[Supplementary] Multiple Instance Learning Framework with Masked Hard Instance Mining for Whole Slide Image Classification

1. Additional Visualization

Here, we attempt to further analyze the impact of Masked Hard Instance Mining (Masked HIM) on WSI classification algorithms based on multiple instance learning. As shown in Figure 1, we visualize the masked instances (middle column), which we call the mined hard instances, to illustrate the relationship between the instance-level tumor prediction probability (cyan patch) and model attention (bright patch) before and after Masked HIM training.

First, thanks to the outstanding saliency patch mining ability of traditional attention-based MIL models, Masked HIM can effectively mask out the most salient regions to indirectly mine hard instances while using random masking to mitigate over-fitting problems. Moreover, as shown in the Figure 2, this discriminatory power improves gradually during the training. To ensure that the instance sequence after masking still retains key instance information related to the slide category, we propose a randomization technique, which will be explained in detail in the following subsection. Second, contrary to intuition, MIL models do not lose their discriminative power for key regions after masking out the most salient instances, due to the MHIM-MIL framework. Instead, they achieve a significant improvement. Figure 1 strongly proves that focusing only on salient instances during the training stage damages the discriminative power of MIL models, and verifies the huge help of hard instances for MIL model training. Moreover, we visualize the instance patch attention after softmax, which can be regarded as the contribution to the final bag embedding. We find that although traditional MIL models seem to pay attention to salient regions, they do not make reasonable use of this part of the information. They ignore most features and extremely focus on individual features in the feature aggregation process, damaging model discriminativeness. In contrast, MIL models trained with Masked HIM seem to put more attention on more "irrelevant regions", but better utilize key region features to generate higher quality bag features and improve model performance.

Figure 9 shows more patch visualizations on the CAMELYON-16 dataset.

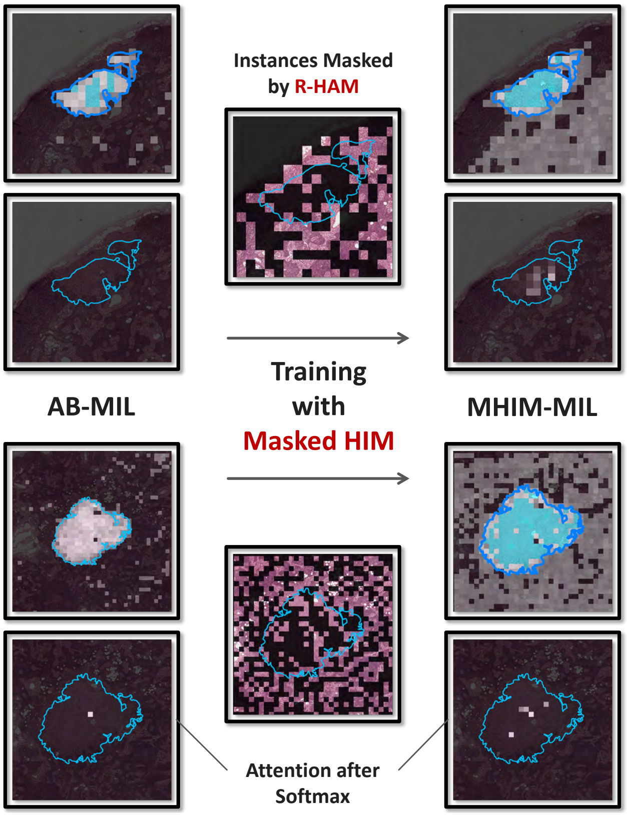


Figure 1: Comparison of patch visualization produced by AB-MIL [5] (baseline) and MHIM-MIL. The blue lines outline the tumor regions. The brighter patch indicates higher attention scores. The cyan colors indicate high probabilities of being tumor for the corresponding locations. Ideally, the cyan patches should cover only the area within the blue lines. In the middle column, the dark patches denote masked instances.

