Supplementary Materials: SAM-MIL

Anonymous Authors

1 DATASET DESCRIPTION

CAMELYON-16 [1] dataset is proposed for studies on metastasis detection in breast cancer. Comprising a total of 400 Whole Slide Images (WSIs), the dataset is officially divided into 270 samples for training and 130 samples for testing, with the testing samples constituting approximately one-third of the total. Following [2, 8, 12], we employed a method of three-times three-fold cross-validation to minimize the impact of data splitting and random seeding on model evaluation, ensuring each slide is utilized in both training and testing phases. Under this arrangement, each fold contains about 133 slides. The mean and standard deviation of performance metrics are reported based on the results of three iterations.
TCGA Lung Cancer dataset encompasses two specific lung cancer sub-types: Lung Adenocarcinoma (LUAD) and Lung Squamous Cell Carcinoma (LUSC). It consists of diagnostic whole slide images, featuring 541 slides of LUAD from 478 distinct cases, and 512 slides of LUSC, also from 478 cases. We applied a 5-fold cross-validation strategy to this dataset, guaranteeing comprehensive evaluation. The results are summarized through the mean and standard deviation of the performance metrics across the five testing folds.

Following prior works [8–11], we crop each WSI into a series of 512 × 512 non-overlapping patches. The background region, including holes, is discarded as in CLAM [8].

2 DATA PREPROCESSING

In order to integrate the segmentation of SAM into the preprocessing workflow of WSIs, it is necessary to adapt the traditional preprocessing protocols. We adopt the data preprocessing protocol of CLAM [8], as illustrated in Fig 1. The preprocessing workflow of WSIs comprises three stages: Foreground Segmentation, Patching & SAM Segmentation, and Feature & SAM Info Extraction.

To ensure high-quality segmentation and extraction of relevant organizational patches, as well as convenient use, we are integrating the Foreground Segmentation and Patching & SAM Segmentation stages into a single process. Under user-defined parameters, we will segment the original WSI slide and create patches, then segment the foreground areas of each slide using SAM. This integrated process yields five distinct types of output files: *Masks* in H5 format, *Patches* also in H5 format, *Segments* stored as PKL files, *Stitches* again in H5 format, and a comprehensive *Process List* in CSV format. Following the outputs from the first steps, the second stage will independently perform feature extraction on each segmented patch using a pretrained model, as well as extract group features with SAM. This stage will produce three distinct types of files: *h5_files* in H5 format, which store both coordinates and extracted features; *pt_files* in PT format, which contain only the features of the patches; and *seg_files*, also in H5 format, which hold the segmentation information from SAM. In subsequent training, our MIL framework will require the feature files for each patch and the SAM segmentation results, along with their respective segmentation areas.

In addition to the standard process outlined, we also offer options to generate the required *pt* and *seg* files from *h5_files*, *Segments*, and *Process List*, as well as from original *pt_files*, *Segments*, and *Process List*. This design allows our preprocessing workflow to be applicable to already-extracted tile features, reducing the time spent on feature extraction. The relevant preprocessing code will be made available in the code repository along with the entire project.

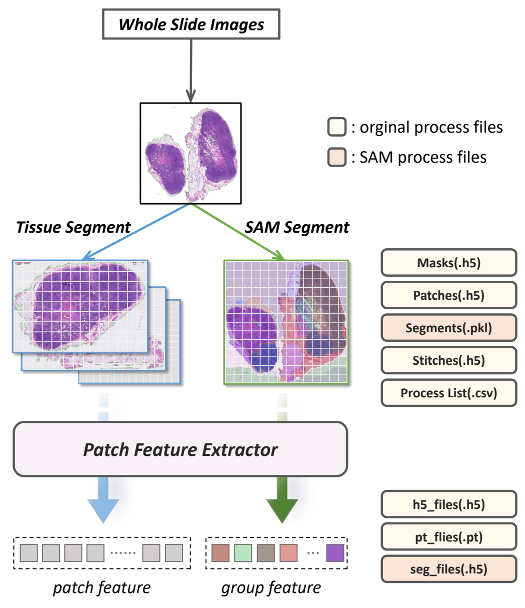

Figure 1: WSIs feature extraction process.

