[Supplementary] Contrastive Cross-Bag Augmentation for Multiple Instance Learning-based Whole Slide Image Classification

Bo Zhang, Xinan Xu, Shuo Yan, Yu Bai, Zheng Zhang, Wufan Wang, Hui Gao, Wendong Wang

1. Implementation Details

Preprocessing. We employ two feature extractors to extract image features: one is a pretrained ResNet50 [2] on ImageNet, and another is ProV-Gigapath [4]. These models embed instance features into 1024-dimensional and 1536-dimensional vectors, respectively. We crop patches from WSIs using a fixed size of 256×256 without overlap. We incorporate Trident [5] as the preprocessing pipeline.

Evaluation Metrics We use Area Under Curve (AUC), image-level Accuracy (ACC), F1 score (F1) to evaluate the performance of our method. For the CAMELYON-16 dataset, we randomly partition 20% of the training samples into a validation set, with experiments repeated five times to ensure statistical reliability. For TCGA-LUNG and TCGA-BRCA, we adopt a five-fold cross-validation protocol to comprehensively assess model performance. The mean and standard deviation of evaluation metrics are reported across all repetitions and folds, respectively.

Model. To reduce the number of parameters, we perform a channel reduction operation before we perform Cross-Bag Augmentation. We adopt a linear layer to reduce the input instance dimension to 256, and after augmentation, we adopt another linear layer to restore the input dimension. We set the number of random views to 8 during training, the size of memory bank k in Bag-level Contrastive Learning to 256, and the number of prototypes C in the Cross-Group Contrastive Learning to 8.

Training. We train our model using the Adam optimizer with a learning rate of 1×10^{-4}. The batch size is set to 1, and training is conducted for 200 epochs. All experiments are performed on a single NVIDIA RTX 4090 GPU. We do not explicitly tune hyperparameters for the various MIL models. Instead, we use a consistent set of hyperparameters across all experiments. This uniform configuration demonstrates the robustness of our approach.

Datasets and Evaluation Metrics

1. **CAMELYON16** contains 270 training slides (159 normal, 111 tumor) and 129 testing slides. In Camelyon16, the tumor area only accounts for approximately less than 10% of the tissue area in the positive slide.
2. **TCGA-LUNG** is a publicly available data set comprising two main types of lung cancer: Lung Adenocarcinoma (TCGA-LUAD) and Lung Squamous Cell Carcinoma (TCGA-LUSC), collected as part of the Cancer Genome Atlas (TCGA) project. The data set includes a total of 1046 diagnostic WSIs, consisting of 534 TCGA-LUAD and 512 TCGA-LUSC cases. These slides were divided into a training set containing 836 WSIs and a testing set comprising 210 WSIs.
3. **TCGA-BRCA** The TCGA-BRAC dataset includes two subtypes of breast cancer, 831 Invasive Ductal (IDC) and 210 Invasive Lobular Carcinoma (ILC), 1041 in total.

2. Additional Quantitative Experiments

We conduct additional experiments to validate the effectiveness of C^2Aug.

2.1. More on Cross-Bag Augmentation

Number of Random Views in Multi-View Fusion We conduct a comparison of different numbers of views for Multi-View Fusion. We evaluate the effectiveness of varying the number of random views. The results are presented in Tab. 1. Setting random view to 8 achieves the highest performance across all metrics for both ResNet50 and Prov-Gigapath models. It achieves an ACC of 89.1, AUC of 93.2, and F1 score of 87.0 for the ResNet50 feature, and an ACC of 96.9, AUC of 98.0, and F1 score of 95.7 for the Prov-Gigapath feature. Rand denotes the Row-Wise Masking described in Section 3.2. We evaluated different numbers of views (4, 8, 16) to determine the optimal value. We conducted the experiments on the CAMELYON-16 dataset using ResNet50 and Prov-Gigapath features. The best performance was achieved when the number of views was set to 8 with the row-wise masking strategy.

View	ResNet50			Prov-Gigapath		
	ACC	AUC	F1	ACC	AUC	F1
Rand = 16	$88.7_{1.4}$	$92.3_{1.6}$	$86.6_{1.2}$	$94.1_{0.3}$	$97.1_{0.3}$	$94.4_{0.2}$
Fixed = 16	$87.6_{1.0}$	$92.1_{1.3}$	$86.4_{1.1}$	$93.7_{0.2}$	$96.8_{0.4}$	$93.6_{0.3}$
Rand = 8	$\mathbf{89.1_{1.2}}$	$\mathbf{93.2_{1.9}}$	$\mathbf{87.0_{1.0}}$	$\mathbf{96.9_{0.2}}$	$\mathbf{98.0_{0.3}}$	$\mathbf{95.7_{0.4}}$
Fixed = 8	$88.1_{1.2}$	$92.5_{1.2}$	$86.4_{1.2}$	$95.3_{0.4}$	$97.5_{0.4}$	$94.9_{0.3}$
Rand = 4	$88.2_{0.9}$	$92.3_{1.1}$	$86.2_{1.1}$	$94.9_{0.3}$	$97.5_{0.3}$	$94.7_{0.4}$
Fixed = 4	$87.8_{1.4}$	$91.9_{1.7}$	$86.0_{1.2}$	$94.9_{0.3}$	$97.3_{0.4}$	$94.7_{0.4}$

Table 1. Comparison of different numbers of views in Multi-View Fusion on the CAMELYON-16 dataset using DSMIL.