2. Additional Quantitative Experiments

2.1. More on Masked Hard Instance Mining

Discussion on Mask Ratio. We explored how various mask ratios affect MHIM-MIL training in this section. We fixed other ratios (β_r, β_l) and varied high attention mask ratio β_h

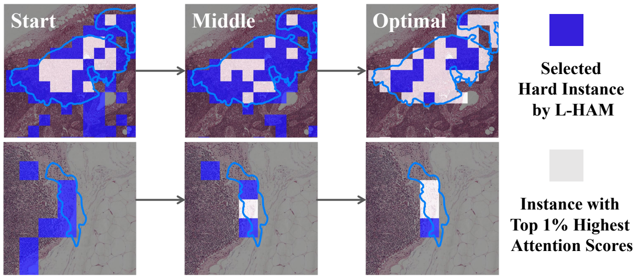

Start
Middle
Optimal
Selected Hard Instance by L-HAM
Instance with Top 1% Highest Attention Scores

Figure 2: Patch visualization during iteration process.

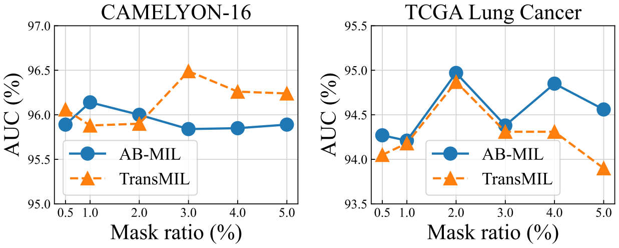


Figure 3: The performances of MHIM-MIL under different high attention mask ratio β_h.

random ratio	low ratio	AUC
AB-MIL		
60%	20%	94.57
70%	10%	94.65
70%	**20%**	**94.97**
70%	30%	94.55
80%	20%	94.49
TransMIL		
50%	20%	94.60
60%	10%	93.97
60%	**20%**	**94.87**
60%	30%	94.37
70%	20%	94.60

(a) TCGA Lung Cancer dataset

random ratio	low ratio	AUC
AB-MIL		
40%	0%	95.90
50%	0%	96.14
50%	**20%**	**95.92**
60%	0%	96.13
TransMIL		
0%	70%	96.36
0%	**80%**	**96.49**
20%	80%	96.33
0%	90%	96.10

(b) CAMELYON-16 dataset

Table 1: Comparison of different random attention mask ratio β_r and low attention mask ratio β_l on both datasets.

alone in Figure 3. We fixed β_h and changed different β_r and β_l in Table 1. Our findings are: 1) A low β_h reduces the difficulty of mined instances, thereby diminishing the overall model performance. Moreover, the randomized trick ensures that the model training does not collapse even at high β_h. More details are provided in the following section. 2) Compared to AB-MIL [5], TransMIL [8] has lower discriminative power for salient instances. This is why TransMIL needs a bigger β_h. 3) MHIM-MIL training is less sensitive to β_r and β_l than to β_h. However, choosing an appropriate mask ratio is still crucial for optimal performance. Specifically, we observed that combining three strategies on the CAMELYON-16 dataset decreases classification performance. We attribute this to excessive instance masking losing important information on the CAMELYON-16 dataset.

Computational Cost. Here, we comprehensively discuss the impact of different MHIM strategies on the computational cost of model training. Table 2 shows the efficiency gains brought by large-scale low-attention masking and random-attention masking. This is especially significant for TransMIL [8], a baseline with both spatial and temporal complexity quadratic to the number of instances. Large-scale masking greatly reduces the input of the student model, thereby reducing memory and time consumption. Although the input of the teacher model is still full length, due to the application of momentum teacher, it hardly introduces extra training cost. In addition, we also find that mixing multiple strategies further reduces the number of instances but also introduces additional computation, which is more obvious on AB-MIL [5] baseline.