3 CONFIGURATION OF SAM

The Segment Anything Model (SAM) produces high quality object masks from input prompts such as points or boxes, and it can be used to generate masks for all objects in an image. It has been trained on a dataset of 11 million images and 1.1 billion masks, and has strong zero-shot performance on a variety of segmentation tasks. Since SAM can efficiently process prompts, masks for the entire image can be generated by sampling a large number of prompts over an image. In our study, we employed the SamAutomaticMaskGenerator class from the official repository to automatically generate masks in images. We made adjustments to the default parameters, which are detailed in the code file. We utilized the official checkpoint for the *vit_h* model as the pretrained weights in our implementation.

4 IMPLEMENTATION DETAILS

Following [8, 9, 11], we use the ResNet-50 model [4] pretrained with ImageNet [3] as the backbone network to extract an initial feature vector from each patch, which has a dimension of 1024. The last convolutional module of the ResNet-50 is removed, and a global average pooling is applied to the final feature maps to generate the initial feature vector. The initial feature vector is then reduced to a 512-dimensional feature vector by one fully-connected layer.

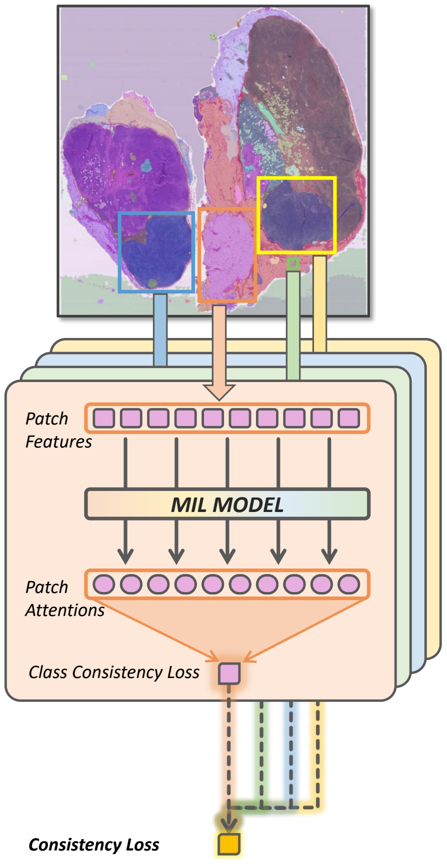

Figure 2: Illustration of proposed consistency loss.

In order to validate the reliability of the extracted group features, experiments were conducted on both ResNet50 [4] and PLIP [5] extracted features. The feature extraction process of PLIP adheres to the specifications outlined in the official repository. All the models are trained for 200 epochs with an early-stopping strategy. The patience of CAMELYON-16 and TCGA are 30 and 20, respectively. We do not use any trick to improve the model performance, such as gradient cropping or gradient accumulation. The batch size is set to 1. All the experiments are conducted with NVIDIA GPUs. Section 11 gives all codes and weights of the pre-trained model.

5 PSEUDOCODES OF SAM-MIL

Algorithm 1 gives the details about SAM-MIL training. The pseudocode illustrates only the training component of the code, omitting the data preprocessing section.

6 CONSISTENCY-BASED ITERATIVE OPTIMIZATION

To effectively constrain the training of pseudo-bags, we propose a method for calculating spatial context-based consistency loss, as shown in Fig 2. First, our model processes input data to extract global attention weights for each token. Assuming visual similarity, we hypothesize that patches visually similar within the model should receive similar attention weights. Specifically, if the SAM classifies certain patches as belonging to the same category, the attention weights of these patches should also exhibit high consistency. Further, leveraging the category information provided by SAM, we group all patches by category. Within each category group, we independently calculate the consistency loss of attention weights. This step evaluates the variance of attention weights among patches within the group, aiming to minimize the weight differences between similar patches. Subsequently, we sum the consistency losses of all groups to obtain the total consistency loss for the entire input slide. Finally, this consistency loss is integrated into the model's overall optimization loss, guiding the model to better differentiate between patch categories during training. This enhances the model's sensitivity and processing capabilities for subtle visual differences. This approach not only enhances the model's understanding of spatial context but also promotes its robustness when dealing with images that have complex backgrounds and style variations.

Strategy	CAMELYON-16		TCGA	
	AUC	F1 Score	AUC	F1 Score
baseline(AB-MIL)	94.54	87.44	94.27	88.69
w/o consistency loss	95.79	89.23	95.66	91.33
w/ consistency loss	**96.08**	**89.36**	**96.01**	**91.42**

Table 1: Impact of spatial context-based consistency loss.