Size of Memory Bank of L_{bag}. We evaluate the optimal k value for L_{bag} by setting the memory bank size to 64, 128, 256, and 512, respectively, on the CAMELYON-16 dataset. ResNet50 is used to extract patch features for the experiments. As shown in Tab. 2, the optimal memory bank size is 256. The best performance is achieved when $k = 256$, with an accuracy of 91.9, an AUC of 94.7, and an F1 score of 91.1.

Number of Prototypes for L_{group}. We further evaluate the hyperparameter C (number of prototypes for L_{group}) to determine its optimal value on the CAMELYON-16 dataset. As illustrated in Tab. 3, when C is set to 8, the model achieves the best AUC performance of 94.7. Thus, the optimal value of C is determined to be 8. Accordingly, $C = 8$ is adopted in all subsequent experiments.

Training Cost and Memory Overhead on Larger Datasets. We validate the training efficiency and resource consumption of C^{2}Aug. As reported in Table 7, although C^{2}Aug incurs a modest increase in GPU memory usage and training time compared with baseline schemes without data augmentation, the corresponding overhead remains manageable even when extended to large-scale datasets. Notably, the training time shows linear scaling when we subsample the CAMELYON16 dataset to different fixed numbers of WSIs.

Effects of Reducing the Number of Bags with Small Instance Ratios. We validate that C^{2}Aug can reduce the number of bags with a low tumor instance ratio. The CAMELYON16 dataset is adopted for experiments due to its availability of patch-level labels. We statistically count the number of bags in each tumor instance/total instance ratio interval (e.g., $\leq 10\%$, 10%–20%) and independently perform MVF, IC, and IE operations. As illustrated in Tab. 8, MVF effectively reduces the number of bags with a small tumor instance ratio ($\leq 10\%$) from 89 to 52. IC and IE primarily address the long-tail issue [3] by generating input sequences of varying lengths, thus IC only reduces 6 bags and IE has no reduction effect for small tumor ratio. The MVF operation can effectively increase the tumor instance ratio within a bag, which is beneficial for training bags with small tumor instance ratios.

Comparison with Other Contrastive Learning-Based Methods. To verify the effectiveness of C^{2}Aug against state-of-the-art contrastive learning methods in the WSI domain, we compare it with three representative approaches: RetCCL, SCL-WC, and MuRCL. As shown in Tab. 9, our method achieves the best performance. This result demonstrates that the bag-level and group-level contrastive learning modules in C^{2}Aug can effectively boost model performance.

Incomplete Robustness Analysis. We further evaluate the robustness of C^{2}Aug against noisy labels. We randomly flip 10% and 20% of the training slide labels in the CAMELYON16 dataset for robustness testing. The labels of the test set remain unchanged. We compare C^{2}Aug with the TransMIL baseline method. As shown in Tab. 10, the performance of both C^{2}Aug and TransMIL decreases under label noise. However, the performance degradation of C^{2}Aug is smaller than that of TransMIL. The experimental results demonstrate that C^{2}Aug exhibits stronger robustness to noisy labels compared with TransMIL.

k	ACC	AUC	F1
k = 64	$89.2_{1.4}$	$92.0_{1.9}$	$88.8_{1.3}$
k = 128	$91.0_{0.9}$	$93.4_{1.2}$	$90.7_{1.2}$
k = 256	$\mathbf{91.9_{1.0}}$	$\mathbf{94.7_{1.9}}$	$\mathbf{91.1_{0.8}}$
k = 512	$90.1_{1.1}$	$93.3_{1.4}$	$89.7_{1.0}$

Table 2. Comparison of different memory bank sizes for L_{bag} on the CAMELYON-16 dataset using ResNet50 feature.

C	ACC	AUC	F1
C = 4	$89.6_{1.4}$	$92.4_{1.9}$	$88.9_{1.3}$
C = 8	$\mathbf{91.9_{1.0}}$	$\mathbf{94.7_{1.9}}$	$\mathbf{91.1_{0.8}}$
C = 16	$91.5_{1.1}$	$94.3_{1.5}$	$90.7_{1.0}$
C = 32	$91.3_{1.6}$	$94.1_{1.4}$	$90.5_{1.1}$

Table 3. Comparison of different numbers of prototypes used for L_{group}

Few-shot Validation. We evaluate the performance of C^{2}Aug under extreme low-data regimes via few-shot experiments. Experiments are conducted with 2, 4, and 8 training shots per category. As reported in Tab. 11, few-shot training yields inferior performance compared to training on the full dataset. The results indicate that our method still requires a sufficient number of training samples for optimal performance.