Mask Ratio Decay. The discriminative ability of the model improves and stabilizes as training goes on. We follow the learning rate decay idea and tune β_h based on training progress to prevent a high initial ratio from hurting later training. We name this technique mask ratio decay and adopt a classic cosine decay function to regulate decay speed. Table 3 demonstrates that this trick significantly boosts performance. We note that we apply the decay strategy only to β_h while maintaining initial values for the other two ratios during training.

Randomly High Attention Masking. MHIM faces a major challenge: it may mask all key information and turn into "error instance mining". We apply the Randomly High Attention Masking technique to address this issue and make sure that mined hard instances include key instance information for the slide category. Figure 4 illustrates our approach: we select instances with the highest 2 × β_h% attention scores as candidate states and randomly mask half

Model	C16	TCGA	Para.	Time	Mem.
AB-MIL	94.00	93.17	657K	**4.0s**	2.4G
HAM	95.68	93.83	657K	**4.0s**	2.7G
R-HAM	**96.14**	**94.79**	657K	4.3s	2.3G
L-HAM	95.81	94.33	657K	4.2s	2.3G
LR-HAM	95.92	94.97	657K	4.4s	**2.2G**
TransMIL	93.51	92.51	2.67M	13.1s	10.6G
HAM	95.90	94.54	2.67M	15.9s	10.3G
R-HAM	95.88	94.60	2.67M	10.3s	5.5G
L-HAM	**96.49**	94.67	2.67M	**10.1s**	5.5G
LR-HAM	96.33	**94.87**	2.67M	**10.1s**	**5.4G**

Table 2: Comparison of time and memory requirements of different masked hard instance mining strategies. We report the model size (Para.), the training time per epoch (Time), and the peak memory usage (Mem.) on the CAMELYON-16 dataset (C16).

Strategy	CAMELYON-16		TCGA	
	AB.	Trans.	AB.	Trans.
$\beta_h\%$	96.04	96.07	94.34	94.56
$\beta_h\% \rightarrow 0\%$	**96.14**	**96.49**	**94.97**	**94.87**

Table 3: Comparison results of applying high attention mask ratio decay.

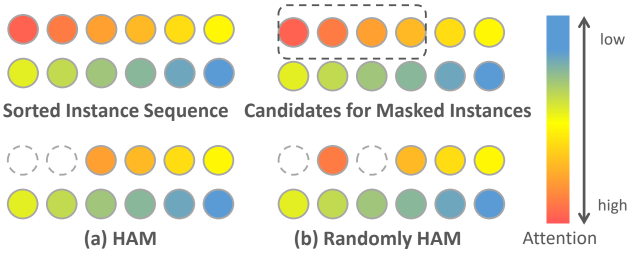


Figure 4: Illustration of Randomly High Attention Masking (Randomly HAM).

of them to keep some key information. Table 4 demonstrates this technique suffers from low training difficulty in the TCGA dataset, where the tumor area ratio is high (typically over 40% [8]), and impairs the discriminability of the model. On the other hand, this technique performs well on the CAMELYON-16 dataset, indicating that it can preserve key information in original instances.

2.2. Initialization of Student Network

MIL models typically employ a fully connected layer to project original 1024-dimensional instance features into 512 dimensions as final instance representation. In MHIM-MIL, we initialize the fully connected layer of the student network with pre-trained parameters to reduce collapse risk from the Siamese structure. [1] elaborates on more details about collapse risk. Figure 5 illustrates how this initialization affects teacher model performance. An uninitialized student model has slow initial training which drags down teacher model performance and harms the iterative optimization of the framework. The upper part of Table 5 displays a large margin in final student model performance with and without this initialization. Moreover, we applied the same initialization to mainstream MIL models to investigate if this initialization boosts performance by aiding Siamese structure optimization. The upper part of Table 5 reveals that this initialization does not noticeably enhance the performance of existing mainstream MIL models and sometimes lowers it. Our experiments confirm that initializing the first fully connected layer of student facilitate the iterative optimization of the MHIM-MIL framework instead of being a universal trick for increasing MIL model performance.

