thus, it is not used herein to calculate initial instance clusters. Besides, the clustering algorithm we adopt here could be taken as a simple variant of classical K-means. Compared with K-means, this variant calculates initial clusters by a different method and updates clusters by a same iterative mechanism but with a different stopping condition. We empirically find that it could often run faster and obtain better overall results than classical K-means. Please refer to Section IV-E for more comparisons and discussions.

(2) **Phenotype-stratified sampling.** We randomly sample instances without replacement from each phenotype stratum, obtaining n pseudo-bags (each with roughly the same instance numbers) from each bag. As a result, the time and space complexity of Algorithm 1 are $\mathcal{O}(lkm_i)$ and $\mathcal{O}(m_i)$, respectively. Given that l and k are usually set in $[10^0, 10^1]$, the procedure of pseudo-bag division can be executed in a linear time and space complexity w.r.t. instance number for each bag. Its actual computational cost is measured through experiments and presented in Section IV-E.

Algorithm 1 Pseudo-Bag Division (for a Single bag)

Input: a WSI bag $X_i = \{x_{i,j}\}_{j=1}^{m_i}$, pseudo-bag number n , phenotype number l , fine-tuning times k .

Output: pseudo-bags $\{X_i^\tau\}_{\tau=1}^n$.

```

1 // Prototype-based clustering
2  $p_i \leftarrow \frac{1}{m_i} \sum_j x_{i,j}$  // bag prototype
3 initialize phenotype indicator  $C_i = \{c_{i,j}\}_{j=1}^{m_i}$ 
4 for  $j \leftarrow 1$  to  $m_i$  do
5    $s_{i,j} \leftarrow \cos(p_i, x_{i,j})$  // cosine similarity
6   let  $c \in \{1, 2, \dots, l\}$ 
7    $c_{i,j} \leftarrow$  find a  $c$  s.t.  $s_{i,j} \in [-1 + \frac{2(c-1)}{l}, -1 + \frac{2c}{l}]$ 
8 end
9 // Phenotype fine-tuning
10 for  $t \leftarrow 1$  to  $k$  do
11   initialize phenotype centroids  $\{f_{i,c} \in \mathbb{R}^d\}_{c=1}^l$ 
12   for  $c \leftarrow 1$  to  $l$  do
13      $I_c \leftarrow \{j \mid c_{i,j} = c\}$ 
14      $f_{i,c} \leftarrow \frac{1}{|I_c|} \sum_{j \in I_c} x_{i,j}$ 
15   end
16   for  $j \leftarrow 1$  to  $m_i$  do
17      $c_{i,j} \leftarrow \arg \min_c \cos(f_{i,c}, x_{i,j})$ 
18   end
19 end
20 // Phenotype-stratified sampling
21 initialize  $n$  empty pseudo-bags  $\{X_i^\tau = \emptyset\}_{\tau=1}^n$ 
22 for  $c \leftarrow 1$  to  $l$  do
23    $I_c \leftarrow \{j \mid c_{i,j} = c\}$ 
24 randomly and uniformly split  $I_c$  into  $n$  parts
25 fetch the instances w.r.t. the  $n$  parts of  $I_c$ 
26 append the instances to  $X_i^1, \dots, X_i^n$ , respectively
27 end

```

(3) *More discussions.* Previous studies of pseudo-bags [6], [11], [42] assume that *pseudo-bags could inherit labels from their parent bag*. This paper also follows this basic assumption. However, this assumption is an ideal case because the inconsistent distribution between pseudo-bags and their parent bag may incur noises to pseudo-bag labels. Accordingly, in pseudo-bag dividing, we especially fine-tune phenotype clusters and adopt the way of phenotype-stratified sampling, to i) preserve the original phenotype distribution as much as possible in those divided pseudo-bags and ii) make pseudo-bags consistent with their parent bag in semantics. These strategies could make the basic assumption of pseudo-bag more reasonable and feasible, contributing to the semantic alignment in target mixing (introduced in Section III-B.3). We will further evaluate and discuss different pseudo-bag division methods through comparative experiments.

(2) *Bag-Level Mixing:* As illustrated in Figure 3(b), we augment WSI bags in this step by masking and random mixing (r-mix) on two input bags. Without loss of generality, let (A, B) denote any bag pair from original training samples, where $A, B \in \{1, 2, \dots, N\}$ and $A \neq B$. We further denote their respective pseudo-bags by $\{X_A^\tau\}_{\tau=1}^n$ and $\{X_B^\tau\}_{\tau=1}^n$, where pseudo-bags could be arranged in arbitrary orders.

a) *Masking:* We randomly mask $1 - \lambda$ and λ pseudo-bags in A and B , respectively. Similar to Mixup [26], $\lambda \in [0, 1]$ is drawn from a Beta distribution, i.e., $\lambda \sim \text{Beta}(\alpha, \alpha)$, where α is non-negative. We write two masked bags as follows:

$$X'_A = (1 - \mathbf{M}_\lambda) \odot \{X_A^\tau\}_{\tau=1}^n, \quad X'_B = \mathbf{M}_\lambda \odot \{X_B^\tau\}_{\tau=1}^n, \quad (3)$$

where $\mathbf{M}_\lambda \in \{0, 1\}^n$ is a binary mask indicating which pseudo-bag to mask and keep, $\sum \mathbf{M}_\lambda = \lfloor \lambda(n+1) \rfloor$, and \odot represents element-wise product.

b) *Random mixing:* As depicted in Figure 3(b), our r-mix operation could output two kinds of augmented bags, i.e., mixed or masked bags. Specifically, in r-mix, we let the masked bag X'_B join with X'_A or not join but directly output. For the first case, the output is a mixed bag:

$$\tilde{X} = (1 - \mathbf{M}_\lambda) \odot \{X_A^\tau\}_{\tau=1}^n \cup \mathbf{M}_\lambda \odot \{X_B^\tau\}_{\tau=1}^n. \quad (4)$$

It also contains n pseudo-bags. In Equation 4, both of A and B are cast as n pseudo-bags and pseudo-bags are the minimum unit in mixing. All of these are the necessary prerequisites for size alignment and subsequent complementary masking operations (i.e., \mathbf{M}_λ and $1 - \mathbf{M}_\lambda$). For the second case, i.e., not joining, the output is a masked bag:

$$X' = \mathbf{M}_\lambda \odot \{X_B^\tau\}_{\tau=1}^n. \quad (5)$$

It is utilized as the other kind of augmented bags for training.

c) *Motivation behind random mixing:* We introduce a simple random mixing mechanism into standard Mixup for two purposes: **i**) enhancing the diversity of training samples and **ii**) making models efficiently learn from vicinity samples (mixed bags). Pseudo-bag-masked bags could be viewed as the intermediate samples between original training data and synthetic mixed data. They could smooth the transition from original distribution to vicinity distribution, thereby helping models to efficiently learn from synthetic vicinity samples. We will evaluate and discuss our r-mix in ablation study.

From Equation 3, 4, and 5, we can see that our bag-level mixing is decoupled from the stage of MIL. Consequently, our scheme could serve as a plugin-and-play data augmentation method for MIL, different from the RankMix [34] that relies on MIL model predictions for bag alignment.