Comparison with Other Augmentation Methods To comprehensively validate the effectiveness of the proposed C^{2}Aug, we compare it with PMIL, MergeUp and PseMix. As shown in Tab. 12, our method achieves the optimal performance among all competing approaches.

Choice of Instance Fusion Method Instance Compression and Multi-View Fusion leverage cross-attention to aggregate multiple instances into a single representation. We compare the proposed compression-based method with existing selection-based approaches to validate its effectiveness. We compare our method with the selection-based approach RankMix [1]. To ensure a fair comparison, we adopt the same strategy as RankMix by using the attention scores from TransMIL during training to guide instance selection. For Multi-View Fusion, we employ the top-K strategy to select instances with the highest attention scores, aligning with the instance selection mechanism used in RankMix. The results are illustrated in Tab. 4, Tab. 5 and Tab. 6. In all three datasets, compression-based method consistently outperformed Top-K selection, with the most significant improvement observed on the TCGA-LUNG dataset, where the AUC increased by 3.9%. This is because top-k selec-

Model	ResNet50			Prov-Gigapath		
	ACC	AUC	F1	ACC	AUC	F1
Cross-Attn	$\mathbf{89.1_{1.2}}$	$\mathbf{93.2_{1.9}}$	$\mathbf{87.0_{1.0}}$	$\mathbf{96.9_{0.2}}$	$\mathbf{98.0_{0.3}}$	$\mathbf{95.7_{0.4}}$
Top-K	$86.0_{1.2}$	$89.3_{1.2}$	$84.1_{0.8}$	$94.4_{0.3}$	$96.4_{0.5}$	$94.2_{0.3}$

Table 4. Comparison of different instance fusion methods on the CAMELYON-16 dataset using DSMIL.

Model	ResNet50			Prov-Gigapath		
	ACC	AUC	F1	ACC	AUC	F1
Cross-Attn	$\mathbf{91.2_{2.0}}$	$\mathbf{93.5_{1.7}}$	$\mathbf{89.8_{1.3}}$	$\mathbf{94.7_{0.1}}$	$\mathbf{97.5_{0.2}}$	$\mathbf{93.6_{0.3}}$
Top-K	$87.3_{1.1}$	$91.0_{1.4}$	$85.6_{1.1}$	$92.1_{0.2}$	$95.2_{0.3}$	$91.7_{0.2}$

Table 5. Comparison of different instance fusion methods on the TCGA-LUNG dataset using DSMIL.

Model	ResNet50			Prov-Gigapath		
	ACC	AUC	F1	ACC	AUC	F1
Cross-Attn	$\mathbf{89.4_{0.6}}$	$\mathbf{93.7_{1.9}}$	$\mathbf{88.4_{1.1}}$	$\mathbf{94.6_{0.2}}$	$\mathbf{96.8_{0.3}}$	$\mathbf{93.9_{0.2}}$
Top-K	$86.2_{1.3}$	$90.3_{1.2}$	$84.3_{0.9}$	$91.2_{0.4}$	$92.4_{0.7}$	$90.9_{0.3}$

Table 6. Comparison of different instance fusion methods on the TCGA-BRCA dataset using DSMIL.

tion methods may discard critical instances, leading to performance degradation. Specifically, highly scored instances are likely to be repeatedly selected across different representations, leading to excessive redundancy and diminishing the diversity of the aggregated instance-level information.

#Slide	Trans-MIL		Mixup-MIL		C^2Aug (Ours)	
	GPU Mem.	Training Time	GPU Mem.	Training Time	GPU Mem.	Training Time
1000	4.7 GB	123s	6.4 GB	164s	7.2 GB	223s
5000	4.7 GB	625s	6.4 GB	834s	7.3 GB	1001s
10000	4.9 GB	1260s	6.4 GB	1710s	7.3 GB	2320s
100000	4.9 GB	11881s	6.4 GB	16532s	7.3 GB	20530s

Table 7. Training Time & Memory / Epoch using Trans-MIL.

Ratio	# Bags	MVF	IC	IE
10%	89	52 (-**37**)	83 (-**6**)	89
10% $\sim$ 20%	8	12 (+4)	12 (+4)	7 (-1)
20% $\sim$ 40%	10	15 (+5)	11 (+1)	11 (+1)
40% $\sim$ 60%	3	9 (+6)	4 (+1)	3
60% $\sim$ 80%	1	5 (+4)	1	1
80% $\sim$ 100%	0	18 (+18)	0	0
total	111	111	111	111

Table 8. Number of tumor bags in the CAMELYON16 dataset after applying three augmentation methods, stratified by tumor instance ratio. 'Ratio': the tumor instance ratio within a tumor slide. # Bags: Number of bags falling within the corresponding ratio interval. Instance-level labels are obtained from dataset annotations.