Strategy	CAMELYON-16		TCGA	
	AB.	Trans.	AB.	Trans.
w/o Ran. HAM	95.71	96.37	**94.97**	**94.87**
w/ Ran. HAM	**96.14**	**96.49**	94.52	94.17

Table 4: Comparison results of applying randomly high attention masking (Ran. HAM).

Model	CAMELYON-16	TCGA
AB-MIL w/ init	93.98 (-0.02)	92.75 (-0.42)
MHIM-MIL w/ init	**96.14 (+0.63)**	**94.97 (+0.49)**
TransMIL w/ init	94.22 (+0.71)	93.36 (+0.85)
MHIM-MIL w/ init	**96.49 (+0.90)**	**94.87 (+0.95)**
w/ init		
CLAM-SB	94.53 (-0.12)	93.43 (-0.24)
DSMIL	94.96 (+0.39)	93.93 (+0.22)
DTFD-MIL	95.23 (+0.08)	93.80 (-0.03)

Table 5: Comparison results of different initialized MIL models.

2.3. Transformer Attention

Transformer typically consists of a multi-layer multi-head structure where each head within each layer generates independent attention scores. Thus, extracting the most effective attention score among them is very challenging. In particular, the baseline model TransMIL [8] comprises two layers with eight heads per layer. We separately examined the effect of attention scores from different layers and vari-

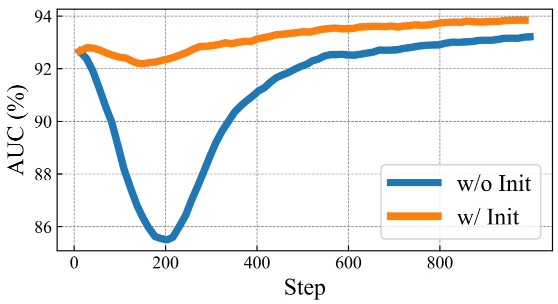
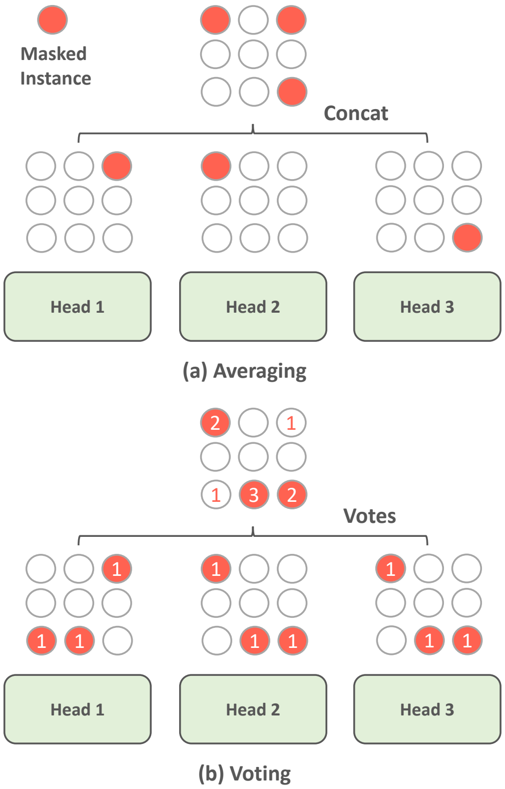


Figure 5: Performance comparison of teacher models under initialized or uninitialized student networks.



(a) Averaging

(b) Voting

Figure 6: Illustration of averaging and voting multi-head attention fusion strategy.

ous multi-head fusion strategies. The upper part of Table 6 demonstrates the advantage of attention scores from the first layer over those from the final layer. We attribute this to the first layer producing more accurate attention scores for identifying hard instances. This is because the multi-head self-attention (MSA) operation modifies original features which causes a large deviation between hard instances mined by the last layer and the actual situation, while only the input of the first layer is the original instance features.