3) *Target-Level Mixing:* For the mixed bag \tilde{X} , we adopt the same λ as that used in Equation 4 for target mixing, i.e.,

$$\tilde{y} = \lambda y_A + (1 - \lambda) y_B. \quad (6)$$

Intuitively, there are λ and $1 - \lambda$ pseudo-bags from A and B , in \tilde{X} . Given the basic assumption that pseudo-bags share the same label as their parent bag, the interpolation scale of y_A and y_B could be set to λ (based on pseudo-bag mixing ratio) for semantic alignment. For the masked bag X' , we set $y' = y_B$, also following that basic assumption.

Let \mathcal{D}_{aug} denote the final training data augmented by our PseMix. It thus contains (\tilde{X}, \tilde{y}) and (X', y') . In our implementation, we adjust their proportion in \mathcal{D}_{aug} by adopting a hyper-parameter $p \in [0, 1]$ to set the probability of mixing in r-mix. Therefore, for any sample $\mathcal{D}_{\text{aug}}^s \in \mathcal{D}_{\text{aug}}$, it is written as follows:

$$\mathcal{D}_{\text{aug}}^s = \begin{cases} (\tilde{X}, \tilde{y}) & \text{with a probability of } p, \\ (X', y') & \text{with a probability of } 1 - p. \end{cases} \quad (7)$$

TABLE II
STATISTICAL DETAILS OF THREE WSI DATASETS

Item	TCGA-BRCA	TCGA-LUNG	TCGA-RCC
# Patients	898	947	895
# Slides	953	1,044	937
# Patches	2,961,552	3,235,064	3,317,384
# Patches per slide	3,107.6	3098.7	3,540.4

IV. EXPERIMENTS

In this section, we mainly evaluate the effectiveness and advantages of PseMix through comparative experiments and ablation studies. We describe experimental settings in Section IV-A. Then we validate PseMix using three different WSI datasets and compare it with other mixing data augmentation methods in Section IV-B. In Section IV-C and IV-D, we further show the advantages of PseMix in generalization and robustness. Finally, ablation studies and hyper-parameter sensitivity analysis are presented in Section IV-E and IV-F, respectively.

A. Experimental Settings

1) Datasets and Tasks: We use the following pathology datasets in this study: i) TCGA-BRCA, the subtyping of Invasive Ductal Carcinoma and Invasive Lobular Carcinoma for invasive breast carcinoma, ii) TCGA-LUNG, the recognition of Lung Adenocarcinoma and Lung Squamous Cell Carcinoma in Non-Small Cell Lung Carcinoma, and iii) TCGA-RCC, the classification of Clear Cell, Papillary, and Chromophobe for Renal Cell Carcinoma. These three datasets are often adopted to evaluate MIL models [9], [19]. They are publicly-available at TCGA (<https://portal.gdc.cancer.gov>). As shown in Table II, a total of 2,740 slides are collected. They are preprocessed with the tools developed by CLAM [7].

2) Performance Evaluation: Due to the relatively small size of WSI datasets, we follow the study design of other MIL-based WSI classification works by using 4-fold cross-validation (cv) to evaluate model performance more accurately and reliably. In each of 4 training sessions, a validation set is randomly split from training data for early stopping and model selection, resulting in a ratio of 65:10:25 for the size of training, validation, and test sets on the patient level. The two classification metrics, Area Under the Curve (AUC) and accuracy (ACC), are used. And we report their mean and standard deviation across 4 folds. For ACC calculation, the class with the largest probability is taken as classification prediction. In addition, we further assess the generalization ability of models by measuring the AUC and loss gap between training and test set, following [20] and [21].

3) Implementation Details: We set α to 1 by default, which means that λ is distributed uniformly over $[0, 1]$. For the hyper-parameters of pseudo-bag division, we empirically set $n = 30$ and $l = k = 8$ by default across all used datasets and networks. We will analyze important hyper-parameters in Section IV-F. All experiments are run on a machine with two NVIDIA GeForce RTX 3090 GPUs. Please refer to our codes (<https://github.com/liupe101/PseMix>) for more details.

B. Basic Classification Performance

We first verify whether PseMix could help improve the classification performance of MIL models. The models trained without any data augmentation, *i.e.*, vanilla models, are taken as the baseline for comparison. Moreover, we run other input mixing methods (see Table I) for further comparisons:

- *ReMix* [33], which mixes the prototypes of two bags but only processes the bags within the same class;
- *Mixup* [26], the original interpolation-based Mixup, in which two bags are aligned in instance number before interpolation by random dropping instances from the bag with larger instance number;
- *RankMix* [34], an improved interpolation-based one, in which the instances of each bag are first ranked in order according to their attention scores, and then two bags are aligned in number by dropping the instances with lower score from the bag with larger instance number;
- *InstanceMix*, an instance-level Mixup baseline, in which each bag is not divided into pseudo-bags and two input bags are mixed after randomly masking some instances, rather than pseudo-bags, according to the λ with Beta distribution, specially designed for the comparison to our pseudo-bag-level scheme.

Besides data augmentation, other regular training settings are shared for all baselines and networks.

There are two findings in Table III. (1) PseMix almost always helps improve the performance of MIL models in WSI classification. Its average improvements over vanilla models are consistently positive across three MIL networks, ranging from 0.93% to 1.75%. (2) PseMix could often surpass other related input mixing methods in average ACC and AUC. Moreover, the maximum gap in average ACC between PseMix and others is 1.01% (when it is on DSMIL); as in average AUC, the maximum is 0.90% (when it is on ABMIL). These results suggest that our pseudo-bag Mixup scheme is an effective and superior Mixup variant for improving the performance of MIL models in WSI classification.

As described in Section III, our PseMix is a masking-based mixing, pseudo-bag-level Mixup scheme. It outperforms two interpolation-based variants and a masking-based instance-level variant in most cases. Moreover, it is superior to them in average metrics for three MIL networks. (1) For two interpolation-based variants, *i.e.*, original Mixup and its improved RankMix, both of them often need to drop instances from the bag with more instances, to align two irregular bags for interpolation. This operation may result in artificial information loss, especially when there is a big gap in instance numbers. Furthermore, interpolation-based Mixup often performs worse than masking-based ones in image classification, since interpolation operations equally treat the background and foreground in images, as demonstrated in [29]. By contrast, our PseMix does not rely on interpolation but masks pseudo-bags for bag mixing. (2) For InstanceMix, it is based on masking like PseMix but operates at the instance level. And it treats all instances equally, without the critical semantic alignment for Mixup. Our PseMix tackles this by manipulating pseudo-bags.