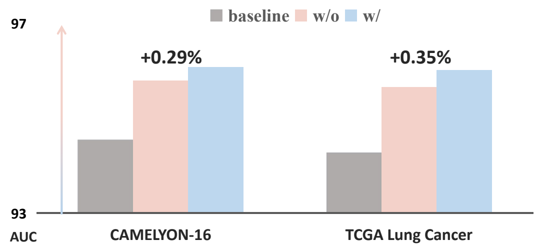

Figure 3: Impact of spatial context-based consistency loss.

The experimental results, as shown in Table 1, demonstrate the effectiveness of consistency loss for model training. Figure 3 illustrates the improvements achieved by the model under the constraints of consistency loss.

7 EFFECT OF SAM GUIDED GROUP FEATURE

Our proposed SAM-Guided Feature Extractor's group features are suitable not only for our designed MIL framework but also for application to various mainstream MIL models, thereby achieving immediate performance improvements. In our experiments, features extracted by ResNet50 [4] and PLIP [5] were processed under the guidance of SAM and validated in the CAMELYON-16 dataset. Besides the previously mentioned baseline AB-MIL, we selected two advanced MIL models [9, 10] to assess the effectiveness of our group features. The complete corresponding experimental results are displayed in Table 2. Figure 5 illustrates the improvements achieved by the group features. The results indicate that incorporating features extracted by SAM led to performance improvements in both methods. Furthermore, it is observed that, due to the lack of pre-training on medical images, the features extracted

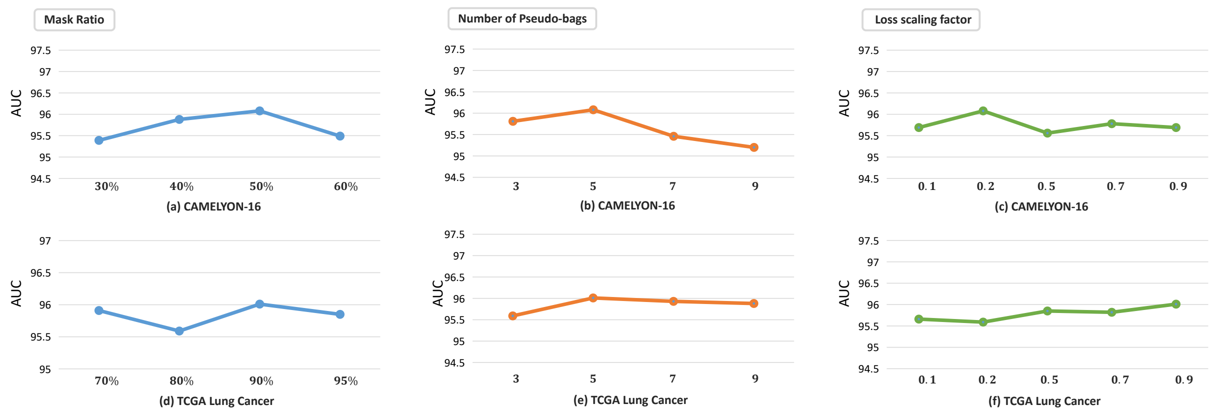

Figure 4: Discussion of important hyper-parameters.

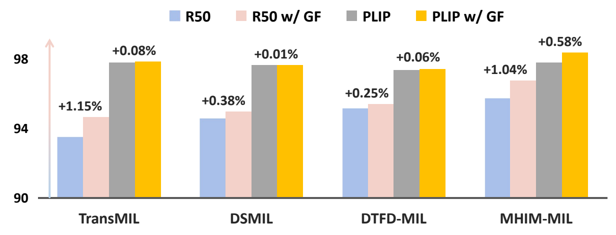

Figure 5: Effect of Group Features in Different Benchmarks.

using ResNet50 are somewhat inferior in performance compared to those extracted using PLIP. However, after supplementing with group features based on spatial context extraction, the features extracted by ResNet50 demonstrated a greater improvement in performance. This potentially suggests that the group features we proposed provide significant guidance for models lacking prior knowledge. Overall, our SAM-Guided Feature Extractor can serve as a plug-and-play module applicable to various mainstream MIL models.