Additionally, prior work [3] equalizes the contribution of each head and distributes the total mask count among different heads, which is called "averaging". However, this strategy fails to prevent the effect of the invalid heads on MHIM. As shown in Figure 7, some heads of TransMIL lack discrimination ability for instances and produce identical attention scores which we term as invalid heads. Invalid heads dilute localization accuracy for hard instances under averaging strategy and impair the training of MHIM-MIL. To mitigate this issue, we suggest a voting strategy that employs majority rule to eliminate noise from invalid heads, as shown in Figure 6. The lower part of Table 6 proves the effectiveness of this strategy.

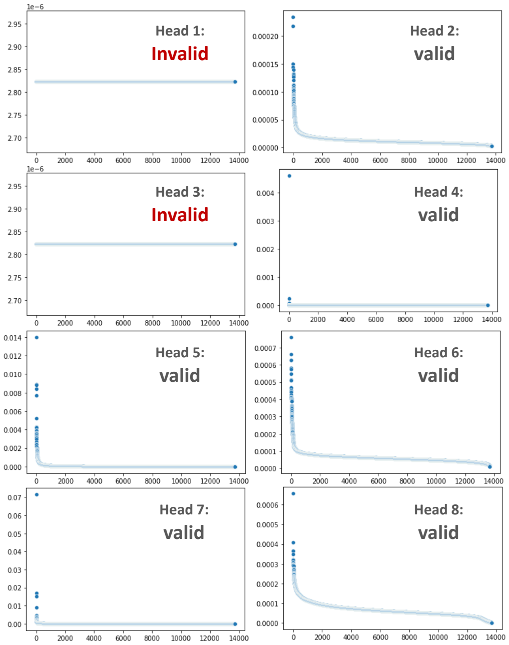


Figure 7: Attention visualization of different heads in TransMIL first layer.

case	CAMELYON-16	TCGA
first	**96.49**	**94.87**
last	95.58 (-0.91)	93.90 (-0.97)
averaging	96.38 (-0.11)	94.40 (-0.47)
voting	**96.49**	**94.87**

Table 6: Comparison results of variants of TransMIL attention.

2.4. Discussion on Hyperparameter

Here, we provide a systematic discussion of an important hyperparameter α in our framework. It balances the impact of self-supervised and fully supervised information during model training. Figure 8 demonstrates that α affects

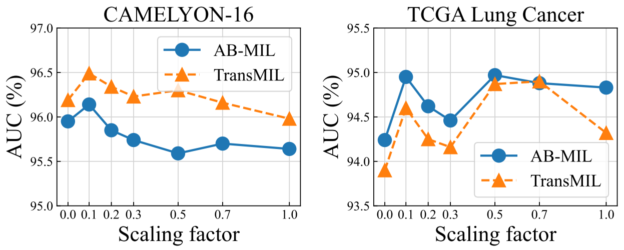


Figure 8: The performances of MHIM-MIL under different loss scaling factors α.

the training of both models consistently, with values that are either too high or too low resulting in biased training. Particularly, when α is too high, it impairs the positive effect of slide labels on model learning. This effect is more pronounced on the CAMELYON-16 dataset, as the model frequently misclassifies some challenging slides, requiring supervision from slide labels.

3. Data Pre-processing

Following prior works [7–9], we crop each WSI into a series of non-overlapping patches of size 256 × 256 at 20X magnification and discard the background region, including holes, as in CLAM [7]. After pre-processing, we obtain a total of 3.6M patches from the CAMELYON-16 dataset, with an average of about 9000 patches per bag, and 10.8M patches from the TCGA Lung Cancer dataset, with an average of about 10300 patches per bag.

4. Implementation Details

Following [7–9], we use the ResNet-50 model [4] pretrained with ImageNet [2] 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. The momentum rate of EMA is 0.9999 and the temperature of consistency loss is 0.1. An Adam optimizer [6] with learning rate of 2×10^{-4} and weight decay of 1×10^{-5} is used for the model training. The Cosine strategy is adopted to adjust the learning rate. All the models are trained for 200 epochs with an early-stopping strategy. The patience of CAMELYON-16 and TCGA Lung Cancer 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 were conducted with an NVIDIA RTX3090 GPU.