TABLE III
CLASSIFICATION PERFORMANCE OF VANILLA MIL MODELS AND PSEMIX-BASED ONES ON THREE WSI DATASETS

Network	Method	TCGA-BRCA		TCGA-LUNG		TCGA-RCC		Average (%)	
		ACC (%)	AUC (%)	ACC (%)	AUC (%)	ACC (%)	AUC (%)	ACC	AUC
ABMIL [18]	vanilla	86.34 ± 1.97	87.05 ± 2.97	83.88 ± 4.94	92.23 ± 2.81	88.90 ± 1.20	97.36 ± 1.10	86.37	92.21
	w/ ReMix	86.46 ± 1.67	87.74 ± 3.97	84.48 ± 3.99	91.66 ± 2.04	88.24 ± 1.99	96.93 ± 0.90	86.39	92.11
	w/ Mixup	85.91 ± 2.09	87.31 ± 3.54	86.02 ± 2.67	92.75 ± 2.47	90.39 ± 2.05	97.78 ± 0.76	87.44	92.61
	w/ RankMix	85.62 ± 1.34	87.48 ± 3.97	84.73 ± 1.70	91.76 ± 2.07	89.44 ± 2.00	97.37 ± 0.68	86.60	92.20
	w/ InstanceMix	86.24 ± 1.94	87.98 ± 3.18	84.83 ± 2.76	91.74 ± 2.22	88.73 ± 1.96	97.60 ± 0.41	86.60	92.44
	w/ PseMix (ours)	86.64 ± 3.11	89.49 ± 3.69	86.45 ± 3.28	93.01 ± 2.06	90.50 ± 1.66	98.02 ± 0.55	87.86	93.51
	Δ over vanilla	+ 0.30	+ 2.44	+ 2.57	+ 0.78	+ 1.60	+ 0.66	+ 1.49	+ 1.30
DSMIL [8]	vanilla	86.75 ± 1.48	87.73 ± 2.04	85.70 ± 3.18	92.99 ± 2.94	89.81 ± 3.29	97.65 ± 0.81	87.42	92.79
	w/ ReMix	85.74 ± 2.38	87.98 ± 3.59	83.30 ± 3.71	91.59 ± 2.06	87.49 ± 1.30	96.54 ± 1.74	85.51	92.04
	w/ Mixup	86.35 ± 2.26	88.30 ± 3.10	87.52 ± 3.12	94.22 ± 2.39	90.60 ± 2.29	97.77 ± 0.85	88.16	93.43
	w/ RankMix	84.80 ± 1.73	86.49 ± 2.58	86.39 ± 3.33	93.51 ± 2.69	90.27 ± 1.25	97.38 ± 0.87	87.15	92.46
	w/ InstanceMix	85.39 ± 3.23	87.94 ± 3.48	86.45 ± 3.39	92.40 ± 2.09	89.48 ± 2.28	97.54 ± 0.55	87.11	92.63
	w/ PseMix (ours)	88.22 ± 2.65	89.65 ± 3.19	88.68 ± 2.19	93.92 ± 2.19	90.62 ± 1.20	97.89 ± 0.44	89.17	93.82
	Δ over vanilla	+ 1.47	+ 1.92	+ 2.98	+ 0.93	+ 0.81	+ 0.24	+ 1.75	+ 1.03
TransMIL [9]	vanilla	85.31 ± 0.65	88.83 ± 1.37	85.31 ± 4.01	92.14 ± 2.56	90.61 ± 1.62	97.88 ± 0.80	87.08	92.95
	w/ ReMix	79.88 ± 2.12	78.76 ± 4.17	83.27 ± 3.05	90.67 ± 3.02	87.52 ± 2.53	96.60 ± 1.04	83.56	88.68
	w/ Mixup	81.52 ± 2.39	88.41 ± 3.07	87.62 ± 3.76	94.52 ± 2.49	91.03 ± 1.60	97.88 ± 0.64	86.72	93.60
	w/ RankMix	84.69 ± 1.13	87.73 ± 1.31	86.45 ± 2.66	93.75 ± 2.54	90.63 ± 1.92	97.97 ± 0.51	87.26	93.15
	w/ InstanceMix	86.77 ± 2.35	89.44 ± 3.01	86.74 ± 2.25	92.72 ± 2.34	91.04 ± 1.57	97.84 ± 0.31	88.18	93.33
	w/ PseMix (ours)	86.98 ± 1.47	90.40 ± 2.29	87.67 ± 2.80	93.47 ± 1.91	91.14 ± 1.94	97.76 ± 0.69	88.60	93.88
	Δ over vanilla	+ 1.67	+ 1.57	+ 2.36	+ 1.33	+ 0.53	- 0.12	+ 1.52	+ 0.93

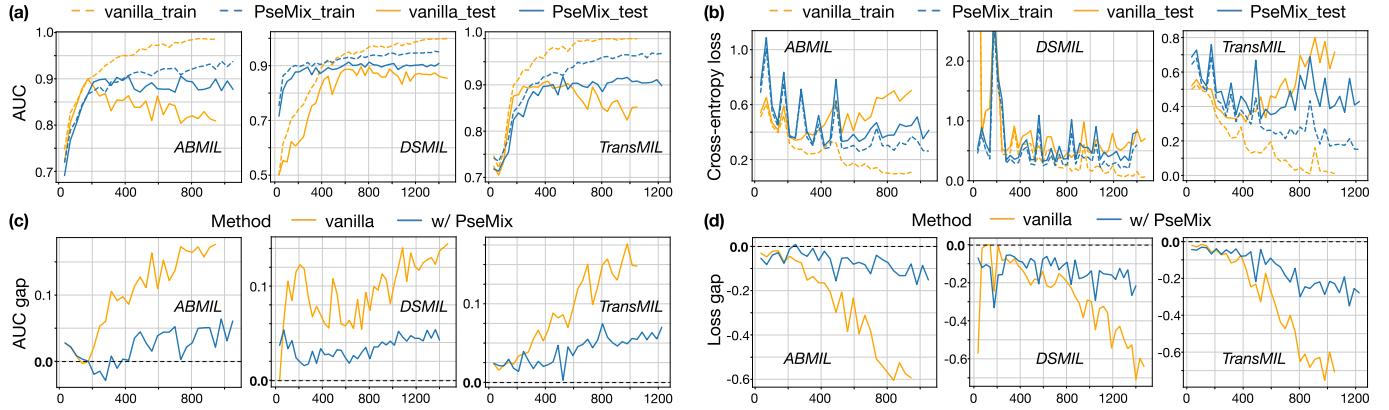


Fig. 4. Generalization gap (AUC and cross-entropy loss) between training and test set. Three MIL networks are trained on TCGA-BRCA.

For those reasons, PseMix could often perform better than previous Mixup variants in WSI classification.

C. Special Generalization Tests

Most previous works of WSI classification usually focus on promoting classification performance, rarely concerned with more aspects of model generalization. Here we demonstrate more advantages of PseMix in generalization by using generalization gap metrics and out-of-distribution data.

1) **Generalization Gap:** Following [20] and [21], we measure the generalization gap between training and test sets to show the other metrics of generalization performance. We have the following observations from Figure 4. (1) Vanilla models always perform better on training set but worse on test set in training. (2) By contrast, their PseMix-based counterparts often have lower test losses and higher test AUCs, showing smaller generalization gaps. An intuitive, possible explanation is that PseMix enables MIL models to learn from the

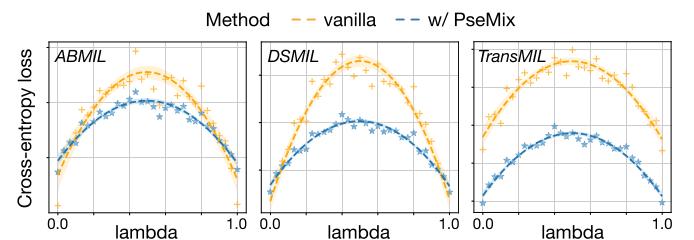


Fig. 5. Cross-entropy loss of in-between training data and their corresponding soft-labels.

new training samples drawn from vicinity distribution and these new vicinity samples could help models to expand generalization boundary, leading to the better performance of PseMix-based models on unseen data.