Methods	CAMELYON-16			TCGA-LUNG			TCGA-BRCA		
	ACC	AUC	F1	ACC	AUC	F1	ACC	AUC	F1
C^2Aug (Ours)	$\mathbf{91.9_{1.0}}$	$\mathbf{94.7_{1.9}}$	$\mathbf{91.0_{0.8}}$	$\mathbf{91.2_{0.3}}$	$\mathbf{94.1_{1.7}}$	$\mathbf{88.7_{0.6}}$	$\mathbf{87.6_{0.8}}$	$\mathbf{92.5_{1.8}}$	$\mathbf{86.4_{0.5}}$
RetCCL	$90.8_{1.3}$	$93.3_{1.0}$	$90.3_{0.6}$	$90.4_{0.2}$	$92.9_{1.0}$	$85.9_{0.7}$	$86.3_{0.8}$	$90.9_{1.9}$	$85.2_{0.4}$
SCL-WC	$90.4_{0.9}$	$93.1_{2.1}$	$90.0_{0.5}$	$90.1_{0.4}$	$93.1_{1.4}$	$85.7_{0.6}$	$85.9_{1.2}$	$91.4_{1.4}$	$84.9_{0.5}$
MuRCL	$90.2_{1.1}$	$93.2_{1.8}$	$89.9_{0.7}$	$90.2_{0.5}$	$93.4_{1.9}$	$85.9_{0.4}$	$85.8_{0.9}$	$91.0_{1.1}$	$84.6_{0.4}$

Table 9. Comparison with Contrastive Learning-Based Methods.

Additional Results on the Cross-Bag Augmentation. To evaluate the effectiveness of C^{2}Aug, we conduct ablation studies on the TCGA-LUNG and CAMELYON-16 datasets using DTFD-MIL as the MIL model. Specifically, we assess the contribution of each component by removing Cross-Bag Augmentation (w/o CB), Multi-View Fusion (w/o MVF), Instance Compression (w/o IC), and Instance Expansion (w/o IE), respectively. As illustrated in Tab. 13 and Tab. 14, consistent with the results observed in the TransMIL model, the C^{2}Aug method demonstrates effectiveness on both DTFD-MIL and DSMIL. It achieves performance improvements of at least 1.4% on DSMIL and at least 3.4% on DTFD-MIL. In summary, C^{2}Aug achieves consistent performance improvements across the three datasets, demonstrating its effectiveness.

Noise Ratio	TransMIL			C^{2}Aug (Ours)		
	ACC	AUC	F1	ACC	AUC	F1
0%	$86.3_{0.8}$	$89.9_{1.4}$	$85.2_{1.6}$	$\mathbf{91.9_{1.0}}$	$\mathbf{94.7_{1.9}}$	$\mathbf{91.0_{0.8}}$
10%	$73.4_{0.4}$	$77.5_{0.7}$	$73.9_{0.8}$	$80.3_{0.7}$	$83.3_{0.7}$	$78.3_{1.1}$
20%	$65.3_{0.7}$	$62.8_{1.4}$	$65.2_{0.9}$	$73.3_{0.7}$	$77.6_{1.3}$	$70.1_{0.3}$

Table 10. Comparison of Trans-MIL and C^{2}Aug in Noise settings.

Methods	CAMELYON-16			TCGA-LUNG			TCGA-BRCA		
	ACC	AUC	F1	ACC	AUC	F1	ACC	AUC	F1
C^{2}Aug (Ours)	$\mathbf{91.9_{1.0}}$	$\mathbf{94.7_{1.9}}$	$\mathbf{91.0_{0.8}}$	$\mathbf{91.2_{0.3}}$	$\mathbf{94.1_{1.7}}$	$\mathbf{88.7_{0.6}}$	$\mathbf{87.6_{0.8}}$	$\mathbf{92.5_{1.8}}$	$\mathbf{86.4_{0.5}}$
2-shots	$52.3_{0.3}$	$40.3_{0.5}$	$42.3_{0.6}$	$54.4_{0.3}$	$42.3_{0.7}$	$44.3_{0.4}$	$53.9_{0.4}$	$41.9_{0.4}$	$44.1_{0.5}$
4-shots	$53.5_{0.4}$	$42.4_{0.4}$	$43.5_{0.4}$	$55.2_{0.2}$	$43.7_{0.4}$	$45.1_{0.5}$	$54.2_{0.5}$	$43.5_{0.3}$	$44.9_{0.2}$
8-shots	$54.7_{0.4}$	$44.7_{0.5}$	$44.7_{0.3}$	$56.3_{0.1}$	$45.1_{0.3}$	$45.7_{0.3}$	$54.6_{0.3}$	$44.7_{0.6}$	$45.0_{0.5}$

Table 11. Few-shot performance using TransMIL.