8 DISCUSSION OF HYPER-PARAMETER IN SAM-MIL

The bottom part of Figure 4 shows the results. We can find that SAM-MIL is not sensitive to this parameter, and different values can achieve high-level performance. This reflects the generality of the preset optimal parameter in different scenarios.

9 ADDITIONAL VISUALIZATION

To more intuitively understand the effect of the spatial contextual awareness, we visualize the tumor probabilities (cyan patch) of patches produced by AB-MIL and SAM-MIL, as illustrated in Figure 6. Here, SAM-MIL employs AB-MIL as its baseline model. From the visualization on the left side of the Figure 6, it is observed that in larger tumor areas, AB-MIL often assigns high tumor probabilities to patches in non-tumor areas. This phenomenon can be attributed to the limited generalization capabilities of conventional attention-based MIL models, which focus predominantly on salient regions during training. In contrast, our model demonstrates enhanced concentration of tumor probabilities in tumor areas and reduced misclassifications in non-tumor areas, indicating superior classification performance. Furthermore, the visualization on the right side of the Figure 6 reveals that our baseline model tends to overlook smaller tumor areas, leading to decreased performance. However, SAM-MIL, benefiting from the introduced spatial context, can effectively recognize and correctly classify these smaller tumor regions.

Feature	CAMELYON-16	
	AUC	F1 Score
TransMIL [9]		
w/ R50	93.51	85.10
w/ R50 + Group Feature	94.66 (+1.15)	86.27 (+1.17)
w/ PLIP	97.77	92.77
w/ PLIP + Group Feature	97.85 (+0.08)	92.23 (-0.54)
DSMIL [7]		
w/ R50	94.57	87.65
w/ R50 + Group Feature	94.95 (+0.38)	87.64 (-0.01)
w/ PLIP	97.64	93.24
w/ PLIP + Group Feature	97.65 (+0.01)	93.24 (-0.01)
DTFD-MIL [11]		
w/ R50	95.15	87.62
w/ R50 + Group Feature	95.40 (+0.25)	90.56 (+2.94)
w/ PLIP	97.35	94.86
w/ PLIP + Group Feature	97.41 (+0.06)	95.03 (+0.23)
MHIM-MIL [10]		
w/ R50	95.72	88.98
w/ R50 + Group Feature	96.76 (+1.04)	91.51 (+2.53)
w/ PLIP	97.79	94.13
w/ PLIP + Group Feature	98.37 (+0.58)	94.70 (+0.57)

Table 2: Effect of Group Features in Different Benchmarks.

10 LIMITATION

Although the integration of spatial context information through the SAM significantly improves the classification performance of WSIs by facilitating the construction of relationships between instances, it is important to note that the incorporation of SAM necessitates additional data preprocessing time and computational resources. Moreover, the performance improvement attributable to SAM-MIL

