

AugDiff: Diffusion based Feature Augmentation for Multiple Instance Learning in Whole Slide Image

Zhuchen Shao¹, Liuxi Dai², Yifeng Wang², Haoqian Wang^{1*}, Yongbing Zhang^{2*}

¹Tsinghua Shenzhen International Graduate School, Tsinghua University

²Harbin Institute of Technology (Shenzhen)

shaozc0412@gmail.com, dailiuxi21@stu.hit.edu.cn, wangyifeng@stu.hit.edu.cn,
wanghaoqian@tsinghua.edu.cn, ybzhang08@hit.edu.cn

Abstract

Multiple Instance Learning (MIL), a powerful strategy for weakly supervised learning, is able to perform various prediction tasks on gigapixel Whole Slide Images (WSIs). However, the tens of thousands of patches in WSIs usually incur a vast computational burden for image augmentation, limiting the MIL model’s improvement in performance. Currently, the feature augmentation-based MIL framework is a promising solution, while existing methods such as Mixup often produce unrealistic features. To explore a more efficient and practical augmentation method, we introduce the Diffusion Model (DM) into MIL for the first time and propose a feature augmentation framework called AugDiff. Specifically, we employ the generation diversity of DM to improve the quality of feature augmentation and the step-by-step generation property to control the retention of semantic information. We conduct extensive experiments over three distinct cancer datasets, two different feature extractors, and three prevalent MIL algorithms to evaluate the performance of AugDiff. Ablation study and visualization further verify the effectiveness. Moreover, we highlight AugDiff’s higher-quality augmented feature over image augmentation and its superiority over self-supervised learning. The generalization over external datasets indicates its broader applications.

1. Introduction

Computational pathology is a research direction that combines deep learning with high-resolution, wide-field-of-view Whole Slide Images (WSIs) [37, 18]. An important thing to figure out is how to design a deep learning network for acquiring information from gigapixel images. Based only on WSI-level annotation, Multiple Instance Learning (MIL) is currently a widely used solution paradigm [22, 23, 4, 31, 6, 43].

As a method of weakly supervised learning, MIL needs

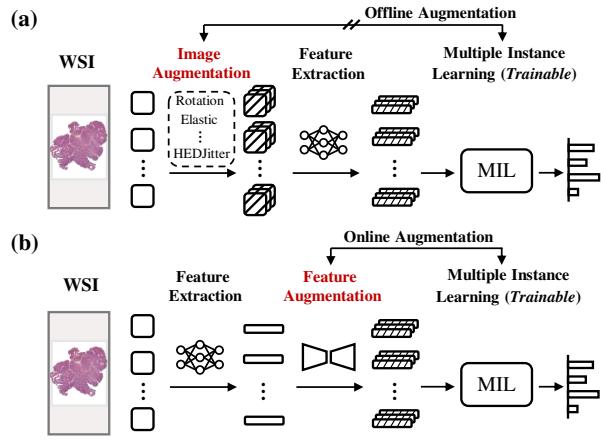


Figure 1. **Background and Motivation.** (a) The standard framework for combining MIL and WSI-related tasks: 1) The gigapixel WSI is sliced into several patches. 2) The patches are used for image augmentation before being embedded into features. 3) The training of the MIL model is supervised by WSI-level labels. It should be noted that image augmentation can only be utilized offline to assist with MIL. (b) The framework facilitates MIL training through feature augmentation. In contrast to (a), feature augmentation is integrated into MIL training online, preventing repeating feature extraction while allowing for real-time augmentation. Note that quantitative changes in the figure depict the expansion of the dataset. The training dataset for (a) is expanded in the image augmentation, whereas the training dataset for (b) is expanded in the feature augmentation.

to learn the relationship from many unlabeled instances to bag-level labels. Similarly, WSI-related tasks require aggregating numerous unlabeled patches to predict WSI-level labels. Since the tens of thousands of patches in WSIs can impose a tremendous computing burden, MIL is frequently based on aggregating patch features. Unfortunately, limited by the small number of WSI-level training data, the performance of MIL methods is always sub-optimal [30, 46]. Intuitively, image augmentation can efficiently overcome this issue, while image augmentation and feature extraction over tens of thousands of patches are time-consuming

*Corresponding authors.

and computationally demanding. As shown in Figure 1(a), the image augmentation can only facilitate MIL training offline, which cannot effectively improve feature diversity. Additionally, traditional image augmentation methods often make limited adjustments at the feature level due to the vast amount of redundant and irrelevant information in digital images [28, 14].

Some research has begun to concentrate on the framework of feature augmentation. As shown in Figure 1(b), the feature augmentation framework avoids repeated feature extraction. The extracted patch features can be repeatedly augmented in each epoch of MIL training via online augmentation. Some existing methods include the Mixup-based feature mixing framework [13, 42] and the GAN-based feature generation framework [44]. However, limited by unrealistic generation and unstable performance, current feature augmentation frameworks do not always perform better than the image augmentation framework. As a result, how to achieve the goal of faster speed and better performance within the MIL feature augmentation framework remains an urgent problem to be solved.

Currently, the Diffusion Model (DM) [16] is an innovative generative modeling solution. It outperformed GAN in various image generation tasks, and its generation is more diverse [12, 26, 36, 11]. Furthermore, the Latent Diffusion Model [28] shows that DM could also generate various features and achieve satisfactory results in downstream tasks. Inspired by this, we consider introducing the diversity-generating ability of DM into MIL. It should be noted that the feature augmentation should avoid destroying the basic semantic information, *e.g.*, the features of the normal patch ought not to be transformed into those of the tumor patch. Hence, how to flexibly adjust Diffusion for semantic information retention and feature augmentation is a crucial consideration in the framework design.

We propose a Diffusion-based feature augmentation framework, AugDiff. The main contributions of our work include the following:

1) We use the DM in combination with MIL training for the first time. Specifically, we integrate the diversity-generating capability of the DM with the online augmentation capability of the feature augmentation, which can efficiently and effectively improve MIL training.

2) We guide the training of DM with various image augmentations to assist DM in generating diverse feature augmentation. We exploit the step-by-step generation characteristic of DM to control the retention of semantic information during the augmentation sampling process.

3) We conduct extensive experiments on three distinct cancer datasets, two different feature extractors, and three prevalent MIL algorithms. Our AugDiff achieves the highest AUC metric performance compared to existing MIL augmentation frameworks. In addition, we demonstرات the

significant speed advantage of AugDiff over offline Patch Augmentation. Ablation study and visualization further verify the effectiveness of AugDiff.

4) To demonstrate the superiority of AugDiff, we illustrate the higher quality of AugDiff-augmented features than Patch Augmentation. Moreover, We highlight the superiority of image augmentation-guided AugDiff over the agent task-guided self-supervised learning framework. Also, cross-tests with different cancer datasets show that AugDiff has a solid ability to generalize.

2. Related Work

Deep learning typically requires a large amount of labeled training data to prevent network over-fitting [33]. Unfortunately, in many fields, including medical image analysis, large quantities of annotated data are not always available [9]. At present, data augmentation is a low-cost method for enhancing the quantity and quality of training data sets, which is widely applied in the field of medical image analysis [39, 17, 8].

Existing research on data augmentation mainly includes non-parametric augmentation (such as rotation, stretching, etc.) and parametric augmentation (such as GAN and VAE). For the non-parametric augmentation, some studies [38, 1] have discussed the influence of common image augmentation, such as color augmentation and normalization, on multiple WSI-related downstream tasks. For the parametric image augmentation, some studies [41, 5] have discussed the efficacy of the GAN-based image synthesis framework for downstream classification tasks.

As a typical weakly supervised learning method, the performance of MIL is usually constrained by the limited labeled datasets in WSI-related tasks. Due to the properties of gigapixels in WSI, traditional image-based data augmentation methods are typically computationally intensive and time-consuming. Consequently, some studies have investigated the combination of feature augmentation and MIL, including methods based on patch feature mixing and patch feature generation. The feature mixing methods include feature mixup [13, 42] and feature pseudo-bag construction [30, 46]. Due to a lack of learnable parameters, this type of feature augmentation method frequently results in a lack of variation in feature synthesis. Besides, the deep learning-based feature generation schemes include GAN-based feature augmentation [44]. However, an unstable feature augmentation framework can easily destroy the semantic information of the original patch features, leading to deteriorated performance of the downstream tasks. Currently, the performance of most feature augmentation methods is inferior to that of image augmentation. How to design an efficient and effective feature augmentation method that achieves better performance and higher speed than image augmentation is an important research topic.