Methods	CAMELYON-16			TCGA-LUNG			TCGA-BRCA		
	ACC	AUC	F1	ACC	AUC	F1	ACC	AUC	F1
C^{2}Aug (Ours)	$\mathbf{91.9_{1.0}}$	$\mathbf{94.7_{1.9}}$	$\mathbf{91.0_{0.8}}$	$\mathbf{91.2_{0.3}}$	$\mathbf{94.1_{1.7}}$	$\mathbf{88.7_{0.6}}$	$\mathbf{87.6_{0.8}}$	$\mathbf{92.5_{1.8}}$	$\mathbf{86.4_{0.5}}$
PMIL	$89.3_{1.3}$	$92.9_{1.3}$	$87.1_{0.9}$	$90.8_{0.8}$	$93.2_{1.1}$	$87.8_{0.8}$	$86.6_{1.1}$	$86.9_{1.9}$	$81.2_{2.4}$
MergeUp	$90.8_{1.1}$	$93.8_{0.9}$	$89.4_{0.9}$	$89.8_{0.4}$	$92.7_{1.1}$	$88.0_{0.7}$	$86.8_{0.9}$	$91.3_{1.3}$	$85.3_{0.7}$
PseMix	$86.2_{2.9}$	$86.3_{4.1}$	$80.1_{4.7}$	$89.2_{0.4}$	$91.3_{1.1}$	$86.1_{0.7}$	$84.8_{1.8}$	$85.2_{2.1}$	$84.6_{1.9}$

Table 12. Comparison with other method using TransMIL.

2.2. t-SNE Visualization of Bag-level Contrastive Learning

t-SNE Visualization of L_{bag} and L_{group} on TCGA-LUNG and TCGA-BRCA Datasets. We visualize the bag-level feature representations of test samples from TCGA-LUNG and TCGA-BRCA datasets under models trained with and

Model	ResNet50			Prov-Gigapath		
	ACC	AUC	F1	ACC	AUC	F1
C^{2}Aug	$\mathbf{89.1_{1.2}}$	$\mathbf{93.2_{1.9}}$	$\mathbf{87.0_{1.0}}$	$\mathbf{96.9_{0.2}}$	$\mathbf{98.0_{0.3}}$	$\mathbf{95.7_{0.4}}$
w/o CB	$85.4_{1.7}$	$89.7_{1.6}$	$83.5_{1.9}$	$91.9_{0.3}$	$96.2_{0.3}$	$92.7_{0.5}$
w/o MVF	$86.1_{1.4}$	$90.0_{1.3}$	$84.3_{0.9}$	$93.1_{0.4}$	$96.4_{0.5}$	$93.9_{0.3}$
w/o IC	$86.6_{1.2}$	$90.4_{1.7}$	$85.4_{1.1}$	$93.5_{0.3}$	$97.0_{0.6}$	$94.2_{0.2}$
w/o IE	$87.7_{1.4}$	$91.3_{1.9}$	$85.9_{1.2}$	$93.9_{0.3}$	$97.3_{0.4}$	$94.7_{0.4}$

Table 13. Effects of Cross-Bag Augmentation on CAMELYON-16 dataset using DSMIL. 'CB': Cross-Bag, 'MVF': Multi-View Fusion, 'IC'/'IE': Instance Compression/Expansion.

Model	ResNet50			Prov-Gigapath		
	ACC	AUC	F1	ACC	AUC	F1
C^{2}Aug	$\mathbf{92.5_{0.9}}$	$\mathbf{94.3_{1.0}}$	$\mathbf{91.4_{0.8}}$	$\mathbf{97.9_{0.1}}$	$\mathbf{98.3_{0.3}}$	$\mathbf{97.7_{0.4}}$
w/o CB	$87.3_{1.2}$	$91.2_{1.5}$	$85.2_{1.7}$	$92.4_{0.3}$	$96.9_{0.5}$	$93.6_{0.2}$
w/o MVF	$87.8_{1.6}$	$91.4_{1.4}$	$85.9_{1.0}$	$93.7_{0.4}$	$96.9_{0.4}$	$95.0_{0.3}$
w/o IC	$88.0_{0.9}$	$92.3_{1.1}$	$86.2_{0.9}$	$94.0_{0.1}$	$97.4_{0.3}$	$95.2_{0.2}$
w/o IE	$89.1_{1.1}$	$92.5_{1.3}$	$87.6_{1.0}$	$94.5_{0.5}$	$97.7_{0.5}$	$95.4_{0.7}$

Table 14. Effects of Cross-Bag Augmentation on CAMELYON-16 dataset using DTFD-MIL. 'CB': Cross-Bag, 'MVF': Multi-View Fusion, 'IC'/'IE': Instance Compression/Expansion.