2) **Test on in-Between Training Data:** We further examine the model performance on out-of-distribution in-between data, following Mixup [26]. Specifically, we synthesize new in-between data for testing by applying Equation 4 to training

TABLE IV
ROBUSTNESS AGAINST PATCH OCCLUSION

Patch masking	Method	ABMIL		DSMIL		TransMIL		Average (%)	
		ACC (%)	AUC (%)	ACC (%)	AUC (%)	ACC (%)	AUC (%)	ACC	AUC
20%	vanilla	85.60 \pm 2.14	87.13 \pm 2.65	85.11 \pm 1.87	86.94 \pm 2.99	81.93 \pm 2.44	87.51 \pm 1.49	84.21	87.19
	w/ PseMix	87.27 \pm 3.01	89.91 \pm 3.05	86.96 \pm 2.32	89.12 \pm 3.90	87.07 \pm 1.78	90.25 \pm 2.53	87.10	89.76
40%	vanilla	84.65 \pm 3.02	86.86 \pm 2.56	85.95 \pm 2.03	87.07 \pm 3.41	80.57 \pm 4.20	86.28 \pm 1.98	83.72	86.74
	w/ PseMix	86.65 \pm 2.63	89.87 \pm 2.99	85.50 \pm 2.77	88.77 \pm 3.94	84.77 \pm 3.63	89.61 \pm 2.02	85.64	89.42
60%	vanilla	85.17 \pm 3.44	87.05 \pm 2.57	84.57 \pm 2.25	86.94 \pm 3.65	76.43 \pm 10.16	84.26 \pm 2.85	82.06	86.08
	w/ PseMix	86.65 \pm 2.63	89.64 \pm 3.04	84.77 \pm 2.58	88.85 \pm 4.07	82.27 \pm 6.11	88.02 \pm 2.27	84.56	88.84
80%	vanilla	84.44 \pm 2.85	87.05 \pm 1.88	84.26 \pm 2.51	86.43 \pm 3.27	64.44 \pm 19.91	76.82 \pm 4.90	77.71	83.43
	w/ PseMix	85.06 \pm 5.09	88.88 \pm 3.43	80.71 \pm 4.31	87.28 \pm 4.71	78.92 \pm 5.80	84.43 \pm 6.18	81.56	86.86

TABLE V
ROBUSTNESS AGAINST LABEL CORRUPTION

Checkpoint	Label corruption	Method	ABMIL		DSMIL		TransMIL		Average (%)	
			ACC (%)	AUC (%)	ACC (%)	AUC (%)	ACC (%)	AUC (%)	ACC	AUC
Best epoch	20%	vanilla	84.55 \pm 2.84	83.81 \pm 4.83	83.51 \pm 2.00	83.84 \pm 4.20	81.96 \pm 1.16	82.32 \pm 2.74	83.34	83.32
		w/ PseMix	84.54 \pm 3.25	83.67 \pm 4.66	85.80 \pm 3.55	85.27 \pm 4.93	84.87 \pm 1.91	86.41 \pm 3.98	85.07	85.12
	50%	vanilla	80.82 \pm 1.77	76.77 \pm 4.15	79.96 \pm 1.57	75.71 \pm 3.95	79.85 \pm 0.15	74.54 \pm 5.60	80.21	75.67
Last epoch	50%	w/ PseMix	81.35 \pm 2.75	78.72 \pm 2.33	83.62 \pm 1.96	79.94 \pm 4.89	81.95 \pm 0.46	81.61 \pm 2.75	82.31	80.09
	80%	vanilla	78.28 \pm 0.99	62.81 \pm 7.83	79.02 \pm 1.10	59.40 \pm 2.86	74.99 \pm 3.98	53.92 \pm 4.25	77.43	58.71
		w/ PseMix	78.28 \pm 0.83	61.13 \pm 8.78	79.75 \pm 1.19	64.81 \pm 2.77	77.53 \pm 2.89	59.76 \pm 6.02	78.52	61.90
Last epoch	20%	vanilla	77.61 \pm 4.23	77.07 \pm 5.42	82.05 \pm 1.17	80.91 \pm 4.12	76.58 \pm 2.20	77.31 \pm 2.68	78.75	78.43
		w/ PseMix	81.20 \pm 2.29	80.53 \pm 7.14	83.74 \pm 4.66	82.71 \pm 7.41	82.05 \pm 1.31	82.52 \pm 5.29	82.33	81.92
	50%	vanilla	75.34 \pm 3.05	70.01 \pm 5.81	75.43 \pm 3.48	73.04 \pm 4.11	70.35 \pm 3.77	68.94 \pm 3.37	73.71	70.66
Last epoch		w/ PseMix	79.05 \pm 5.08	75.14 \pm 4.12	82.78 \pm 0.90	78.08 \pm 4.07	78.82 \pm 2.11	75.51 \pm 3.72	80.22	76.24
	80%	vanilla	58.03 \pm 10.23	60.34 \pm 5.37	55.75 \pm 14.53	58.63 \pm 9.27	53.67 \pm 7.72	51.20 \pm 3.45	55.82	56.72
		w/ PseMix	57.53 \pm 9.73	60.16 \pm 7.76	65.99 \pm 11.58	63.82 \pm 3.65	73.81 \pm 6.00	57.82 \pm 9.21	65.78	60.60

samples, and obtain their soft labels by the target mixing of Equation 6. Then, we get the predictions of in-between data through the models trained before, *i.e.*, those presented in Table III, and calculate a prediction loss by measuring the cross-entropy loss between those predictions and soft labels. From the test results shown in Figure 5, we could see that i) the cross-entropy loss on mixed data would become higher when the mix ratio λ approaches 0.5, *i.e.*, when making mixed bags far from both of two input bags; ii) it is clear that the models trained with PseMix often show lower test losses on out-of-distribution data points (*i.e.*, $0.0 < \lambda < 1.0$). The second observation is consistent with our intuitive explanation to PseMix, *i.e.*, PseMix-based models often perform better on mixed in-between data, suggesting that they have learned from new vicinity samples.

D. Special Robustness Tests

Here we further demonstrate yet another benefit of our PseMix data augmentation scheme. TCGA-BRCA is used in this test.

1) Robustness Against Patch Occlusion: We randomly mask the instances of bags of test set samples and then use them for testing. From Table IV, we observe that PseMix-based models almost always obtain clear improvements over vanilla ones in various masking ratios. These improvements range from 1.92% to 3.85% and 2.57% to 3.43% in average ACC and AUC, apparently larger than those gains shown in Table III.

These facts suggest that our PseMix also could help models in the robustness to patch occlusion.

2) Robustness Against Label Corruption: To evaluate the model's robustness to label noise, we use label-corrupted training samples to train MIL models. Specifically, some training samples are randomly selected and then tagged with any one random class label, following [26]. Results are reported on original test sets using the best (at the epoch with minimum validation loss) and the last models (at the last epoch). We specially report performance for both the best model and the last one to assess the performance gap between them, so as to study the negative effect of label-corrupted training data on model performance. By comparing that gap between a vanilla model and its PseMix-based counterpart, we could infer that the model with a smaller gap may be more resistant to label-corrupted samples and has a more stable training process, *i.e.*, it is more robust to label-corrupted training samples.