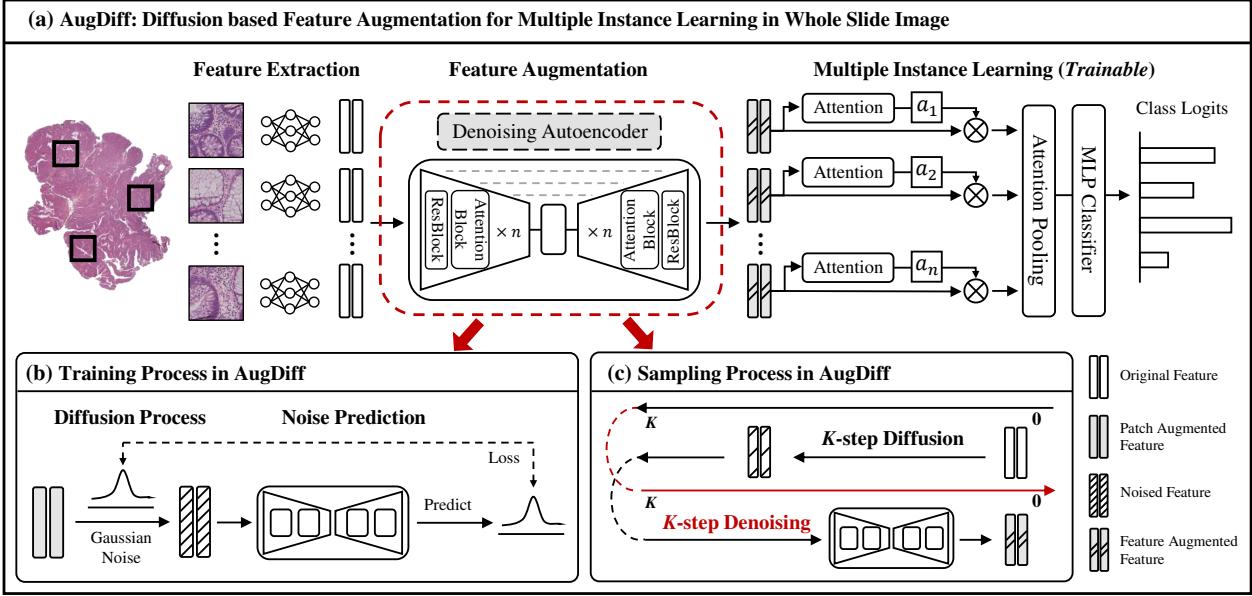


Figure 2. **Overview of AugDiff.** (a) For tasks related to WSI, the feature-based MIL framework has two steps: pre-processing and MIL model training. Pre-processing includes WSI splitting and feature extraction. Then, during MIL training, the model learns to predict bag-level labels based on all the extracted features, and this paper adds the step of feature augmentation. (b) The training process of the Denoising AutoEncoder (DAE) in AugDiff comprises both adding noise in the diffusion process and predicting noise by DAE. (c) The sampling process in the AugDiff is a two-stage feature augmentation, which includes K -step Diffusion and K -step Denoising.

3. Method

3.1. Problem Formulation

WSI is a high-resolution, wide-field-of-view image and often only has a WSI-level label. For such weakly supervised learning problems, MIL is an efficient solution. Here, each WSI is viewed as a bag, and each small patch cut from the WSI is considered an instance. It should be noted that the bag-level label is known, but the instance-level label is unknown. The performance of the MIL model in WSI-related tasks is frequently constrained by a small number of training data. Data augmentation is considered a promising method to get rid of this problem. However, every patch-based data augmentation requires unavoidable and repeated feature extraction, resulting in high computational and time costs. In addition, due to the high cost of patch augmentation, only a limited number of augmentations can be allowed, thereby limiting the variety of augmentations. Here, feature-level augmentation is used to facilitate MIL training. The whole framework is shown in Figure 2.

3.2. Diffusion Model based Feature Augmentation Framework in MIL Training

We propose AugDiff, a Diffusion-based feature augmentation framework in MIL training. This subsection will briefly introduce the Diffusion Model (DM) before focusing on the training and testing in AugDiff.

Diffusion Model. Deep neural networks can approximate the probability distribution $p(x)$ of the data. Specifically, the DM approximates the $p(x)$ by learning the reverse process of a fixed Markov Chain of length T . The forward diffusion process can be defined as successively adding Gaussian noise ϵ to the input x to produce a set of noisy samples $\{x_t\}_{t=1}^T$. The reverse process can be simplified by training a Denoising AutoEncoder (DAE), $\epsilon_\theta(x_t, t)$, to predict a denoised variant of its input x_t ($t = 1, \dots, T$). The corresponding goal is as follows:

$$L_{DM} = \mathbb{E}_{x, \epsilon \sim \mathcal{N}(0, 1), t} [\|\epsilon - \epsilon_\theta(x_t, t)\|_2^2]. \quad (1)$$

Furthermore, to be more computationally efficient, the Latent Diffusion Model (LDM) investigates training a DAE $\epsilon_\theta(z_t, t)$ at the feature level to approximate the probability distribution $p(z)$ of features. A conditional mechanism is also introduced into DAE to model the conditional distribution $p(z|y)$, where the conditional DAE can be denoted as $\epsilon_\theta(z_t, t, y)$. It influences the generation process by controlling the input condition y , which is processed using word vector mapping τ_θ from the category to the embedding vector. The corresponding goal is as follows:

$$L_{LDM} = \mathbb{E}_{z, y, \epsilon \sim \mathcal{N}(0, 1), t} [\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))\|_2^2]. \quad (2)$$

MIL training in WSI-based tasks is typically performed at the feature level due to the high computational cost.

Therefore, we use LDM to design a feature augmentation framework in MIL tasks.

Feature Augmentation based MIL Training We further demonstrate how DM can be utilized in MIL training. Specifically, 1) the diversity of the samples generated by the DM can assist MIL training in efficiently expanding the training data [12]. 2) The step-by-step generation framework of DM can have better controllability in feature augmentation [21]. We can determine how much semantic information is kept in MIL training with different feature extractors and datasets by changing the number of feature augmentation steps. Inspired by these advantages, the proposed framework is illustrated in Algorithm 1. The input features are first augmented by AugDiff and then provided to the MIL model to predict bag-level labels in each epoch of MIL training. We select three classic MIL methods, including the attention-based AMIL [23], the loss-guided attention-based LossAttn [32], and the dual-stream attention based DSMIL [19]. It should be noted that during the MIL training process, the trained DAE in AugDiff does not participate in the optimization process.

Algorithm 1 AugDiff for MIL Training

Input: The bags in the training dataset $\mathbf{Z}_O = \{Z_O^i\}_{i=1}^M$. Bag label $\{Y^i\}_{i=1}^M$. Proposed Diffusion-Denoising sampling process \mathcal{T} . Trained DAE ϵ_θ . Diffusion and Denoising step K in the sampling process.
Output: Trained MIL model ρ .
while not converged **do**
 for Z_O^i in \mathbf{Z}_O **do**
 %1. AugDiff performs feature augmentation on the input Z_O^i . Then, the MIL model takes the augmented feature \tilde{Z}_O^i as input to predict the score vector.
 $p \leftarrow \rho(\mathcal{T}(Z_O^i, \epsilon_\theta, K))$
 %2. Take the gradient descent step to optimize the MIL model.
 $\nabla \{-Y^i \log(p)\}$
 end
end

Training Process in AugDiff. To guide DAE to learn effective feature augmentation directions through the feature distribution of image augmentation, we first construct a training data set for the training process in AugDiff. According to the findings of [38], we choose six common and effective image augmentations: 1) Random rotation, 2) Random Elastic deformation, 3) Random Affine transformation, 4) Random Gaussian blurring, 5) Random Color Jitter, 6) Random Hematoxylin-Eosin-DAB (HED) Jitter.

In the training process of AugDiff, we employ the time-conditional UNet [29, 28] as DAE. Specifically, we randomly add a noise of level t to the feature. Then the noisy

feature is fed into DAE to predict the added noise. The objective of the optimization is to minimize the mean-squared error loss between the true noise and the predicted noise. The specific process is shown in Algorithm 2.