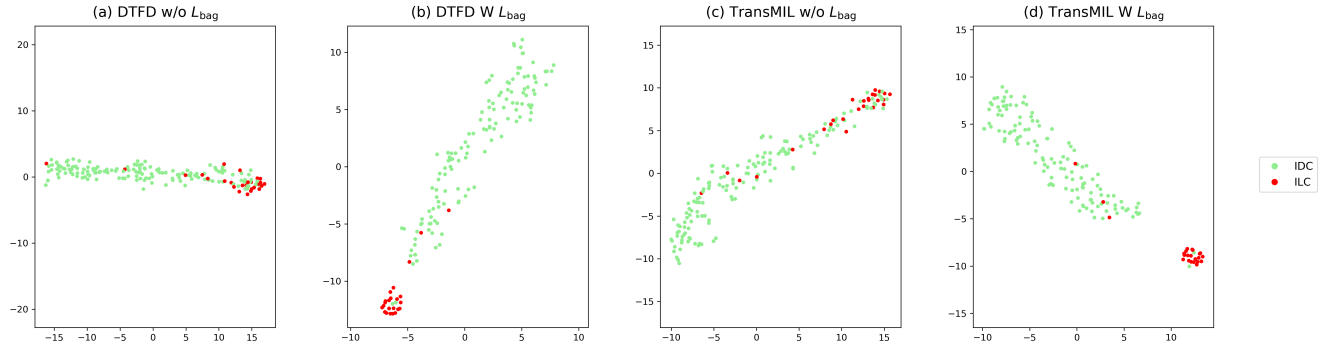

Figure 1. The tSNE visualization compares instance-level features from the TCGA-BRCA dataset for different models and conditions: (a) DTFD without L_{group}, (b) DTFD with L_{group}, (c) TransMIL without L_{group}, and (d) TransMIL with L_{group}, all using features extracted by ResNet50.

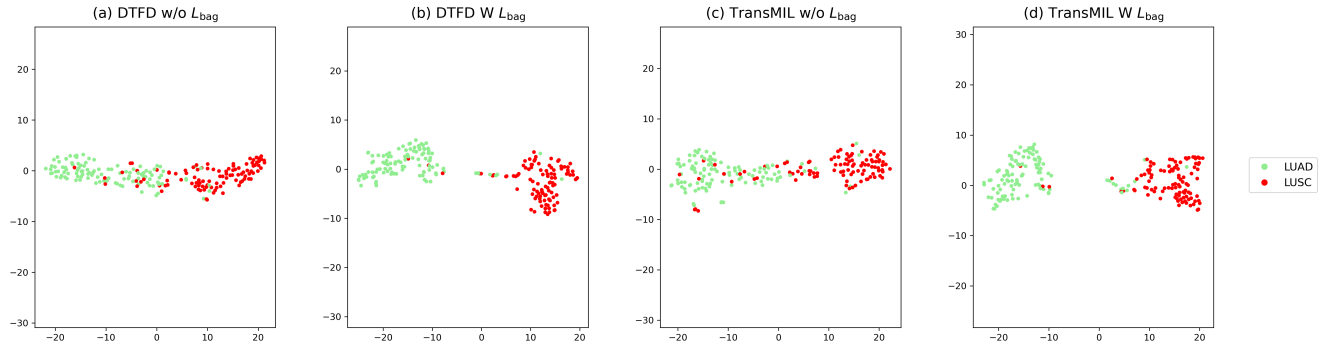

Figure 2. The tSNE visualization compares instance-level features from the TCGA-LUNG dataset for different models and conditions: (a) DTFD without L_{group}, (b) DTFD with L_{group}, (c) TransMIL without L_{group}, and (d) TransMIL with L_{group}, all using features extracted by ResNet50.

without L_{bag} to verify its effectiveness. As shown in Fig. 1 and Fig. 2, the results are consistent with those on the CAMELYON-16 dataset. The bag-level representations trained without L_{bag} present a dispersed distribution of tumor features. In comparison, the tumor bag-level features learned with L_{bag} show obvious clustering characteristics. Notably, we do not adopt tumor and normal as the classification criteria for these two datasets. Instead, we utilize fine-grained pathological subtypes for sample differentiation. Specifically, LUAD and LUSC are applied for TCGA-LUNG, while IDC and ILC are adopted for TCGA-BRCA.

Class Prototypes Visualization. Preliminary experimental results show that partial prototypes learn tumor-related features (Fig. 3 and Fig. 4). This indicates that different prototypes are responsible for capturing distinct semantic features.

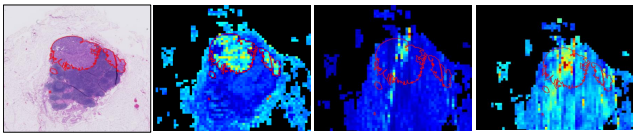

Figure 3. Cosine Similarity Between Class Prototypes and Instances.