As shown in Table V, (1) we find that the models trained with our PseMix often show great performance gains, 7 out of 9 cases for the best epoch and 8 out of 9 cases for the last one. Moreover, in average performance, PseMix-based models consistently exceed vanilla ones by large margins (1.80% \sim 5.58% in AUC and 1.09% \sim 9.96% in ACC). Furthermore, (2) we notice that vanilla models often suffer from greater performance drops than PseMix-based ones, when changing the epoch from best to last. For instance, a vanilla ABMIL model drops by 6.94% in accuracy when

TABLE VI

DIFFERENT METHODS OF PSEUDO-BAG DIVISION. *FT* INDICATES FINE-TUNING. \bar{T} IS THE AVERAGE TIME COST PER SLIDE FOR DIVISION

Network	Division method	TCGA-BRCA		TCGA-LUNG		TCGA-RCC		Average (%)		\bar{T} ($\times 10^{-2}s$)
		ACC (%)	AUC (%)	ACC (%)	AUC (%)	ACC (%)	AUC (%)	ACC	AUC	
ABMIL	random	88.01 \pm 2.53	89.30 \pm 3.26	86.06 \pm 2.71	91.77 \pm 2.16	89.55 \pm 1.82	97.68 \pm 0.46	87.87	92.92	0.016
	K-means	86.97 \pm 1.82	88.47 \pm 2.35	85.21 \pm 2.49	91.87 \pm 1.92	89.95 \pm 1.77	97.77 \pm 0.46	87.38	92.70	154.329
	prototype	86.33 \pm 3.30	89.18 \pm 3.36	86.06 \pm 2.35	91.86 \pm 2.20	89.66 \pm 2.32	97.74 \pm 0.46	87.35	92.93	0.129
	prototype + <i>FT</i>	87.50 \pm 2.37	89.00 \pm 3.74	86.32 \pm 2.76	91.81 \pm 2.40	90.08 \pm 2.11	97.73 \pm 0.47	87.97	92.85	0.382
DSMIL	random	86.45 \pm 1.76	89.33 \pm 3.50	85.80 \pm 3.19	93.12 \pm 1.93	88.93 \pm 1.26	97.60 \pm 0.37	87.06	93.35	0.016
	K-means	84.45 \pm 2.97	87.53 \pm 4.11	86.85 \pm 2.31	92.91 \pm 1.26	88.92 \pm 1.26	97.50 \pm 0.54	86.74	92.65	154.329
	prototype	86.25 \pm 2.33	89.57 \pm 3.50	86.06 \pm 3.79	92.65 \pm 1.95	88.62 \pm 2.18	97.41 \pm 0.61	86.98	93.21	0.129
	prototype + <i>FT</i>	87.28 \pm 1.83	89.72 \pm 3.65	85.91 \pm 2.77	92.89 \pm 1.60	89.56 \pm 2.38	97.64 \pm 0.54	87.58	93.42	0.382
TransMIL	random	86.67 \pm 1.93	88.40 \pm 2.37	84.31 \pm 2.53	91.90 \pm 1.74	90.53 \pm 1.97	97.82 \pm 0.19	87.17	92.71	0.016
	K-means	87.31 \pm 1.80	89.18 \pm 2.48	85.71 \pm 2.41	92.65 \pm 2.24	89.89 \pm 1.89	97.78 \pm 0.33	87.64	93.20	154.329
	prototype	86.24 \pm 2.32	88.78 \pm 2.02	85.72 \pm 1.54	92.09 \pm 0.80	89.86 \pm 2.81	97.75 \pm 0.27	87.27	92.87	0.129
	prototype + <i>FT</i>	86.65 \pm 2.26	88.50 \pm 3.52	86.22 \pm 1.47	92.63 \pm 1.63	90.83 \pm 1.08	97.78 \pm 0.23	87.90	92.97	0.382

TABLE VII
DIFFERENT TARGET MIXING STRATEGIES. MR MEANS MIXING RATIO

Network	λ of target mixing	TCGA-BRCA		TCGA-LUNG		TCGA-RCC		Average (%)	
		ACC (%)	AUC (%)	ACC (%)	AUC (%)	ACC (%)	AUC (%)	ACC	AUC
ABMIL	instance MR	86.76 \pm 1.85	87.99 \pm 2.31	85.00 \pm 2.75	91.67 \pm 2.14	90.18 \pm 1.32	97.68 \pm 0.42	87.31	92.45
	pseudo-bag MR	87.50 \pm 2.37	89.00 \pm 3.74	86.32 \pm 2.76	91.81 \pm 2.40	90.08 \pm 2.11	97.73 \pm 0.47	87.97	92.85
DSMIL	instance MR	85.83 \pm 2.07	88.49 \pm 3.39	86.57 \pm 2.97	92.83 \pm 2.10	89.99 \pm 2.70	97.46 \pm 0.67	87.46	92.93
	pseudo-bag MR	87.28 \pm 1.83	89.72 \pm 3.65	85.91 \pm 2.77	92.89 \pm 1.60	89.56 \pm 2.38	97.64 \pm 0.54	87.58	93.42
TransMIL	instance MR	87.28 \pm 1.88	89.05 \pm 2.97	85.62 \pm 2.06	92.94 \pm 2.26	90.17 \pm 2.09	97.68 \pm 0.33	87.69	93.23
	pseudo-bag MR	86.65 \pm 2.26	88.50 \pm 3.52	86.22 \pm 1.47	92.63 \pm 1.63	90.83 \pm 1.08	97.78 \pm 0.23	87.90	92.97

turning to use the last model, for a 20% corruption ratio; whereas its PseMix-based counterpart only drops by 3.34% in the same case. These observations tell us that our PseMix could often make model training more stable and help MIL models to be more robust to label noise in training.

E. Ablation Study

1) *Study on Different Pseudo-Bag Division Methods*: As described in Section III-B, different methods could be adopted to generate pseudo-bags, such as random and phenotype-stratified sampling. Here we compare our adopted method, *i.e.*, bag-prototype-based clustering + phenotype fine-tuning (prototype + *FT*), with the other three, random sampling, K-means-based and bag-prototype-based stratified sampling. We set $p = 1$ for this study. Test results are presented in Table VI.

These test results indicate that our prototype + *FT* is competitive in four pseudo-bag division methods. Our evidences have three-fold. (1) prototype + *FT* could often obtain better overall performances than the other three, especially in average ACC. (2) It takes three orders of magnitude less time than the K-means-based one that also utilizes stratified sampling. (3) Compared to random and bag-prototype-based division, it merely introduces moderate time costs for phenotype clustering or fine-tuning but performs better in most cases.

From the results of Table VI, we notice that compared with our prototype + *FT*, a K-means-based method often shows a negative influence on PseMix data augmentation. One possible reason behind this is that the initial clusters obtained by bag-prototype-based clustering could help to yield

a clearer phenotype hierarchy than those obtained by K-means. A clearer phenotype hierarchy could lead to the pseudo-bags that are more consistent with their parent bag in phenotype distribution, contributing to the semantic alignment in target mixing and thus improving the end performance of PseMix, as discussed in Section III-B.1. Specifically, our initial clusters are determined based on the similarity measurement between instances and bag prototype, and this similarity measurement is often helpful for differentiating instance phenotype [42]. By contrast, classical K-means ignores the characteristic of pathology WSIs to select initial cluster centers by random sampling. Moreover, its performance is known to be sensitive to initial clusters. As a result, it often takes a significantly longer time for convergence and yields a phenotype hierarchy worse than prototype + *FT*.