Algorithm 2 Training Process in AugDiff

Input: The set of original feature \mathcal{Z}_O and six sets of different augmented feature $\mathcal{Z}_R, \mathcal{Z}_E, \mathcal{Z}_A, \mathcal{Z}_B, \mathcal{Z}_C, \mathcal{Z}_H$.
Output: Trained DAE ϵ_θ .
while not converged **do**
 %1. Sample a data from the clean data distribution $q(\mathbf{z}_0)$ of $\mathcal{Z}_O \cup \mathcal{Z}_R \cup \mathcal{Z}_E \cup \mathcal{Z}_A \cup \mathcal{Z}_B \cup \mathcal{Z}_C \cup \mathcal{Z}_H$.
 $\mathbf{z}_0 \sim q(\mathbf{z}_0)$
 %2. Randomly sample noise from Gaussian distribution.
 $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
 %3. Randomly select a step and apply the diffusion process.
 $t \sim \text{Uniform}(\{1, \dots, T\})$
 $\mathbf{z}_t \sim \prod_{i=1}^t q(\mathbf{z}_i | \mathbf{z}_{i-1})$
 %4. Take the gradient descent step.
 $\nabla_\theta \|\epsilon - \epsilon_\theta(\mathbf{z}_t, t)\|_2^2$
end

Sampling Process in AugDiff. Intuitively, the feature augmentation process should make no significant changes to the semantic information of input instance features. That is, the features of the tumor patch cannot be changed into the features of the normal patch after augmentation. To accomplish this, we split the sampling process into two stages: 1) K -step Diffusion and 2) K -step Denoising.

To keep the original semantic information, the denoising process should start with a noise that contains the original semantic information instead of a randomly sampled noise. So, in the first stage, we apply the K -step Diffusion to the original features, where K is less than T . Then, in K -step Denoising, we employ the trained DAE to denoise the input features by K steps. Since DAE is trained with a large number of augmented features, it can output the augmented version of the original features. Besides, it is worth mentioning that this two-stage sampling not only preserves the original semantic information but also significantly accelerates the sampling process. The sampling of AugDiff is shown in Algorithm 3.

4. Experiments

4.1. Implementation Details.

(1) *Dataset Description.* We conducted experiments using three different cancer datasets, as shown in Table 1, including SICAPv2 [34] for prostate cancer, UnitoPatho [2] for colorectal cancer, and TMAs [20] for pancreatic cancer. For UnitoPatho we used the official splitting, while for SICAPv2 and TMAs we used 4-fold cross-validation.

Algorithm 3 Sampling Process in AugDiff

Input: The original bag $Z_O = \{\mathbf{z}_0^i\}_{i=1}^N$. Diffusion and Denoising step K in the sampling process.

Output: Augmented bag \tilde{Z} .

$$\tilde{Z} = \emptyset$$

```

for  $\mathbf{z}_0^i$  in  $Z_O$  do
    %1. With  $\mathbf{z}_0^i$  as the start,  $K$  diffusion steps are taken.
     $\tilde{\mathbf{z}}_K^i \sim \prod_{t=1}^K q(\mathbf{z}_t^i | \mathbf{z}_{t-1}^i)$ 
    %2. Denoising diffusion process.
    for  $t = K, \dots, 1$  do
        % Use the trained  $\epsilon_\theta$  and take the DDIM strategy.
         $\tilde{\mathbf{z}}_{t-1}^i \leftarrow \text{DDIM}(\tilde{\mathbf{z}}_t^i, t)$ 
    end
     $\tilde{Z} \leftarrow \tilde{Z} \cup \{\tilde{\mathbf{z}}_0^i\}$ 
end

```

Datasets	Size	Num	Total	Train	Val	Test
SICAPv2 [34]	512	121	155	95	27	33
UnitoPatho [2]	512	330	292	174	30	88
TMAs [20]	224	115	786	472	118	196

Table 1. Dataset Description. We report data partitioning as well as statistics for the three used datasets. **Size.** The size of the patch utilized in the dataset, with each patch performing a series of data augmentations before feature extraction. **Num.** The average number of patches contained in each WSI in the dataset. **SICAPv2.** SICAPv2 is a dataset containing prostate histological WSIs with Gleason-grade annotations. We assign the Gleason grade labels to the following five categories: ‘0+0’: 0, ‘3+3’: 1, ‘3+4’: 2, ‘4+3’: 2, ‘4+4’: 3, ‘3+5’: 3, ‘5+3’: 3, ‘4+5’: 4, ‘5+4’: 4, ‘5+5’: 4. **UnitoPatho.** UnitoPatho is a dataset for the classification of colorectal cancer polyps that contains four categories: Normal tissue, Hyperplastic Polyp, Tubular Adenoma, and Tubulo-Villous Adenoma. **TMAs.** TMAs is a dataset for classifying pancreatic tissue with three categories: pancreatic ductal adenocarcinoma, chronic pancreatitis, and normal pancreatic tissue.

(2) **MIL Methods.** We selected three classic MIL methods, including AMIL [23], LossAttn [32], and DSMIL [19]. The training is implemented using the open source code [31]. (3) **Feature Extraction.** We chose two different encoders for feature extraction, including ResNet18 [15] and RegNetX_004 [27]. The pre-trained weights for the feature extractor are derived from the timm library [40], and the features are 512- and 384-dimensional, respectively. (4) **Comparison Experiments.** We compared with the no augmentation method, pseudo bag construction method used in [46, 30, 3], feature augmentation framework with Mixup [45] which is used in [42, 13], feature augmentation framework with GAN which is used in [44], and image-level augmentation methods. (5) **Details of AugDiff.** Our implementation is based on the [28]. The batch size of the training is

1200, the base learning rate is 5.0e-08, the sampling strategy adopts the DDIM [35], the total steps T is set to 20 or 30, and the number of sampling steps K is set to $0.2T$ or $0.4T$.

4.2. Main Results

To demonstrate the general applicability of our framework, we conducted experiments using three prevalent MIL methods on three cancer datasets. The results are shown in Table 2. To ensure a fair comparison, we did not augment the test sets during the evaluation of all frameworks. To demonstrate the superiority of the proposed framework in terms of speed over Patch Augmentation, we performed speed tests and summarized the results in Figure 3. We can obtain the following observations.

1) Compared to MIL training without data augmentation (No Augmentation), current feature augmentation methods failed to consistently improve model performance to the same extent as Patch Augmentation (e.g., UnitoPatho (RegNetX), mean AUC, Feature Mixup -0.1%, Pseudo Bag -0.2%, Pix2Pix -1.1%). The linear mixing operation of Mixup can easily lead to unrealistic features. The pseudo-bag method makes it easy to introduce label noise while expanding the training data. In GAN feature generation training, weak discriminators tend to result in weak generators and consequently unrealistic features. In contrast, Patch Augmentation is an image-level augmentation method that produces more realistic images and extracts more realistic augmented features.

2) Though Patch Augmentation consistently facilitates the MIL training, AugDiff is more effective in terms of both performance and speed (e.g., SICAPv2 (ResNet18), mean AUC, $T=20 +2.3\%$. UnitoPatho (RegNetX), mean AUC, $T=20 +1.5\%$). For performance, the redundancy of pixel space limits Patch Augmentation, leading to limited changes in feature space after Patch Augmentation. In contrast, the AugDiff-learned feature augmentation is more informative and representative. For speed, AugDiff performs more than 30 times faster than the Patch Augmentation framework (as shown in Figure 3). The time-consuming feature extraction process required for the Patch Augmentation framework is unnecessary in our AugDiff framework because it performs augmentation at the feature level.

3) Benefiting from the step-by-step generation of the Diffusion Model, AugDiff can be further improved by increasing the number of sampling steps. Specifically, the parameter T determines the total number of model sampling steps and their fineness. A larger T can often help AugDiff generate better features (e.g., SICAPv2 (RegNetX), mean AUC, compared to $T=20$, $T=30 +0.8\%$). A smaller T can also help AugDiff achieve a balance between performance and time while achieving superior results than competing methods and ensuring a faster sampling rate.