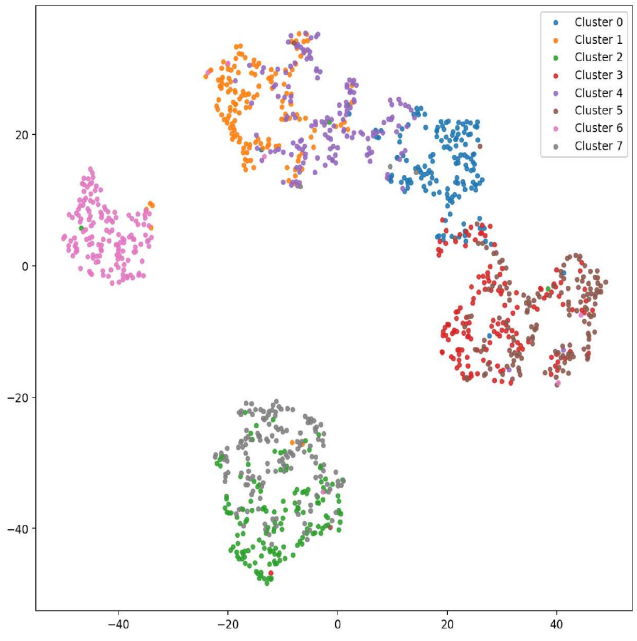

Figure 4. t-SNE of instances compressed by C class prototypes.

Algorithm 1 Multi-view Fusion Pseudo-code

Input:
Input Bags $\mathbf{X_i} \in \{\mathbf{x_{i,1}}, \mathbf{x_{i,2}}, ..., \mathbf{x_{i,L_i}}\} \in \mathbb{R}^{L_i \times D}$, label Y, $\texttt{num\_views} \in Z^+$
Output:
Output Bags $\mathbf{X_i^{'}} \in \mathbb{R}^{L \times D}$

Construct Multi-view Input:
$V_{\text{aug}} \leftarrow \texttt{random}(1, \texttt{m}) - 1$ ▷ Number of views
$\{V_0, V_1, ..., V_m\}$ where $V_m \in \mathbb{R}^{L_i \times D} \leftarrow \texttt{SampleInst}(X_1, X_2, ..., X_n \in Y_i)$ ▷ Randomly Sample Instances from bags with the same class as the input bag X_i
$\mathbf{X}_{\text{views}} \leftarrow \texttt{Concat}([X_{\text{i}}, V_1, V_2, ...V_m])$ where $X_{\text{views}} \in \mathbf{R}^{(m+1) \times L_i \times D}$ ▷ Concatenate X_i and V_{aug} together
Multi-View Fusion:
$\mathbf{X}_{\text{views}} \leftarrow \mathbf{X}_{\text{views}}.\text{unsqueeze}(0)$ ▷ Expand batch dimension
$\mathbf{X}_{\text{views}} \leftarrow \text{Norm}(\mathbf{X}_{\text{views}})$
$\mathbf{residual} \leftarrow \mathbf{X}_{\text{views}}[:, 0, :, :]$ ▷ Use the input bag as residual, shape $[B, L, D]$
$\mathbf{X}_{\text{views}} \leftarrow \mathbf{X}_{\text{views}}.\text{permute}(0, 2, 1, 3)$ ▷ Fuse each instance $x_{i,j}$ with its corresponding instaces from multiple views $v_{1,j}, v_{2,j}, ...m, v_{m,j}$
$\mathbf{K} = \mathbf{X_{views}}, \quad \mathbf{V} = \mathbf{X_{views}}, \quad \mathbf{Q} = \mathbf{X_i}$
$\mathbf{dots} \leftarrow \mathbf{Q} \times \mathbf{K}^T$, dots $\in \mathbb{R}^{B \times L \times m \times d}$ ▷ Matrix multiplication
$\mathbf{rand\_float} \leftarrow \texttt{rand}(B, L, m)$ ▷ Random float between [0, 1] for m views
$\mathbf{rand\_int} \sim \texttt{randint}(0, m, \text{size} = (B, L))$ ▷ Random int between [0, m] for m views
$\mathbf{indices} \leftarrow \text{sort}(\mathbf{rand}, \text{dim} = -1)$ ▷ Used to generate mask
$\mathbf{mask} \leftarrow \mathbf{indices} < \mathbf{rand\_int}.\text{unsqueeze}(-1)$ ▷ Generate mask for dots
$\mathbf{dots}[\mathbf{mask} > \mathbf{0}] \leftarrow -9999$ ▷ Mask out selected instances (Row-Wise Random Masking)
$\mathbf{attn} \leftarrow \text{Softmax}(\mathbf{dots}), \quad \mathbf{X_i^{'}} \leftarrow \mathbf{attn} \cdot \mathbf{v}$
$\mathbf{X_i^{'}} \leftarrow \text{Reshape}(\mathbf{X_i^{'}}, \text{shape} = (B, L, D))$
$\mathbf{X_i^{'}} \leftarrow \mathbf{X_i^{'}} + \mathbf{residual}$
$\mathbf{X_i^{'}} \leftarrow \text{Norm}(\mathbf{X_i^{'}})$
Output: $\mathbf{X_i^{'}}$