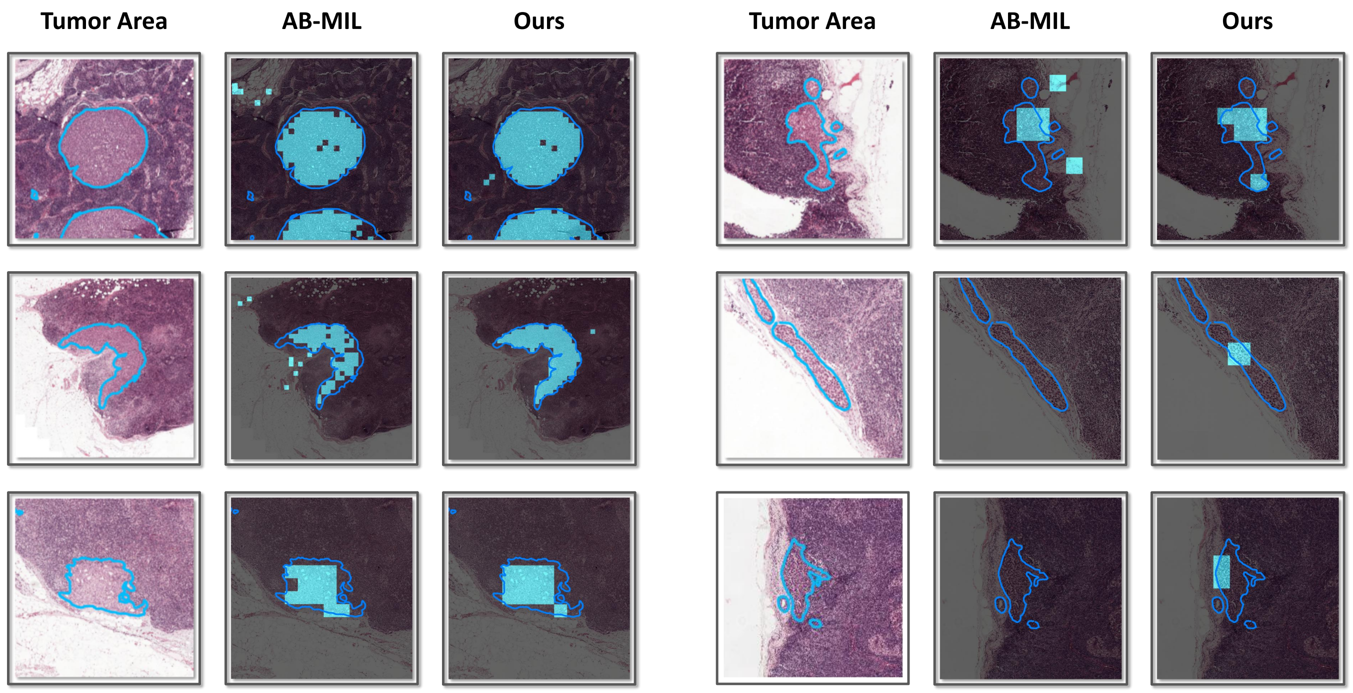

Figure 6: Comparison of patch visualization produced by AB-MIL [6] (baseline) and SAM-MIL. The blue lines outline the tumor regions. The cyan colors indicate high probabilities of being tumor for the corresponding locations. Ideally, the cyan patches should cover only the area within the blue lines. The visualization on the left focuses on the tumor regions within a larger area, while the visualization on the right concentrates on smaller-sized tumor regions.

for WSIs tasks depends largely on SAM's segmentation of visual information in WSIs. However, extremely small tumor areas might be overlooked, potentially leading to misclassification. The absence of medical images in the SAM pre-training data might hinder SAM's ability to perfectly segment tissue and cellular information in WSIs, with visually similar tissues potentially being misclassified into the same category. Therefore, optimizing the segmentation performance of SAM is a focal point for our future work. We can explore the introduction of SAMs pre-trained on medical datasets to enhance the segmentation of WSIs and improve the overall performance of SAM.

11 CODE AND DATA AVAILABILITY

The source code of our project will be uploaded at https://anonymous.4open.science/r/SAM-MIL.

CAMELYON-16 dataset can be found at https://camelyon16.grand-challenge.org.

All TCGA datasets can be found at https://portal.gdc.cancer.gov.

The official Segment Anything Model repository is at https://github.com/facebookresearch/segment-anything. The pre-trained weights can be found at https://dl.fbaipublicfiles.com/segment_anything/sam_vit_h_4b8939.pth.

The script of slide pre-processing and patching modified from CLAM repository at https://github.com/mahmoodlab/CLAM.

The code and weights of PLIP can be found at https://github.com/PathologyFoundation/plip.

REFERENCES

[1] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes Van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen AWM Van Der Laak, Meyke Hermsen, Quirine F Manson, Maschenka Balkenhol, et al. 2017. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *JAMA* 318, 22 (2017), 2199–2210. 1
[2] Richard J Chen, Chengkuan Chen, Yicong Li, Tiffany Y Chen, Andrew D Trister, Rahul G Krishnan, and Faisal Mahmood. 2022. Scaling vision transformers to gigapixel images via hierarchical self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16144–16155. 1
[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255. 1
[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778. 1, 2
[5] Zhi Huang, Federico Bianchi, Mert Yuksekgonul, Thomas J Montine, and James Zou. 2023. A visual–language foundation model for pathology image analysis using medical twitter. *Nature Medicine* (2023), 1–10. 2
[6] Maximilian Ilse, Jakub Tomczak, and Max Welling. 2018. Attention-based deep multiple instance learning. In *ICML*. PMLR, 2127–2136. 4
[7] Bin Li, Yin Li, and Kevin W Eliceiri. 2021. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *CVPR*. 14318–14328. 3
[8] Ming Y Lu, Drew FK Williamson, Tiffany Y Chen, Richard J Chen, Matteo Barbieri, and Faisal Mahmood. 2021. Data-efficient and weakly supervised computational pathology on whole-slide images. *Nature Biomedical Engineering* 5, 6 (2021), 555–570. 1
[9] Zhuchen Shao, Hao Bian, Yang Chen, Yifeng Wang, Jian Zhang, Xiangyang Ji, et al. 2021. Transmil: Transformer based correlated multiple instance learning for whole slide image classification. *NeurIPS* 34 (2021). 1, 2, 3
[10] Wenhao Tang, Sheng Huang, Xiaoxian Zhang, Fengtao Zhou, Yi Zhang, and Bo Liu. 2023. Multiple Instance Learning Framework with Masked Hard Instance Mining for Whole Slide Image Classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4078–4087. 2, 3
[11] Hongrun Zhang, Yanda Meng, Yitian Zhao, Yihong Qiao, Xiaoyun Yang, Sarah E Coupland, and Yalin Zheng. 2022. DTFD-MIL: Double-Tier Feature Distillation