Augmentation	SICAPv2 (ResNet18)								SICAPv2 (RegNetX)							
	AMIL [23]		LossAttn [32]		DSMIL [19]		Mean		AMIL [23]		LossAttn [32]		DSMIL [19]		Mean	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
No Augmentation	0.292 _{.11}	0.684 _{.12}	0.414 _{.09}	0.725 _{.06}	0.341 _{.08}	0.696 _{.02}	0.349	0.702	0.340 _{.06}	0.719 _{.03}	0.382 _{.06}	0.722 _{.04}	0.414 _{.08}	0.762 _{.03}	0.379	0.734
Patch Augmentation	<u>0.402</u> _{.08}	0.750 _{.04}	0.382 _{.05}	0.714 _{.07}	0.346 _{.03}	0.714 _{.03}	0.377	0.726	0.401 _{.06}	0.723 _{.04}	<u>0.406</u> _{.04}	0.716 _{.04}	0.400 _{.07}	0.761 _{.04}	<u>0.402</u>	0.733
Pseudo Bag	0.410 _{.08}	0.708 _{.08}	0.398 _{.07}	0.722 _{.06}	0.410 _{.06}	0.723 _{.05}	0.406	0.718	0.408 _{.04}	0.731 _{.04}	0.370 _{.09}	0.715 _{.03}	0.372 _{.01}	0.755 _{.04}	0.383	0.734
Feature Mixup	0.278 _{.10}	0.676 _{.11}	0.285 _{.12}	0.648 _{.14}	0.258 _{.08}	0.648 _{.07}	0.274	0.657	0.405 _{.03}	0.716 _{.05}	<u>0.392</u> _{.14}	0.673 _{.10}	0.418 _{.09}	0.764 _{.02}	0.405	0.718
Pix2Pix	0.336 _{.10}	0.720 _{.07}	0.349 _{.05}	0.714 _{.06}	0.332 _{.02}	0.725 _{.04}	0.339	0.720	0.404 _{.06}	0.732 _{.06}	0.323 _{.05}	0.666 _{.05}	0.344 _{.07}	0.705 _{.03}	0.357	0.701
AugDiff ($T=20$)	0.372 _{.01}	0.762 _{.05}	0.432 _{.13}	0.733 _{.07}	<u>0.382</u> _{.07}	0.752 _{.04}	0.395	0.749	0.385 _{.06}	0.745 _{.05}	0.345 _{.04}	<u>0.727</u> _{.03}	0.363 _{.06}	0.773 _{.03}	0.364	0.748
AugDiff ($T=30$)	0.393 _{.04}	0.765 _{.04}	<u>0.422</u> _{.12}	0.738 _{.08}	0.365 _{.06}	<u>0.749</u> _{.05}	0.393	0.751	<u>0.420</u> _{.05}	0.745 _{.04}	0.350 _{.03}	0.737 _{.02}	0.408 _{.055}	0.786 _{.01}	0.393	0.756
Augmentation	UnitoPatho (ResNet18)								UnitoPatho (RegNetX)							
	AMIL [23]		LossAttn [32]		DSMIL [19]		Mean		AMIL [23]		LossAttn [32]		DSMIL [19]		Mean	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
No Augmentation	0.659	0.895	0.682	0.904	0.670	0.828	0.670	0.876	0.670	0.858	0.704	0.891	0.704	0.868	0.693	0.872
Patch Augmentation	<u>0.727</u>	0.889	0.704	0.907	0.670	0.864	0.700	0.887	<u>0.704</u>	0.846	0.693	0.895	<u>0.739</u>	0.876	0.712	0.872
Pseudo Bag	0.716	0.891	0.716	0.906	0.670	0.827	0.701	0.875	0.704	0.851	0.704	0.891	0.727	0.867	0.712	0.870
Feature Mixup	0.682	0.896	0.704	0.900	0.693	0.832	0.716	0.888	0.670	0.850	0.716	0.886	0.693	0.877	0.693	0.871
Pix2Pix	0.636	0.854	0.648	0.880	0.682	0.830	0.665	0.863	0.693	0.836	0.625	0.888	0.693	0.860	0.670	0.861
AugDiff ($T=20$)	0.716	0.906	0.727	0.915	0.761	<u>0.913</u>	0.735	0.911	0.716	0.899	0.716	0.899	<u>0.739</u>	0.900	0.724	0.899
AugDiff ($T=30$)	0.761	0.906	0.716	<u>0.911</u>	0.761	<u>0.914</u>	0.746	0.910	0.704	0.892	0.716	0.904	<u>0.750</u>	0.905	0.723	0.900
Augmentation	TMAs (ResNet18)								TMAs (RegNetX)							
	AMIL [23]		LossAttn [32]		DSMIL [19]		Mean		AMIL [23]		LossAttn [32]		DSMIL [19]		Mean	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
No Augmentation	0.821 _{.02}	0.930 _{.02}	0.822 _{.03}	<u>0.938</u> _{.02}	0.805 _{.02}	0.921 _{.02}	0.816	0.930	0.833 _{.03}	0.932 _{.02}	<u>0.836</u> _{.02}	0.943 _{.02}	0.814 _{.03}	0.924 _{.02}	0.828	0.933
Patch Augmentation	0.814 _{.03}	0.933 _{.02}	<u>0.830</u> _{.02}	0.937 _{.01}	<u>0.824</u> _{.01}	0.933 _{.01}	<u>0.823</u>	0.934	<u>0.830</u> _{.03}	0.935 _{.02}	<u>0.842</u> _{.01}	0.941 _{.02}	0.841 _{.03}	0.933 _{.02}	0.838	0.936
Pseudo Bag	0.800 _{.03}	0.927 _{.03}	0.833 _{.03}	<u>0.938</u> _{.01}	0.804 _{.02}	0.926 _{.01}	0.812	0.930	0.828 _{.03}	0.926 _{.02}	0.835 _{.02}	0.934 _{.01}	0.819 _{.01}	0.924 _{.01}	0.827	0.928
Feature Mixup	0.812 _{.03}	0.934 _{.02}	0.818 _{.03}	<u>0.938</u> _{.02}	0.795 _{.02}	0.926 _{.02}	0.808	0.933	<u>0.830</u> _{.03}	0.933 _{.02}	0.828 _{.02}	0.942 _{.02}	0.814 _{.03}	0.916 _{.02}	0.824	0.930
Pix2Pix	0.786 _{.02}	0.914 _{.02}	0.809 _{.03}	0.930 _{.02}	0.742 _{.03}	0.892 _{.02}	0.779	0.912	0.780 _{.04}	0.910 _{.03}	0.793 _{.05}	0.916 _{.02}	0.665 _{.01}	0.852 _{.01}	0.746	0.893
AugDiff ($T=20$)	<u>0.824</u> _{.03}	<u>0.939</u> _{.02}	0.821 _{.03}	0.939 _{.01}	0.826 _{.01}	<u>0.935</u> _{.01}	0.824	<u>0.938</u>	0.821 _{.01}	0.939 _{.01}	0.821 _{.01}	0.938 _{.01}	<u>0.836</u> _{.01}	<u>0.935</u> _{.01}	0.826	0.937
AugDiff ($T=30$)	0.831 _{.05}	0.944 _{.02}	0.812 _{.03}	0.934 _{.01}	0.819 _{.02}	0.941 _{.01}	0.821	0.940	0.828 _{.01}	<u>0.937</u> _{.01}	0.836 _{.03}	<u>0.942</u> _{.02}	0.823 _{.04}	0.940 _{.02}	<u>0.829</u>	0.940

Table 2. **Results of SICAPv2, UnitoPatho and TMAs.** We compare five popular MIL training frameworks based on three widely used MIL models, two different feature extractors, and three different cancer datasets. Metrics include micro ACC and macro AUC, where **bold** represents the highest value, and underlined represents the second-highest value. All results in SICAPv2 and TMAs were obtained through 4-fold cross-validation and described in the form of Mean_{std}. Besides, all results reported in UnitoPatho are based on the official splitting and evaluated over the official test dataset.

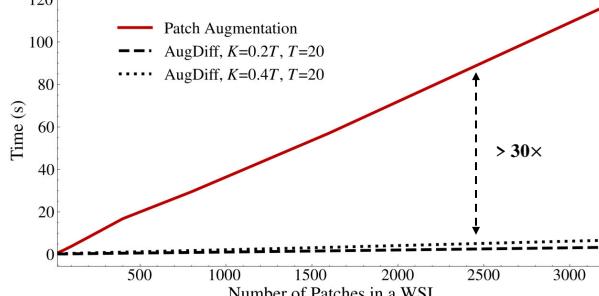


Figure 3. **Speed comparison of Patch Augmentation and AugDiff in MIL training.** We compare the speed of augmenting a WSI with various patch numbers under different augmentation frameworks. The vertical ordinate denotes the time needed for the Patch Augmentation framework or AugDiff, and the horizontal coordinate denotes the total number of patches in a WSI. See the appendix for specific experimental details.

4.3. Ablation Study

The MIL training is significantly influenced by K , as it determines how much the original semantic information is retained during the sampling process. To investigate the effect of parameter K under various scenarios, we tested

four values of K under two datasets, two feature extractors, and three MIL models. All results are presented in Table 3.

A larger K tends to fully exert the augmentation effect of AugDiff. Compared to $K = 0.1T$, the performance of AugDiff with a larger K , such as $0.2T/0.3T/0.4T$, could be significantly improved (e.g., SICAPv2 (ResNet18), compared to $0.1T, 0.2T+0.8\%, 0.3T+1.4\%, 0.4T+1.6\%$). Besides, the different choices of feature extractors and datasets could slightly affect the optimal choice of parameter K . For example, in the SICAPv2 dataset, when ResNet18 is used to extract features, the mean AUC will gradually increase with the increase of K . In contrast, when RegNetX is used to extract features, the optimal mean AUC is achieved near the K value of $0.2T$ or $0.3T$. In the UnitoPatho dataset, using different feature extractors, the performance of the MIL model typically improves as K increases, with $K = 0.4T$ often providing the best average performance.