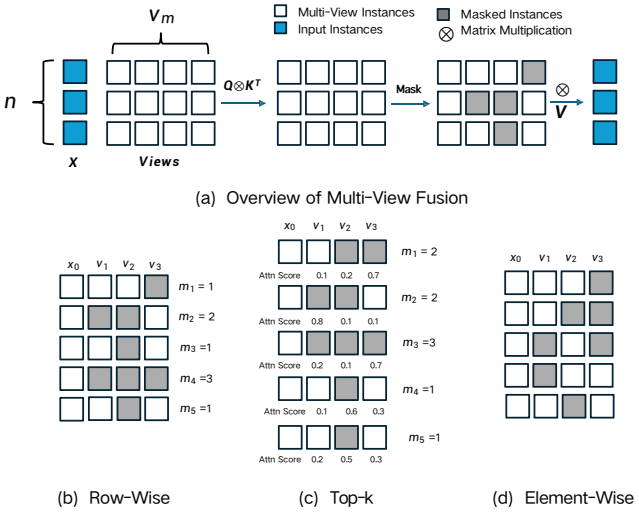

Figure 5. Illustration of Multi-View Fusion.

2.3. Details on Multi-View Fusion

Multi-View Fusion augmentation applies a random view approach to augment each instance. We introduce three methods to mask the QK^T matrix: Top-k Random Masking, Row-wise Random Masking, Element-wise Random Masking, as illustrated in Fig. 5. Specifically, for each input bag $X_i \in R^{L_i \times D}$, and the multi-view $V_{aug} \in \{V_1, V_2, ..., V_m\}$, where $V_m \in R^{L_i \times D}$, and $V_{aug} \in R^{m \times L_i \times D}$. We first perform matrix multiplication between X_i and V_{aug} along the view dimension m, resulting a matrix $dots$ with shape $R^{L_i \times m}$. Then we perform random view for each instance by masking elements within $dots$, and multiple X_i with $dots$ to obtain the augmented view $X_i^{'}$ as the output. We have proposed three random masking methods (b) Row-Wise Random Masking. (c) Top-k Random Masking. (d) Element-Wise Random Masking. Row-Wise Random Masking mask out each row from $dots$ by first generating a random int $m^{'}$ from $[1, m]$, then randomly mask out $m^{'}$ instances within $dots$. Top-k random masking performs a similar approach with Row-Wise Random Masking, but differently, they mask out instances based on the top-$m^{'}$ attention score of each instance. These attention scores are obtained from a base MIL model, such as TransMIL. Element-wise random masking discards instances with a given prob-

ability, analogous to dropout.

The difference between row-wise and element-wise random masking is that, for each value of $m^{'}$, row-wise random masking has an equal probability of masking out instances, whereas in element-wise random masking, as $m^{'}$ increases, the number of masked instances follows a binomial distribution.

3. Pseudo Code

To facilitate a clearer understanding of our Multi-View Fusion module, we present the detailed pseudo-code in Algorithm 1. Specifically, for each input instance X_i, we first retrieve other instances sharing the same label and concatenate them with X_i to form a multi-view input $\mathbf{X}_{\text{views}}$. A cross-attention mechanism is then applied to fuse these views into a new representation. In this process, the keys and values are derived from $\mathbf{X}_{\text{views}}$, while the queries are from the original X_i. To perform random masking, we first generate a random integer $m^{'}$ for each instance, sampled uniformly from the interval $[1, m]$. Then, $m^{'}$ elements are masked out in each row of the matrix $dots$. The masked $dots$ is subsequently multiplied with the input bag X_i to produce the final output $X_i^{'}$.

4. Code and Data Availability

The source code will be released upon acceptance.

The CAMELYON-16 dataset is available at https://camelyon16.grand-challenge.org.

The TCGA-Lung and TCGA-BRCA datasets are available at https://portal.gdc.cancer.gov.

The patch-level feature extraction script is available at https://github.com/mahmoodlab/TRIDENT.

References

[1] Yuan-Chih Chen and Chun-Shien Lu. Rankmix: Data augmentation for weakly supervised learning of classifying whole slide images with diverse sizes and imbalanced categories. In *CVPR*, pages 23936–23945, 2023. 2

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR'16*, pages 770–778, 2016. 1

[3] Honglin Li, Yunlong Zhang, Pingyi Chen, Zhongyi Shui, Chenglu Zhu, and Lin Yang. Rethinking transformer for long contextual histopathology whole slide image analysis. *NeurIPS*, 37:101498–101528, 2024. 2

[4] Hanwen Xu, Naoto Usuyama, Jaspreet Bagga, Sheng Zhang, Rajesh Rao, Tristan Naumann, Cliff Wong, Zelalem Gero, Javier González, Yu Gu, et al. A whole-slide foundation model for digital pathology from real-world data. *Nature*, 630 (8015):181–188, 2024. 1

[5] Andrew Zhang, Guillaume Jaume, Anurag Vaidya, Tong Ding, and Faisal Mahmood. Accelerating data processing and benchmarking of ai models for pathology. *arXiv preprint arXiv:2502.06750*, 2025. 1