5. Pseudocode

We present the PyTorch-style pseudocode for the training scheme of MHIM-MIL in Algorithm 1.

6. Limitation

In this paper, we propose a Masked Hard Instance Mining MIL framework to indirectly mine hard instances in the absence of instance supervision information. Although this strategy can effectively alleviate the over-reliance problem of traditional MIL models on salient instances, it is also challenging to accurately assess the difficulty level of instances and mine the most helpful hard instances for training. Compared with traditional hard sample mining strategies based on supervision information, this sub-optimal and rough strategy affects the convergence speed and discriminability of the model. In future work, we will focus on how to accurately evaluate instance difficulty level in the absence of complete supervision and use the most beneficial instances to facilitate model training.

7. Code and Data Availability

The source code of our project will be uploaded at https://github.com/DearCaat/MHIM-MIL.

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

TCGA Lung Cancer dataset can be found at https://portal.gdc.cancer.gov.

The script of slide pre-processing and patching can be found at https://github.com/mahmoodlab/CLAM.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 3

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[3] Ju He, Jie-Neng Chen, Shuai Liu, Adam Kortylewski, Cheng Yang, Yutong Bai, and Changhu Wang. Transfg: A transformer architecture for fine-grained recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 852–860, 2022. 4

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

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

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[7] 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. 5

[8] 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. *NeurIPS*, 34, 2021. 2, 3, 5

[9] Hongrun Zhang, Yanda Meng, Yitian Zhao, Yihong Qiao, Xiaoyun Yang, Sarah E Coupland, and Yalin Zheng. Dtfd-mil: Double-tier feature distillation multiple instance learning for histopathology whole slide image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18802–18812, 2022. 5

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

```
# f_t, f_s:  teacher and student networks
# f_p:  the pretrained network
# mrh:  high attention mask ratio
# mrl:  low attention mask ratio
# mrr:  random attention mask ratio
# m:  momentum rates
# tp:  temperatures
# a:  consistency loss scaling factor

# initialize
f_t.params = f_p.params
f_s.proj_head.params = f_p.proj_head.params

# teacher network not introduces any parameter
f_t = f_t.eval()

def mask_fn(attn,mask_ratio,largest):
   # sort attention score and get the topk index
   attn = sort(attn)
   topk_ids = topk(attn,k=int(mask_ratio*attn.length),largest=largest)
   # init vote matrix
   vote = 0
   # voting and counting
   vote[topk_ids] = 1
   vote = sum(vote)
   # get mask index
   mask_ids = topk(vote,k=int(mask_ratio*attn.length))

   return mask_ids

for x,y in loader:  # load a minibatch x,y with N slides
   # get attention scores from teacher
   _,bag_feats_t,attn_t = f_t.forward(x)
   # stop gradient of teacher network
   bag_feats_t = bag_feats_t.detach()

   # get masked instance index
   # High Attention Masking
   mask_h = mask_fn(attn_t,mrh,True)
   # Low Attention Masking
   mask_l = mask_fn(attn_t,mrl,False)
   # Random Attention Masking
   mask_r = random_select(attn_t,mrr)
   # Combine all index
   mask_all = mask_h & mask_l & mask_r

   # masked hard instance mining
   x_hard = masking(x,mask_all)

   logits_s,bag_feats_s,_ = f_s.forward(x_hard)

   # consistency loss
   loss_con = -softmax(bag_feats_t / tp) * log_softmax(bag_feats_s)
   # label prediction loss
   loss_cls = CrossEntropy(logits_s,y)
   loss_all = loss_cls + a*loss_con

   # Adam update:  student network
   loss_all.backward()
   update(f_s.params)

   # EMA update:  teacher network
   f_t.params = m*f_t.params+(1-m)*f_s.params

   # high attention mask ratio decay
   CosineDecay(mrh)
```