In the appendix, we also discussed the impact of the conditional-guided mechanism on AugDiff. We compared the performance of AugDiff with unconditional-guided and conditional-guided and visualized the different types of augmentation under the conditional-guided mechanism. In summary, we found that conditional sampling has no significant effect on feature augmentation.

Augmentation	SICAPv2 (ResNet18)				UnitoPatho (ResNet18)			
	AMIL	LossAttn	DSMIL	Mean	AMIL	LossAttn	DSMIL	Mean
$K = 0.1T$	0.752.05	0.729.07	0.719.05	0.733	0.895	0.902	0.887	0.895
$K = 0.2T$	0.765.05	0.727.07	0.732.04	0.741	0.912	0.905	0.913	0.910
$K = 0.3T$	0.764.05	0.726.07	0.750.05	0.747	0.908	0.900	0.918	0.909
$K = 0.4T$	0.762.05	0.733.07	0.752.04	0.749	0.906	0.915	0.913	0.911

Augmentation	SICAPv2 (RegNetX)				UnitoPatho (RegNetX)			
	AMIL	LossAttn	DSMIL	Mean	AMIL	LossAttn	DSMIL	Mean
$K = 0.1T$	0.727.04	0.728.03	0.762.02	0.739	0.859	0.890	0.872	0.874
$K = 0.2T$	0.745.05	0.727.03	0.773.03	0.748	0.873	0.880	0.907	0.887
$K = 0.3T$	0.754.03	0.727.04	0.762.02	0.748	0.880	0.894	0.900	0.891
$K = 0.4T$	0.750.03	0.734.04	0.747.03	0.744	0.899	0.899	0.900	0.899

Table 3. **Effects of different settings in AugDiff.** We discuss the parameter K in the AugDiff sampling process over the SICAPv2 and UnitoPatho datasets. For SICAPv2, the reported AUC are described in the form of Mean_{std}. For UnitoPatho, the AUC are reported over the official test dataset.

4.4. Visualization

We utilized the UMAP[24] technique to perform dimensionality reduction on the embedding space of ResNet18 and then compared the augmented feature distribution of Patch Augmentation with that of AugDiff in the reduced embedding space. Details of our implementation and more visualization results are provided in the appendix. Figure 4 intuitively demonstrate the rationality of AugDiff and its superiority over Patch Augmentation. First, as shown in (a) and (b), as the augmentation rounds of AugDiff increase, the distribution of AugDiff’s augmented features (blue dots) approaches that of Patch Augmentation’s augmented features (red dots), indicating that AugDiff could effectively simulate image-level augmentation in the embedding space. Second, as shown in (b) and supplementary Figure 7, the blue dots form a more distinct structure than the red dots under the same augmentation rounds, indicating that the augmentation quality of AugDiff is better than Patch Augmentation. Furthermore, due to its better efficiency, AugDiff could generate more augmented samples than Patch Augmentation at the same time cost, thus improving the robustness and generalization of the MIL model. This is illustrated in (c), where the blue dots almost entirely cover the red dots.

Augmentation	SICAPv2 (ResNet18)				SICAPv2 (RegNetX)			
	AMIL	LossAttn	DSMIL	Mean	AMIL	LossAttn	DSMIL	Mean
PA (1×)	0.750.04	0.714.07	0.714.03	0.726	0.723.04	0.716.04	0.761.04	0.733
PA (5×)	0.758.04	0.732.07	0.720.02	0.737	0.728.02	0.729.04	0.758.04	0.738
PA (10×)	0.758.04	0.725.06	0.726.03	0.736	0.737.02	0.732.04	0.755.05	0.741
AugDiff	0.762.05	0.733.07	0.752.04	0.749	0.745.05	0.727.03	0.773.03	0.748

Table 4. **Performance comparison of multiple rounds of Patch Augmentation vs. AugDiff.** In the SICAP dataset, we conduct additional rounds of patch augmentation to test the performance. PA denotes Patch Augmentation. $n \times$ denotes the number of times each patch augmentation method is applied to each patch. The AUC is reported in the form of Mean_{std}.

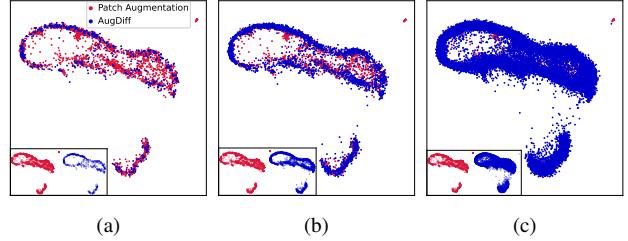


Figure 4. **Comparison between AugDiff and Patch Augmentation in the ResNet18 embedding space.** (a) AugDiff (1×) vs. Patch Augmentation (5×). (b) AugDiff (5×) vs. Patch Augmentation (5×). (c) AugDiff (50×) vs. Patch Augmentation (5×). The red and blue dots represent features of Patch Augmentation and AugDiff, respectively. $n \times$ denotes augmentation rounds.

5. Discussion

Superiority of AugDiff over More Patch Augmentation.

To demonstrate the significant advantage of AugDiff in generating high-quality augmented features, in addition to its speed, we conducted a performance comparison between multiple rounds of patch augmentation and AugDiff. We tested the performance of the MIL model under 1, 5, and 10 rounds of image augmentation on the SICAPv2 dataset. The more rounds of patch augmentation, the more features are utilized for MIL training with the Patch Augmentation framework. It should be noted that each round of patch augmentation includes six common types of patch augmentation. Furthermore, since the MIL training process adopts the early stopping mechanism, 10 rounds of patch augmentation can result in a similar amount of data expansion as AugDiff. The specific results are shown in Table 4.

Compared with one round of patch augmentation, as the augmentation rounds increase, the performance of the MIL model improves (e.g., SICAPv2 (RegNetX), compared to PA (1×), PA (5×) +0.5%, PA (10×) +0.8%). However, we also found that the improvement based on the number of image augmentation rounds is often limited (e.g., SICAPv2 (RegNet18), compared to PA (1×), PA (5×) +1.1%, PA (10×) +1.0%). In addition, by comparing 10 rounds of patch augmentation with AugDiff, we discovered that the feature augmentation strategy based on AugDiff can obtain more competitive results over various feature extractors and MIL models, demonstrating the high quality of AugDiff augmented features.

Superiority of AugDiff over Self-supervised Learning Framework.

AugDiff, as a training framework guided by image augmentation, is more promising than Self-Supervised Learning (SSL) methods guided by agent tasks. Specifically, SSL methods are typically based on preset agent tasks, require large-scale datasets, and have a more disordered training procedure. In contrast, AugDiff is

Augmentation	SICAPv2 (ResNet18)				UnitoPatho (ResNet18)			
	AMIL	LossAttn	DSMIL	Mean	AMIL	LossAttn	DSMIL	Mean
No Augmentation	0.684. ₁₂	0.725. ₀₆	0.696. ₀₂	0.702	0.895	0.904	0.828	0.876
SimSiam	0.606. ₁₀	0.622. ₀₇	0.615. ₀₈	0.614	0.856	0.883	0.902	0.880
AugDiff	0.762. ₀₅	0.733. ₀₇	0.752. ₀₄	0.749	0.906	0.915	0.913	0.911

Table 5. **Discussion on self-supervised learning and feature augmentation.** We choose SimSiam [7] in self-supervised training as a comparison. For SICAPv2, the reported **AUC** are described in the form of Mean_{std} . For UnitoPatho, the **AUC** are reported over the official test dataset.