Multiple Instance Learning for Histopathology Whole Slide Image Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18802–18812. 1, 3

[12] Xiaoxian Zhang, Sheng Huang, Yi Zhang, Xiaohong Zhang, Mingchen Gao, and Liu Chen. 2022. Dual Space Multiple Instance Representative Learning for Medical Image Classification. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press. https://bmvc2022.mpi-inf.mpg.de/0768.pdf 1

Algorithm 1: PyTorch-style pseudocode for SAM-MIL training scheme

```
# f: model networks
# mr: mask ratio
# num_groups: number of pseudo bags
# alpha: pseudo-bag loss scaling factor
# beta: consistency loss scaling factor

# SAM-Guided Group Mask
def mask_fn(x,unique_areas, mask_ratio):
   # Initialize lists for retained and masked indices
   final_retained_idxs = []
   final_masked_idxs = []
   len_keep = 0
   for area in unique_areas do
      # Get indices for the current area
      area_indices = where(area == area)
      # Calculate the number of points in the area
      area_ps = size(area_indices)
      adjusted_random_ratio = adjusted_sigmoid(area) * mask_ratio
      # Select indices to mask based on adjusted ratio
      area_len_keep, area_mask_ids = select_mask_fn(area_ps, adjusted_random_ratio)
      # Map local indices to global indices
      retained_idxs_global = map_to_global(area_indices, area_mask_ids[0][:area_len_keep])
      masked_idxs_global = map_to_global(area_indices, area_mask_ids[0][area_len_keep:])
      # Store global indices
      final_retained_idxs.append(retained_idxs_global)
      final_masked_idxs.append(masked_idxs_global)
      # Update total number of points retained
      len_keep += area_len_keep
   end
   # Concatenate indices for all groups
   retained_idxs = concatenate(final_retained_idxs)
   masked_idxs = concatenate(final_masked_idxs)
   # Merge all indices into one tensor
   mask_ids = concatenate([retained_idxs, masked_idxs])
   return mask_ids

# main loop
for x,y,sam in loader: # load a minibatch x,y,sam with N slides
   unique_areas = sam.getUniqueAreas()
   # get masked instance index
   mask_id = mask_fn(x,unique_areas,mr)
   # mask instance
   x_masked = masking(x,mask_id)
   logits,bag_feats,_ = f.forward(x_masked)

   # Shuffle and group the pseudo-bag
   indices = torch.randperm(bag_feats.size(1))
   shuffled_bag = bag_feats[:, indices, :]
   group_size = shuffled_bag.size(1) // num_group
   bag_groups = [shuffled_bag[:, i * group_size:(i + 1) * group_size, :] for i in range(num_group)]
   label_groups = [y for _ in range(num_group)]
   # Train each group and calculate losses
   group_results = [f.train(group) for group in bag_groups]
   bag_loss = sum(result["cls_loss"] for result in group_results) / len(group_results)

   # consistency loss
   global_attn = f.compute_attn_with_grad(bag_feats, label=y)
   sam_class = sam.getClass()
   consistency_loss = calculate_consistency_loss(global_attn, sam_class, con_batch_size)

   # total loss
   cls_loss = CrossEntropy(logits,y)
   loss_all = cls_loss + alpha * bag_loss + beta * consistency_loss

   # Adam update
   loss_all.backward()
   update(f.params)
```