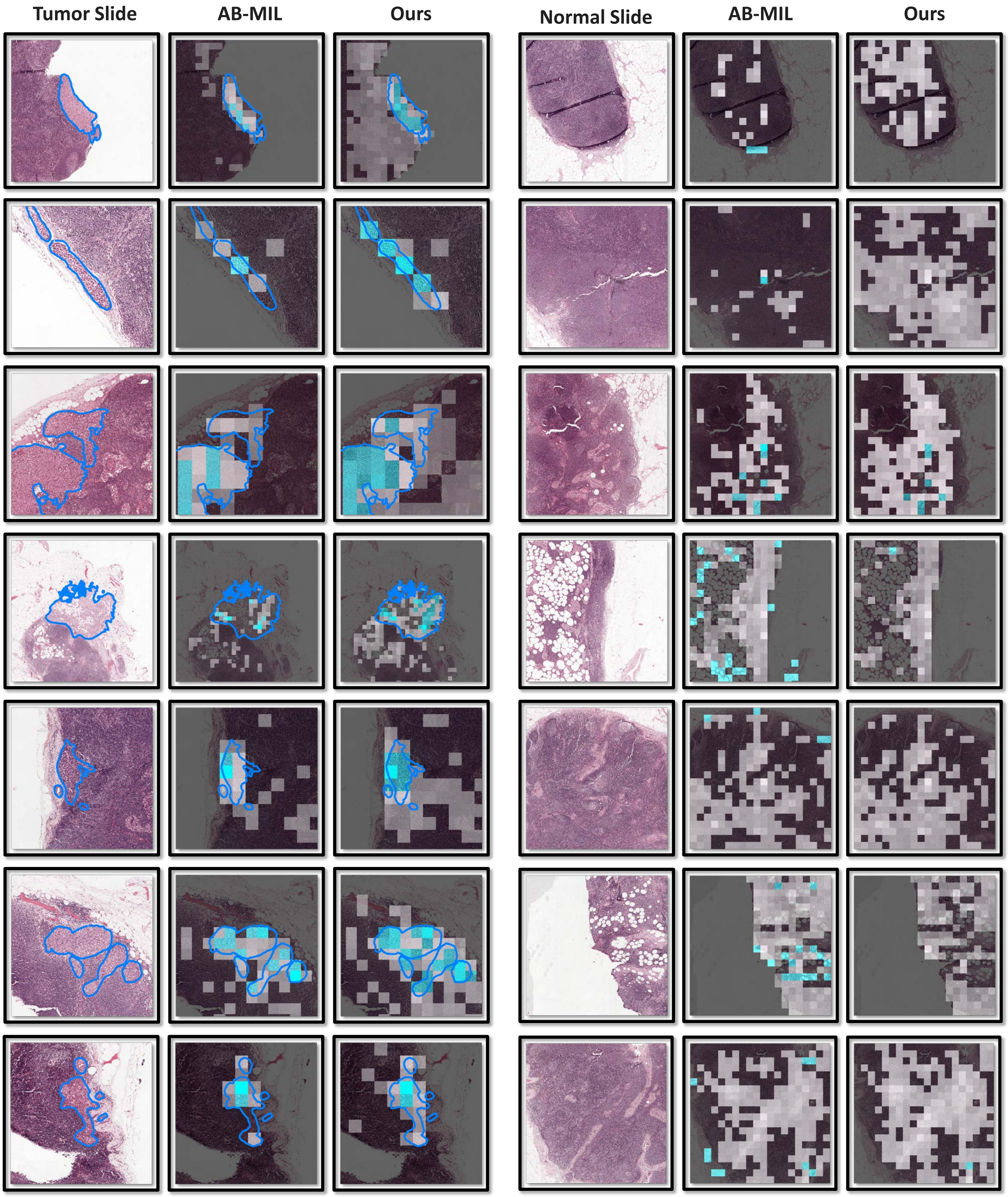


Figure 9: More comparisons of patch visualization between AB-MIL (baseline) and MHIM-MIL. Best viewed in color.