Augmentation	SICAPv2 (ResNet18)				UnitoPatho (ResNet18)			
	AMIL	LossAttn	DSMIL	Mean	AMIL	LossAttn	DSMIL	Mean
Patch Augmentation	0.750. ₀₄	0.714. ₀₇	0.714. ₀₃	0.726	0.889	0.907	0.864	0.887
SICAPv2	0.762. ₀₅	0.733. ₀₇	0.752. ₀₄	0.749	0.907	0.908	0.915	0.910
UnitoPatho	0.759. ₀₅	0.731. ₀₈	0.726. ₀₆	0.739	0.906	0.915	0.913	0.911

Augmentation	SICAPv2 (RegNetX)				UnitoPatho (RegNetX)			
	AMIL	LossAttn	DSMIL	Mean	AMIL	LossAttn	DSMIL	Mean
Patch Augmentation	0.723. ₀₄	0.716. ₀₄	0.761. ₀₄	0.733	0.846	0.895	0.876	0.872
SICAPv2	0.745. ₀₅	0.727. ₀₃	0.773. ₀₃	0.748	0.906	0.890	0.906	0.901
UnitoPatho	0.746. ₀₅	0.726. ₀₄	0.772. ₀₄	0.748	0.899	0.899	0.900	0.899

Table 6. **Discussion on generalization ability of AugDiff.** We discuss about AugDiff’s ability to generalize to external datasets after being pretrained on specific datasets. In the *Augmentation* column, *UnitoPatho* and *SICAPv2* stand for pre-trained data sets used by AugDiff. In SICAPv2, the reported **AUC** are described in the form of Mean_{std} . In UnitoPatho, the **AUC** are reported in the official test dataset.

guided by the mapping of image augmentation in feature space, which is a more direct and effective guide. To compare the applicability of the two frameworks, we selected SimSiam [7], a commonly used SSL method, as a comparison. Specifically, in the downstream dataset, we first pre-trained ResNet18 based on SimSiam and then introduced the pre-trained feature extractor into the traditional MIL framework (No Augmentation). The corresponding results are shown in Table 5.

Since the SSL training process is more disordered, it has a more urgent need for the size of the dataset. In the relatively small dataset SICAPv2, SSL has a less pre-training effect than UnitoPatho. In the relatively large dataset UnitoPatho, the SSL barely exceeds the traditional MIL framework. More importantly, the combination of SSL and MIL is often used in the feature extractor. SSL needs to at least get better results than large-scale ImageNet pre-trained feature extractors. The combination of AugDiff and MIL is used to do the feature augmentation. It only needs to expand the diversity of the features extracted by the ImageNet pre-trained encoder to improve the MIL, which is a more straightforward and less data-demanding task. As a result, with stronger prior information for training and greater suitability for MIL, the AugDiff framework can have a better performance than SSL over datasets of different sizes.

Superiority of AugDiff in Generalization Ability. We conducted cross-tests over two different cancer datasets to verify the generalization ability of pre-trained AugDiff on external datasets. The experimental setup is as follows: 1) AugDiff pre-trained on UnitoPatho is used for feature augmentation over the SICAPv2 dataset; 2) AugDiff pre-trained on SICAPv2 is used for feature augmentation over the UnitoPatho dataset. Due to the fact that AugDiff’s training and testing rely on fixed feature dimensions, there are no cross-tests between feature extractors of different dimensions (e.g., ResNet18 and RegNetX). Table 6 summarizes all experimental results. We found that the pre-trained AugDiff has good feature augmentation capabilities when tested on external datasets, and all the external test results are higher than Patch Augmentation. Besides, since the SICAPv2 dataset may have a more diverse patch morphology, AugDiff was better pre-trained on the SICAPv2 dataset. Note that the morphology often varies between different cancer datasets [10]. Since AugDiff’s training is guided by image augmentation, it avoids overfitting to the specific dataset. Thus, the pre-trained AugDiff can be deployed to a wider variety of datasets and applications.

6. Conclusion

In this work, we present a novel feature augmentation framework based on Diffusion Model (DM), AugDiff, for the MIL training process in the WSI-related tasks. Restricted by limited WSI-level training data, image augmentation is often used to assist MIL training. However, tens of thousands of patches in WSIs limit image augmentation, and it can only assist MIL in an offline manner, which is an inefficient and sub-optimal solution. Feature augmentation is a promising solution with online augmentation, but unrealistic and unstable generations always limit performance. To solve the above issues, we introduce the diversity-generating ability of DM into the MIL training, where the input features can be augmented in each training epoch. Besides, the step-by-step generation characteristic of DM can control the retention of semantic information during the augmentation sampling process. We perform sufficient tests over three cancer datasets, two different feature extractors, and three widely used MIL models. The comparison results show the proposed AugDiff can significantly improve the current framework, and the ablation study shows the rationality of the sampling process. Furthermore, we highlight the high quality of augmented features and the superiority of AugDiff over the SSL method. Discussions on generalization and analysis of visualizations further demonstrate that AugDiff has excellent prospects. In the future, we will evaluate the performance of AugDiff over larger datasets and employ faster sampling techniques to improve the speed further.

References

- [1] Behnaz Abdollahi, Naofumi Tomita, and Saeed Hassanzadeh. Data augmentation in training deep learning models for medical image analysis. *Deep learners and deep learner descriptors for medical applications*, pages 167–180, 2020. [2](#)
- [2] Carlo Alberto Barbano, Daniele Perlo, Enzo Tartaglione, Attilio Fiandrotti, Luca Bertero, Paola Cassoni, and Marco Grangetto. Unitopatho, a labeled histopathological dataset for colorectal polyps classification and adenoma dysplasia grading. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 76–80. IEEE, 2021. [4, 5](#)
- [3] Hao Bian, Zhuchen Shao, Yang Chen, Yifeng Wang, Haqian Wang, Jian Zhang, and Yongbing Zhang. Multiple instance learning with mixed supervision in gleason grading. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 204–213. Springer, 2022. [5](#)
- [4] 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*, pages 1301–1309, 2019. [1](#)
- [5] Richard J Chen, Ming Y Lu, Tiffany Y Chen, Drew FK Williamson, and Faisal Mahmood. Synthetic data in machine learning for medicine and healthcare. *Nature Biomedical Engineering*, 5(6):493–497, 2021. [2](#)
- [6] Richard J Chen, Ming Y Lu, Muhammad Shaban, Chengkuan Chen, Tiffany Y Chen, Drew FK Williamson, and Faisal Mahmood. Whole slide images are 2d point clouds: Context-aware survival prediction using patch-based graph convolutional networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 339–349. Springer, 2021. [1](#)
- [7] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15750–15758, June 2021. [8](#)
- [8] Yizhou Chen, Xu-Hua Yang, Zihan Wei, Ali Asghar Heidari, Nenggan Zheng, Zhicheng Li, Huiling Chen, Haigen Hu, Qianwei Zhou, and Qiu Guan. Generative adversarial networks in medical image augmentation: a review. *Computers in Biology and Medicine*, page 105382, 2022. [2](#)
- [9] Phillip Chlap, Hang Min, Nym Vandenberg, Jason Dowling, Lois Holloway, and Annette Haworth. A review of medical image data augmentation techniques for deep learning applications. *Journal of Medical Imaging and Radiation Oncology*, 65(5):545–563, 2021. [2](#)
- [10] Ozan Ciga, Tony Xu, and Anne Louise Martel. Self supervised contrastive learning for digital histopathology. *Machine Learning with Applications*, 7:100198, 2022. [8](#)
- [11] Florinel-Alin Croitoru, Vlad Hondu, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *arXiv preprint arXiv:2209.04747*, 2022. [2](#)
- [12] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. [2, 4](#)
- [13] Michael Gadermayr, Lukas Koller, Maximilian Tschuchnig, Lea Maria Stangassinger, Christina Kreutzer, Sebastien Couillard-Despres, Gertie Janneke Oostingh, and Anton Hittmair. Mixup-mil: Novel data augmentation for multiple instance learning and a study on thyroid cancer diagnosis. *arXiv preprint arXiv:2211.05862*, 2022. [2, 5](#)
- [14] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. [2](#)
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. [5](#)
- [16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. [2](#)
- [17] Andreas Kleppe, Ole-Johan Skrede, Sepp De Raedt, Knut Liestøl, David J Kerr, and Håvard E Danielsen. Designing deep learning studies in cancer diagnostics. *Nature Reviews Cancer*, 21(3):199–211, 2021. [2](#)
- [18] Daisuke Komura and Shumpei Ishikawa. Machine learning methods for histopathological image analysis. *Computational and structural biotechnology journal*, 16:34–42, 2018. [1](#)
- [19] 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 Conference on Computer Vision and Pattern Recognition*, 2021. [4, 5, 6](#)
- [20] Bin Li, Michael S Nelson, Omid Savari, Agnes G Loeffler, and Kevin W Eliceiri. Differentiation of pancreatic ductal adenocarcinoma and chronic pancreatitis using graph neural networks on histopathology and collagen fiber features. *Journal of Pathology Informatics*, 13:100158, 2022. [4, 5](#)
- [21] Jun Hao Liew, Hanshu Yan, Daquan Zhou, and Jiashi Feng. Magicmix: Semantic mixing with diffusion models. *arXiv preprint arXiv:2210.16056*, 2022. [4](#)
- [22] Ming Y Lu, Tiffany Y Chen, Drew FK Williamson, Melissa Zhao, Maha Shady, Jana Lipkova, and Faisal Mahmood. Ai-based pathology predicts origins for cancers of unknown primary. *Nature*, 594(7861):106–110, 2021. [1](#)
- [23] 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, 4, 5, 6](#)
- [24] L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints*, Feb. 2018. [7](#)
- [25] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018. [11](#)
- [26] Muzaffer Özbeş, Salman UH Dar, Hasan A Bedel, Onat Dalmaç, Şaban Öztürk, Alper Güngör, and Tolga Çukur. Unsupervised medical image translation with adversarial diffusion models. *arXiv preprint arXiv:2207.08208*, 2022. [2](#)