2) *Study on Different Target Mixing Strategies*: Apart from the semantic alignment strategy based on pseudo-bag mixing ratio (called pseudo-bag MR), another strategy based on instance mixing ratio (called instance MR) could also be used for target mixing. Specifically, in $\tilde{y} = \lambda y_A + (1 - \lambda)y_B$, λ could be determined according to the ratio of the number of instances from A and B in a mixed bag \tilde{X} , instead of the ratio of the number of pseudo-bags. We set $p = 1$ in this experiment to avoid the potential influence of random mixing.

From the comparative results of Table VII, we could see that the strategy of pseudo-bag MR leads to better overall performance than its instance MR-based counterpart, in 5 of 6 comparisons. These empirical results suggest that using a pseudo-bag MR as the parameter of target mixing (λ) could be better for semantic alignment than using an instance MR.

TABLE VIII

ABLATION STUDY ON OUR PSEMIX. PB INDICATES ONLY USING PSEUDO-BAGS FOR DATA AUGMENTATION. FT, MIXUP, AND R-MIX MEANS OUR PHENOTYPE FINE-TUNING, PSEUDO-BAG-BASED MIXUP, AND RANDOM MIXING, RESPECTIVELY

Method	PB	FT	Component		TCGA-BRCA		TCGA-LUNG		TCGA-RCC		Average (%)	
			Mixup	R-mix	ACC (%)	AUC (%)	ACC (%)	AUC (%)	ACC (%)	AUC (%)	ACC	AUC
- ABMIL network												
Baseline	✓				86.43 ± 2.70	87.23 ± 2.04	85.20 ± 2.82	92.84 ± 1.94	91.36 ± 0.81	97.82 ± 0.62	87.66	92.63
PseMix	✓	✓			85.71 ± 2.25	87.39 ± 2.38	85.11 ± 2.60	92.19 ± 1.87	90.38 ± 0.84	97.94 ± 0.74	87.07	92.51
	✓		✓		86.33 ± 3.30	89.18 ± 3.36	86.06 ± 2.35	91.86 ± 2.20	89.66 ± 2.32	97.74 ± 0.46	87.35	92.93
	✓	✓	✓		87.50 ± 2.37	89.00 ± 3.74	86.32 ± 2.76	91.81 ± 2.40	90.08 ± 2.11	97.73 ± 0.47	87.97	92.85
	✓	✓	✓	✓	86.64 ± 3.11	89.49 ± 3.69	86.45 ± 3.28	93.01 ± 2.06	90.50 ± 1.66	98.02 ± 0.55	87.86	93.51
- DSMIL network												
Baseline	✓				86.13 ± 2.56	87.97 ± 3.12	87.16 ± 3.00	94.07 ± 2.74	89.95 ± 2.08	97.76 ± 0.77	87.75	93.27
PseMix	✓	✓			85.72 ± 1.63	89.31 ± 2.83	86.96 ± 2.91	93.57 ± 2.49	90.50 ± 1.79	97.93 ± 0.48	87.73	93.60
	✓		✓		86.25 ± 2.33	89.57 ± 3.50	86.06 ± 3.79	92.65 ± 1.95	88.62 ± 2.18	97.41 ± 0.61	86.98	93.21
	✓	✓	✓		87.28 ± 1.83	89.72 ± 3.64	85.91 ± 2.77	92.89 ± 1.60	89.56 ± 2.38	97.64 ± 0.54	87.58	93.42
	✓	✓	✓	✓	88.22 ± 2.65	89.65 ± 3.19	88.68 ± 2.19	93.92 ± 2.19	90.62 ± 1.20	97.89 ± 0.44	89.17	93.82
- TransMIL network												
Baseline	✓				86.66 ± 1.10	89.51 ± 1.29	87.50 ± 2.88	93.90 ± 1.79	89.34 ± 1.71	97.65 ± 1.05	87.83	93.69
PseMix	✓	✓			86.56 ± 0.91	89.31 ± 1.22	87.13 ± 3.37	93.33 ± 1.61	90.16 ± 1.82	97.69 ± 0.94	87.95	93.44
	✓		✓		86.24 ± 2.32	88.78 ± 2.02	85.72 ± 1.54	92.09 ± 0.80	89.86 ± 2.81	97.75 ± 0.27	87.27	92.87
	✓	✓	✓		86.65 ± 2.26	88.50 ± 3.52	86.22 ± 1.47	92.63 ± 1.63	90.83 ± 1.08	97.78 ± 0.23	87.90	92.97
	✓	✓	✓	✓	86.98 ± 1.47	90.40 ± 2.29	87.67 ± 2.80	93.47 ± 1.91	91.14 ± 1.94	97.76 ± 0.69	88.60	93.88

TABLE IX
DIFFERENT PATCH FEATURE EXTRACTORS

SSL-based feature extractor	Method	ABMIL		DSMIL		TransMIL		Average (%)	
		ACC (%)	AUC (%)	ACC (%)	AUC (%)	ACC (%)	AUC (%)	ACC	AUC
HIPT	vanilla	84.89 ± 2.44	86.21 ± 2.22	82.50 ± 2.04	83.31 ± 2.88	82.80 ± 2.48	83.88 ± 3.55	83.40	84.47
	w/ PseMix	85.09 ± 2.39	86.39 ± 2.85	85.50 ± 2.82	87.00 ± 2.92	84.89 ± 2.06	84.94 ± 3.44	85.16	86.11
CTransPath	vanilla	90.02 ± 1.99	92.83 ± 1.75	88.77 ± 1.74	92.58 ± 1.59	90.44 ± 1.46	92.97 ± 1.42	89.74	92.79
	w/ PseMix	90.85 ± 1.56	93.66 ± 2.11	89.28 ± 1.44	92.62 ± 2.18	91.38 ± 1.59	93.54 ± 2.02	90.50	93.27

This is largely caused by the characteristic of pseudo-bags, namely, pseudo-bags could inherit labels from their parent bag, which is the basic assumption of pseudo-bag. Due to this basic assumption, pseudo-bag is more likely to be aligned with its parent bag in semantics, compared with instance. Accordingly, utilizing the pseudo-bag MR for target mixing could often result in a better alignment of \tilde{X} with \tilde{y} in semantics.

3) Study on Our PseMix: On the basis of prior pseudo-bag-based methods, PseMix could be decomposed into these new components: a phenotype fine-tuning technique for pseudo-bag division (termed FT), pseudo-bag Mixup (termed Mixup), and a random mixing mechanism (termed R-mix). To elucidate the influence of each new component on end performance, we conduct an ablation study on PseMix. The baseline of this ablation study is a pseudo-bag augmentation method (termed PB). It generates pseudo-bags by prototype-based clustering and stratified sampling, without FT and Mixup.

From the ablation study results in Table VIII, we have three empirical findings as follows. (1) For FT, it has no obvious positive effects on pseudo-bag augmentation in terms of overall performance. However, its effectiveness on pseudo-bag Mixup is observed in 5 of 6 comparisons. These results show that pseudo-bag Mixup augmentation is more likely to benefit from phenotype fine-tuning than pseudo-bag augmentation.