- [27] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces, 2020. 5
- [28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022. 2, 4, 5
- [29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 4
- [30] Wei Shao, Tongxin Wang, Zhi Huang, Zhi Han, Jie Zhang, and Kun Huang. Weakly supervised deep ordinal cox model for survival prediction from whole-slide pathological images. *IEEE Transactions on Medical Imaging*, 40(12):3739–3747, 2021. 1, 2, 5
- [31] 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, 5
- [32] Xiaoshuang Shi, Fuyong Xing, Yuanpu Xie, Zizhao Zhang, Lei Cui, and Lin Yang. Loss-based attention for deep multiple instance learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5742–5749, 2020. 4, 5, 6
- [33] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019. 2
- [34] Julio Silva-Rodríguez, Adrián Colomer, María A Sales, Rafael Molina, and Valery Naranjo. Going deeper through the gleason scoring scale: An automatic end-to-end system for histology prostate grading and cribriform pattern detection. *Computer methods and programs in biomedicine*, 195:105637, 2020. 4, 5
- [35] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 5
- [36] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 2
- [37] Chetan L Srinidhi, Ozan Ciga, and Anne L Martel. Deep neural network models for computational histopathology: A survey. *Medical Image Analysis*, 67:101813, 2021. 1
- [38] David Tellez, Geert Litjens, Péter Bárdi, Wouter Bulten, John-Melle Bokhorst, Francesco Ciompi, and Jeroen Van Der Laak. Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. *Medical image analysis*, 58:101544, 2019. 2, 4
- [39] Jeroen Van der Laak, Geert Litjens, and Francesco Ciompi. Deep learning in histopathology: the path to the clinic. *Nature medicine*, 27(5):775–784, 2021. 2
- [40] Ross Wightman. Pytorch image models. <https://github.com/rwightman/pytorch-image-models>, 2019. 5
- [41] Yuan Xue, Jiarong Ye, Qianying Zhou, L Rodney Long, Sameer Antani, Zhiyun Xue, Carl Cornwell, Richard Zaino, Keith C Cheng, and Xiaolei Huang. Selective synthetic augmentation with histogan for improved histopathology image classification. *Medical image analysis*, 67:101816, 2021. 2
- [42] Jiawei Yang, Hanbo Chen, Yu Zhao, Fan Yang, Yao Zhang, Lei He, and Jianhua Yao. Remix: A general and efficient framework for multiple instance learning based whole slide image classification. In *Medical Image Computing and Computer Assisted Intervention—MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part II*, pages 35–45. Springer, 2022. 2, 5
- [43] 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
- [44] Imaad Zaffar, Guillaume Jaume, Nasir Rajpoot, and Faisal Mahmood. Embedding space augmentation for weakly supervised learning in whole-slide images. *arXiv preprint arXiv:2210.17013*, 2022. 2, 5
- [45] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 5
- [46] Hongrun Zhang, Yanda Meng, Yitian Zhao, Yihong Qiao, Xiaoyun Yang, Sarah E Coupland, and Yalin Zheng. Dfd-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, 5

7. Appendix

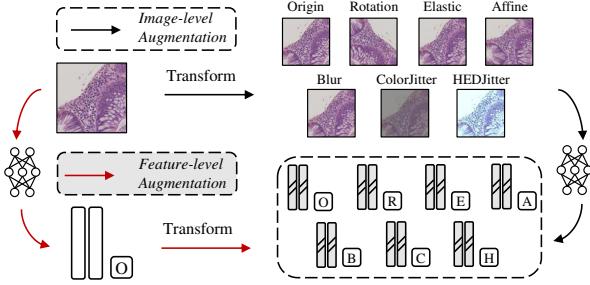


Figure 5. Construct the feature training Dataset for Denoising Autoencoder with Patch Augmentation and Feature Extraction. We show how to construct a training dataset for feature augmentation and give an example of different image augmentations.

7.1. Supplementary Implementation Details

Image Augmentation. The following parameters are chosen for image augmentation: 1) Random rotation, which includes rotation by 90 degrees and vertical and horizontal mirroring. 2) Random Elastic deformation with alpha equal to 2 and sigma equal to 0.06. 3) Random Affine transformation with an alpha value of 0.1. 4) Random Gaussian blurring with a radius of 0.5 to 1.5. 5) Random Color Jitter includes brightness and contrast image perturbation with a brightness intensity ratio between 0.65 and 1.35 and a contrast intensity ratio between 0.5 and 1.5. 6) Random Hematoxylin-Eosin-DAB (HED) Jitter with intensity ratios is 0.05. In Figure 5, we show an example of data augmentation.

Speed Comparison. We compare the speed between the Patch Augmentation framework and AugDiff. Three steps are involved in the Patch Augmentation framework: 1) reading every patch in WSI, 2) making the patch augmentation, and 3) extracting every patch’s features. The following steps are part of AugDiff: 1) reading the features of every patch in WSI and 2) performing a feature data augmentation. We test the speed of AugDiff with different settings.

7.2. Effects of Condition-guided Mechanism

The ablation study discuss the effect of the conditional-guided mechanism in AugDiff. The conditional guidance mechanism refers to direct guidance of feature augmentation during the training and sampling, which is widely used in image generation. In the test, we evaluate three MIL methods of two types of feature extractors on two datasets. All results are presented in Table 7.

Although the conditional guidance mechanism is widely used in image generation, it does not perform significantly better than unconditional guidance in the task of feature

augmentation. Table 7 shows that conditional and unconditional augmentation perform similarly on different datasets and feature extractors. There are the following reasons: 1) The core of feature augmentation to facilitate MIL model training is generating additional features to enlarge the training data set. The impact of increasing the number of features is greater than that of conditional and unconditional augmentation. In addition, the diverse generation capabilities of the Diffusion model further weaken the difference between conditional and unconditional augmentation. 2) Compared with the difference between different augmentation types in image augmentation, the difference between distinct feature augmentation may be more challenging to learn. The existing simple conditional mechanism may still need to improve. Besides, the visual analysis in the appendix further validates our results.

Augmentation	SICAPv2 (ResNet18)				UnitoPatho (ResNet18)			
	AMIL	LossAttn	DSMIL	Mean	AMIL	LossAttn	DSMIL	Mean
Uncondition	0.762. ⁰⁴	0.728. ⁰⁷	0.745. ⁰⁴	0.745	0.908	0.916	0.913	0.912
Condition	0.762. ⁰⁵	0.733. ⁰⁷	0.752. ⁰⁴	0.749	0.906	0.915	0.913	0.911
Augmentation	SICAPv2 (RegNetX)				UnitoPatho (RegNetX)			
	AMIL	LossAttn	DSMIL	Mean	AMIL	LossAttn	DSMIL	Mean
Uncondition	0.741. ⁰⁵	0.722. ⁰⁴	0.775. ⁰⁴	0.746	0.904	0.898	0.903	0.902
Condition	0.745. ⁰⁵	0.727. ⁰³	0.773. ⁰³	0.748	0.899	0.899	0.900	0.899

Table 7. Effects of different settings in AugDiff. We discuss the effects of the condition-guided mechanism in the AugDiff sampling process over the SICAPv2 and UnitoPatho datasets. In SICAPv2, the reported AUC are described in the form of Mean_{std}. In UnitoPatho, the AUC are reported in the official test dataset.

7.3. UMAP-based Visualization

Implementation of UMAP. We utilize the open-source implementation [25] of UMAP with the following hyperparameters: neighbors=50, dist=0.1, and random_state=42. We select a WSI (17B0024162) from the SICAPv2 dataset containing 88 patches. We apply each of the six Patch Augmentation to all patches for 5 rounds, resulting in a total of $5 \times 6 \times 88 = 2,640$ image-level augmented patches (corresponding to the red dots in Figure 4a, 4b and 4c). Additionally, we apply AugDiff with each of the six conditions (excluding condition=0) to all patches for 100 times, resulting in a total of $100 \times 6 \times 88 = 52,800$ feature-level augmented patches. It should be noted that we used $T = 30$ and $K = 0.4T$ in the sampling process in AugDiff. Then, we take 1, 5, and 50 patches from feature-level augmented patches under each condition for visualization, respectively (corresponding to the blue dots in Figure 4a, 4b and 4c, respectively).

More Visualization Results. To better understand AugDiff, we additionally use UMAP for two visualizations: 1) the augmented feature distributions of AugDiff

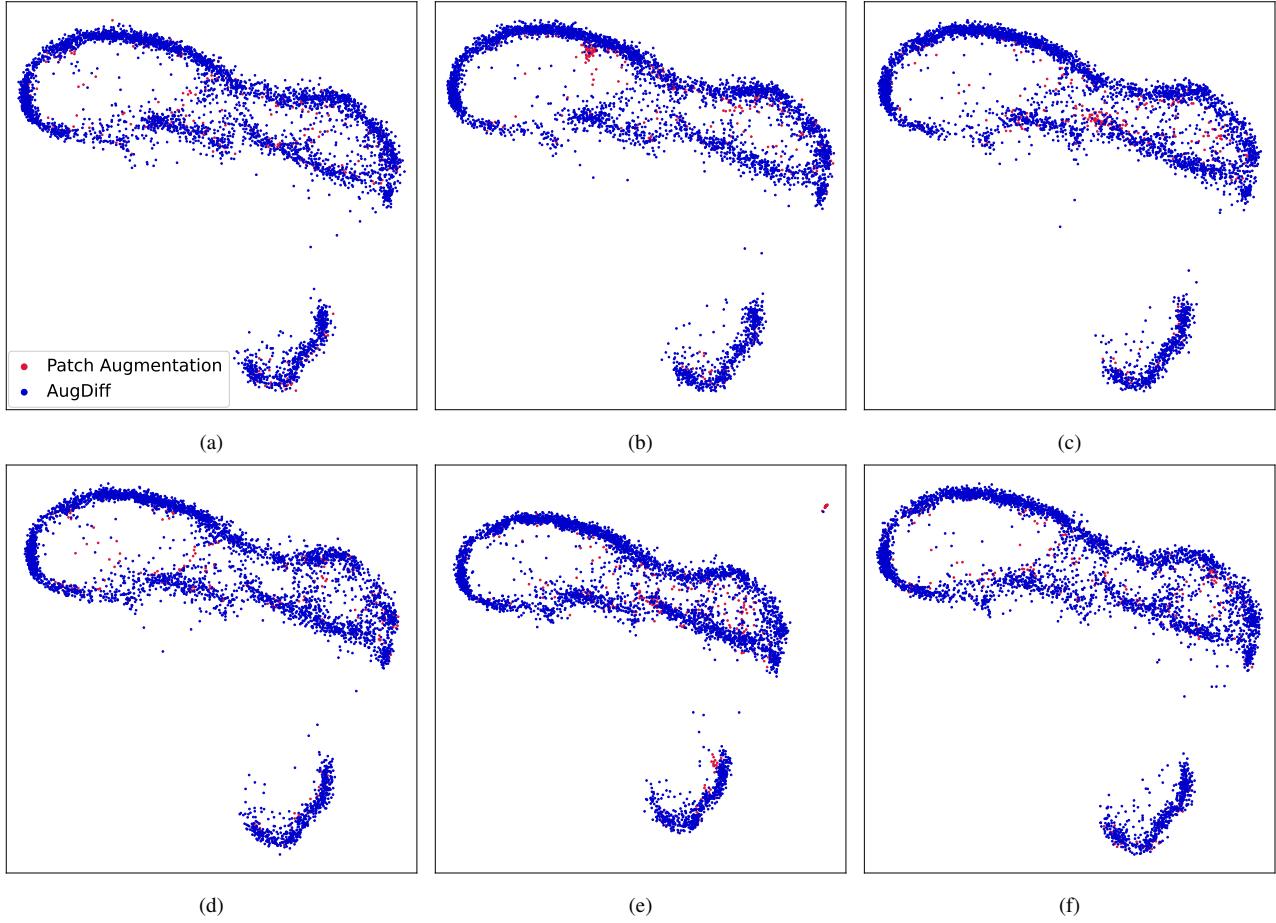


Figure 6. Comparison between AugDiff and Patch Augmentation in the low-dimensional ResNet18 embedding space when adopting distinct augmentation types. (a) AugDiff of condition=1 vs. Patch Augmentation of Rotation. (b) AugDiff of condition=2 vs. Patch Augmentation of Elastic. (c) AugDiff of condition=3 vs. Patch Augmentation of Affine. (d) AugDiff of condition=4 vs. Patch Augmentation of Blur. (e) AugDiff of condition=5 vs. Patch Augmentation of ColorJitter. (f) AugDiff of condition=6 vs. Patch Augmentation of HEDJitter. The red dots represent augmented samples of Patch Augmentation ($5\times$), and the blue dots represent augmented samples of AugDiff ($50\times$). $n\times$ denotes augmentation rounds.

under different augmentation types, *i.e.*, under conditions from 1 to 6 (Figure 6), and 2) the augmented feature distributions of AugDiff and Patch Augmentation under different augmentation rounds. For 1), we apply AugDiff under each of six conditions for 50 rounds, then separately visualize the 4,400 generated features and the 440 image-level augmented features of the corresponding augmentation type. It can be observed from Figure 6 that different types of image-level augmentation lead to remarkably similar distributions (see the red dots in a-f). Meanwhile, AugDiff can effectively simulate each image-level augmentation under the corresponding condition, which makes the augmented features distribution of Augdiff under various conditions close to each other (see the blue dots in a-f). That explains why the condition-guides mechanism has little impact on the model performance, as shown in Table

7. For 2), we applied AugDiff and Patch Augmentation 1, 5, 10, 25, and 50 rounds, respectively. We separately visualized the augmented AugDiff and Patch Augmentation samples under each augmentation time. As shown in Figure 7, the augmented features’ density distribution trends of AugDiff and Patch Augmentation are slightly different. Specifically, as the augmentation rounds increase, AugDiff puts higher density onto the border of the manifold, while Patch Augmentation puts a higher density inside the manifold. According to the results in Table 4, we hypothesize that the augmented samples distributed on the manifold border are of better quality because the performance gain of Patch Augmentation decreases with the increment of augmentation rounds. Therefore, AugDiff can bring more significant performance gain with less time cost than Patch Augmentation as the augmentation rounds increase.

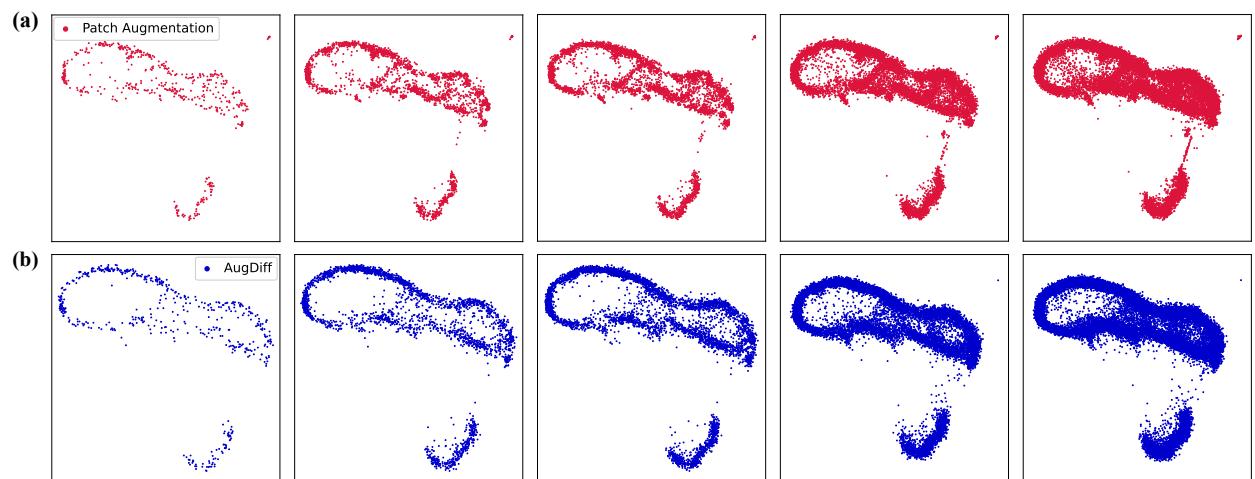


Figure 7. The distribution trends of Patch Augmentation and AugDiff. (a) We apply Patch Augmentation for 1, 5, 10, 25, 50 rounds and separately visualized the feature distribution of generated instances, respectively. (b) We apply AugDiff for 1, 5, 10, 25, 50 rounds and separately visualized the feature distribution of generated instances, respectively.