(2) For pseudo-bag Mixup augmentation, it is comparable with or slightly better than pseudo-bag augmentation (when using ABMIL) in presence of FT, seen from the comparison between PB + FT+ Mixup and PB + FT. A significant performance improvement is not often observed. This result indicates that a simple and direct mixing of pseudo-bags could not often lead to better performance, reflecting the challenge of applying Mixup to heterogeneous WSI bags. (3) For our R-mix, we find that it obtains the best overall performance in 5 of 6 comparisons. This result suggests that utilizing both mixed and masked bags for data augmentation could often be more effective for MIL training than only using mixed bags (*i.e.*, pseudo-bag Mixup augmentation) or masked bags (*i.e.*, pseudo-bag augmentation). This result also demonstrates that our random mixing plays a critical role in PseMix. There are two possible reasons for this. The one is that random mixing brings more data diversity because it produces two kinds of augmented bags for training, *i.e.*, mixing and masked bags, as described in Section III-B.2. The other one is that the involvement of masked bags could make models learn from mixed bags (vicinity samples) more efficiently. Because pseudo-bag-masked bags could be cast as the intermediate samples between original training data and synthetic mixed data. These intermediate samples may enable models to learn

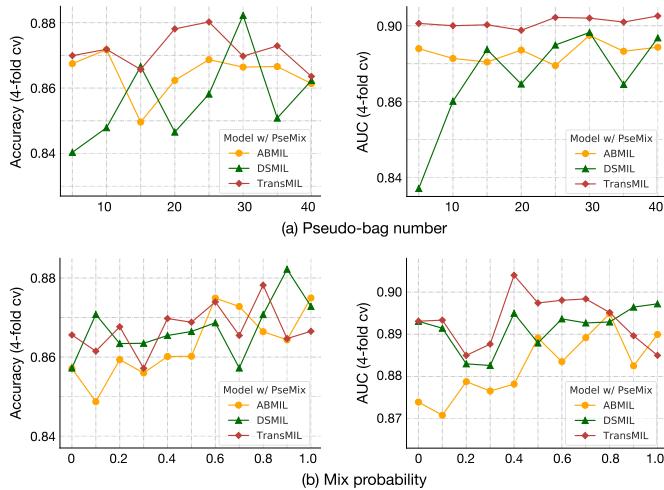


Fig. 6. Study on the two important hyper-parameters of PseMix, pseudo-bag number (n) and mix probability (p).

from vicinity samples more efficiently, by smoothing the transition from original distribution to vicinity distribution.

F. Further Analysis

Here we examine more experimental settings to understand their sensitivity to model performance. We mainly present the experimental results measured on TCGA-BRCA.

1) Other Feature Extractors: Apart from the classical ResNet-50 model adopted in [7], we also try two more advanced feature extractors. They are HIPT [19] and CTransPath [46], pre-trained with self-supervised learning on WSIs. We directly use their released model weights for feature extraction. The test results of Table IX reveal that PseMix still consistently improves the performance of vanilla models across three MIL networks, even for different feature extractors. This result suggests the adaptability of PseMix.

2) Hyper-Parameter Analysis: We test the two most important hyper-parameters in PseMix, n (the pseudo-bag number of each WSI bag) and p (the probability of our random mixing). Test results are shown in Figure 6. For n , a value near our default setting ($n = 30$) tends to obtain better overall performance. All the hyper-parameters of pseudo-bag division are set by default across all datasets and networks without tuning. For p , a larger value is likely to result in better test performances. We set p to 0.8, 0.9, and 0.4 for ABMIL, DSMIL, and TransMIL, respectively, based on their respective losses on validation sets; the other values of p could obtain better results on test sets.

3) Combining With Basic Data Augmentation: Since some basic image augmentation methods could also be adopted for WSI data, we combine our PseMix with them to further examine its adaptability. Specifically, we follow [36] to adopt two basic image augmentation methods, vertical flip and stain augmentation, for WSI patches. In MIL training, the patch features of each WSI are augmented by them with a probability of 50%, in which each basic method is chosen with an equal probability. The model trained with basic augmentation methods is the baseline used for comparison. From the results of

TABLE X
COMBINING PSEMIX WITH BASIC DATA AUGMENTATION METHODS

Network	Data augmentation	ACC (%)	AUC (%)
ABMIL	baseline	85.70 ± 2.20	88.35 ± 2.80
	baseline + PseMix	87.49 ± 2.32	89.34 ± 3.27
DSMIL	baseline	87.92 ± 1.01	90.02 ± 2.09
	baseline + PseMix	88.86 ± 2.12	90.04 ± 3.39
TransMIL	baseline	85.09 ± 1.94	89.15 ± 2.78
	baseline + PseMix	86.97 ± 1.77	89.56 ± 3.32

Table X, we find that our PseMix augmentation still improves the performance of MIL models in the presence of basic image augmentation. This experiment could further verify the adaptability of PseMix.

V. CONCLUSION

This paper proposes a Pseudo-bag Mixup (PseMix) data augmentation scheme for MIL-based WSI classification. It utilizes a pseudo-bag concept to fulfill Mixup alignment, thereby generalizing the basic idea of Mixup from common images to special WSIs. This scheme is compatible with most prevalent MIL networks. Moreover, it is efficient and plugin-and-play, neither involving time-consuming operations nor relying on the prediction of MIL models. Comparative experiments and ablation studies confirm that PseMix is an effective Mixup variant for WSI classification. It could often improve the performance of MIL models and obtain better overall performance than other related mixing strategies. Other than that, it is observed that MIL models could often benefit more from PseMix in many other notable aspects, such as generalization gap, in-between data generalization, patch occlusion robustness, and label noise robustness. In the future, our PseMix could serve as a promising data augmentation method to help develop the WSI classification models with better generalization and robustness for clinical pathology diagnosis.

REFERENCES

- [1] K. Bera, K. A. Schalper, D. L. Rimm, V. Velcheti, and A. Madabhushi, “Artificial intelligence in digital pathology—New tools for diagnosis and precision oncology,” *Nature Rev. Clin. Oncol.*, vol. 16, no. 11, pp. 703–715, Nov. 2019.
- [2] M. K. K. Niazi, A. V. Parwani, and M. N. Gurcan, “Digital pathology and artificial intelligence,” *Lancet Oncol.*, vol. 20, no. 5, pp. 253–261, 2019.
- [3] M. D. Zarella et al., “A practical guide to whole slide imaging: A white paper from the digital pathology association,” *Arch. Pathol. Lab. Med.*, vol. 143, no. 2, pp. 222–234, Feb. 2019.
- [4] C. Mercan, S. Aksoy, E. Mercan, L. G. Shapiro, D. L. Weaver, and J. G. Elmore, “Multi-instance multi-label learning for multi-class classification of whole slide breast histopathology images,” *IEEE Trans. Med. Imag.*, vol. 37, no. 1, pp. 316–325, Jan. 2018.
- [5] G. Campanella et al., “Clinical-grade computational pathology using weakly supervised deep learning on whole slide images,” *Nature Med.*, vol. 25, no. 8, pp. 1301–1309, Aug. 2019.
- [6] W. Shao, T. Wang, Z. Huang, Z. Han, J. Zhang, and K. Huang, “Weakly supervised deep ordinal cox model for survival prediction from whole-slide pathological images,” *IEEE Trans. Med. Imag.*, vol. 40, no. 12, pp. 3739–3747, Dec. 2021.
- [7] M. Y. Lu, D. F. K. Williamson, T. Y. Chen, R. J. Chen, M. Barbieri, and F. Mahmood, “Data-efficient and weakly supervised computational pathology on whole-slide images,” *Nature Biomed. Eng.*, vol. 5, no. 6, pp. 555–570, Mar. 2021.

- [8] B. Li, Y. Li, and K. W. Eliceiri, "Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14313–14323.
- [9] Z. Shao et al., "TransMIL: Transformer based correlated multiple instance learning for whole slide image classification," in *Proc. Adv. Neural Inf. Process. Syst.*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds. 2021, pp. 1–12.
- [10] Z. Wang, L. Yu, X. Ding, X. Liao, and L. Wang, "Lymph node metastasis prediction from whole slide images with transformer-guided multiinstance learning and knowledge transfer," *IEEE Trans. Med. Imag.*, vol. 41, no. 10, pp. 2777–2787, Oct. 2022.
- [11] H. Zhang et al., "DTFD-MIL: Double-tier feature distillation multiple instance learning for histopathology whole slide image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 18780–18790.
- [12] P. Liu, L. Ji, F. Ye, and B. Fu, "AdvMIL: Adversarial multiple instance learning for the survival analysis on whole-slide images," *Med. Image Anal.*, vol. 91, Jan. 2023, Art. no. 103020.
- [13] Z. Zhu, L. Yu, W. Wu, R. Yu, D. Zhang, and L. Wang, "MuRCL: Multi-instance reinforcement contrastive learning for whole slide image classification," *IEEE Trans. Med. Imag.*, vol. 42, no. 5, pp. 1337–1348, May 2023.
- [14] Y. Zheng et al., "Kernel attention transformer for histopathology whole slide image analysis and assistant cancer diagnosis," *IEEE Trans. Med. Imag.*, vol. 42, no. 9, pp. 2726–2739, Sep. 2023.
- [15] J. Shi et al., "A structure-aware hierarchical graph-based multiple instance learning framework for PT staging in histopathological image," *IEEE Trans. Med. Imag.*, vol. 42, no. 10, pp. 3000–3011, Oct. 2023.
- [16] N. A. Koohbanani, B. Unnikrishnan, S. A. Khurram, P. Krishnaswamy, and N. Rajpoot, "Self-path: Self-supervision for classification of pathology images with limited annotations," *IEEE Trans. Med. Imag.*, vol. 40, no. 10, pp. 2845–2856, Oct. 2021.
- [17] Z. Wang, C. Saoud, S. Wangsircharoen, A. W. James, A. S. Popel, and J. Sulam, "Label cleaning multiple instance learning: Refining coarse annotations on single whole-slide images," *IEEE Trans. Med. Imag.*, vol. 41, no. 12, pp. 3952–3968, Dec. 2022.
- [18] M. Ilse, J. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2127–2136.
- [19] R. J. Chen et al., "Scaling vision transformers to gigapixel images via hierarchical self-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 16123–16134.
- [20] Y. Jiang, D. Krishnan, H. Mobahi, and S. Bengio, "Predicting the generalization gap in deep networks with margin distributions," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–19.
- [21] L. Zhang, Z. Deng, K. Kawaguchi, A. Ghorbani, and J. Zou, "How does mixup help with robustness and generalization?" in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–24.
- [22] V. N. Vapnik, *The Nature of Statistical Learning Theory*, vol. 41, no. 4. New York, NY, USA: Springer, Nov. 2000.
- [23] C. Szegedy et al., "Intriguing properties of neural networks," 2014, *arXiv:1312.6199*.
- [24] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *Proc. Int. Conf. Learn. Represent.* OpenReview.net, 2017.
- [25] Z. Li, K. Kamnitsas, and B. Glocker, "Analyzing overfitting under class imbalance in neural networks for image segmentation," *IEEE Trans. Med. Imag.*, vol. 40, no. 3, pp. 1065–1077, Mar. 2021.
- [26] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–13.
- [27] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6022–6031.
- [28] J. Kim, W. Choo, and H. O. Song, "Puzzle mix: Exploiting saliency and local statistics for optimal mixup," in *Proc. ICML*, vol. 119, 2020, pp. 5275–5285.
- [29] J.-N. Chen, S. Sun, J. He, P. Torr, A. Yuille, and S. Bai, "TransMix: Attend to mix for vision transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12125–12134.
- [30] X. Han, Z. Jiang, N. Liu, and X. Hu, "G-mixup: Graph data augmentation for graph classification," in *Proc. 39th Int. Conf. Mach. Learn.*, vol. 162, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds. Jul. 2022, pp. 8230–8248.
- [31] Y. Wang et al., "PointPatchMix: Point cloud mixing with patch scoring," 2023, *arXiv:2303.06678*.
- [32] H. Gong, G. Chen, M. Mao, Z. Li, and G. Li, "VQAMix: Conditional triplet mixup for medical visual question answering," *IEEE Trans. Med. Imag.*, vol. 41, no. 11, pp. 3332–3343, Nov. 2022.
- [33] J. Yang et al., "Remix: A general and efficient framework for multiple instance learning based whole slide image classification," in *Med. Image Comput. Comput. Assist. Intervent.*, 2022, pp. 35–45.
- [34] Y.-C. Chen and C.-S. Lu, "RankMix: Data augmentation for weakly supervised learning of classifying whole slide images with diverse sizes and imbalanced categories," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 23936–23945.
- [35] I. Zaffar, G. Jaume, N. Rajpoot, and F. Mahmood, "Embedding space augmentation for weakly supervised learning in whole-slide images," in *Proc. IEEE 20th Int. Symp. Biomedical Imag.*, Apr. 2022, pp. 1–4.
- [36] M. Kang, H. Song, S. Park, D. Yoo, and S. Pereira, "Benchmarking self-supervised learning on diverse pathology datasets," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 3344–3354.
- [37] X. Guo, C. Yang, Y. Liu, and Y. Yuan, "Learn to threshold: ThresholdNet with confidence-guided manifold mixup for polyp segmentation," *IEEE Trans. Med. Imag.*, vol. 40, no. 4, pp. 1134–1146, Apr. 2021.
- [38] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.
- [39] D. Tellez et al., "Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology," *Med. Image Anal.*, vol. 58, Dec. 2019, Art. no. 101544.
- [40] S. Cheng et al., "Robust whole slide image analysis for cervical cancer screening using deep learning," *Nature Commun.*, vol. 12, no. 1, p. 5639, Sep. 2021.
- [41] Z. Shao, L. Dai, Y. Wang, H. Wang, and Y. Zhang, "AugDiff: Diffusion based feature augmentation for multiple instance learning in whole slide image," 2023, *arXiv:2303.06371*.
- [42] R. Yang, P. Liu, and L. Ji, "ProtoDiv: Prototype-guided division of consistent pseudo-bags for whole-slide image classification," 2023, *arXiv:2304.06652*.
- [43] Y. Cui et al., "Bayes-MIL: A new probabilistic perspective on attention-based multiple instance learning for whole slide images," in *Proc. 11th Int. Conf. Learn. Represent.* OpenReview.net, 2023.
- [44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [45] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
- [46] X. Wang et al., "Transformer-based unsupervised contrastive learning for histopathological image classification," *Med. Image Anal.*, vol. 81, Oct. 2022, Art. no. 102559.