A PROOF OF LEMMA 1

We initialize the $\boldsymbol{w}$ as $\mathbf{0}$ and first update $\boldsymbol{w}$ for T steps, where each step uses one slide to compute the loss and gradient. In the optimization for $\boldsymbol{w}$, we use a learning rate of η and if the prediction for a slide is right, i.e., $\hat{Y}_i > 0.5$ for a positive patch or $\hat{Y}_i < 0.5$ for a negative patch, the η is set as 0. After we update the $\boldsymbol{w}$, the attention variables are optimized.

proof. The gradient for $\boldsymbol{w}$ and $\boldsymbol{a}$

$$\frac{\partial \mathcal{L}_i}{\partial \boldsymbol{w}} = \boldsymbol{H}_i \boldsymbol{a}(\hat{Y}_i - Y_i), \tag{11}$$

$$\frac{\partial \mathcal{L}_i}{\partial \boldsymbol{a}} = \boldsymbol{H}_i^T \boldsymbol{w}(\hat{Y}_i - Y_i), \tag{12}$$

$$\frac{\partial \mathcal{L}_i}{\partial \boldsymbol{u}} = (\boldsymbol{I}_{\boldsymbol{a}} - \boldsymbol{A}^2)\boldsymbol{H}_i^T \boldsymbol{w}(\hat{Y}_i - Y_i), \tag{13}$$

where $\boldsymbol{I}_{\boldsymbol{a}} = \boldsymbol{I}\boldsymbol{a}$ and $\boldsymbol{A}^2 = [\boldsymbol{a} \odot \boldsymbol{a}, \cdots, \boldsymbol{a} \odot \boldsymbol{a}] \in \mathbb{R}^{K \times K}$.

Consider the inner product of $\boldsymbol{w} - \eta \frac{\partial \mathcal{L}_i}{\partial \boldsymbol{w}}$ and the optimal $\boldsymbol{w}^*$,

$$(\boldsymbol{w} - \eta \frac{\partial \mathcal{L}_i}{\partial \boldsymbol{w}})^T \boldsymbol{w}^* = \boldsymbol{w}^T \boldsymbol{w}^* - (\hat{Y}_i - Y_i)\boldsymbol{a}^T \boldsymbol{H}_i^T \boldsymbol{w}^*. \tag{14}$$

If $Y_i = 0$, $\boldsymbol{H}_i^T \boldsymbol{w}^* < \mathbf{0}$, we have

$$(\boldsymbol{w} - \eta \frac{\partial \mathcal{L}_i}{\partial \boldsymbol{w}})^T \boldsymbol{w}^* \geq \boldsymbol{w}^T \boldsymbol{w}^* + \eta\gamma\hat{Y}_i \geq \boldsymbol{w}^T \boldsymbol{w}^* + \frac{1}{2}\eta\gamma \tag{15}$$

If $Y_i = 1$, we have,

$$(\boldsymbol{w} - \eta \frac{\partial \mathcal{L}_i}{\partial \boldsymbol{w}})^T \boldsymbol{w}^* = \boldsymbol{w}^T \boldsymbol{w}^* + \eta(1 - \hat{Y}_i)\boldsymbol{a}^T \boldsymbol{H}_i \boldsymbol{w}^* \geq \boldsymbol{w}^T \boldsymbol{w}^* + \eta\zeta(1 - \hat{Y}_i) \geq \boldsymbol{w}^T \boldsymbol{w}^* + \frac{1}{2}\eta\zeta. \tag{16}$$

Next consider the scale of $\boldsymbol{w}$,

$$(\boldsymbol{w} - \eta \frac{\partial \mathcal{L}_i}{\partial \boldsymbol{w}})^T (\boldsymbol{w} - \eta \frac{\partial \mathcal{L}_i}{\partial \boldsymbol{w}}) = \boldsymbol{w}^T \boldsymbol{w} - 2\eta(\hat{Y}_i - Y_i)\boldsymbol{a}^T \boldsymbol{H}_i^T \boldsymbol{w} + \eta^2(\hat{Y}_i - Y_i)^2 \boldsymbol{a}^T \boldsymbol{H}_i^T \boldsymbol{H}_i \boldsymbol{a} \tag{17}$$

$$\leq \boldsymbol{w}^T \boldsymbol{w} + \eta^2 \tag{18}$$

After T steps of update, since the $\boldsymbol{w}$ is initialized as $\mathbf{0}$, we have

$$\boldsymbol{w}^{*T} \boldsymbol{w} \geq \frac{1}{2}\eta T \max(\gamma, \zeta), \|\boldsymbol{w}\| \leq \eta\sqrt{T} \tag{19}$$

$$\frac{\boldsymbol{w}^{*T} \boldsymbol{w}}{\|\boldsymbol{w}\|\|\boldsymbol{w}^*\|} \geq \frac{\frac{1}{2}\eta T \max(\gamma, \zeta)}{\eta\sqrt{T}} = \frac{1}{2}\sqrt{T}\max(\gamma, \zeta) \tag{20}$$

As a result of the Cauchy-Schwartz inequality,

$$\frac{\boldsymbol{w}^{*T} \boldsymbol{w}}{\|\boldsymbol{w}\|\|\boldsymbol{w}^*\|} \leq 1, \tag{21}$$

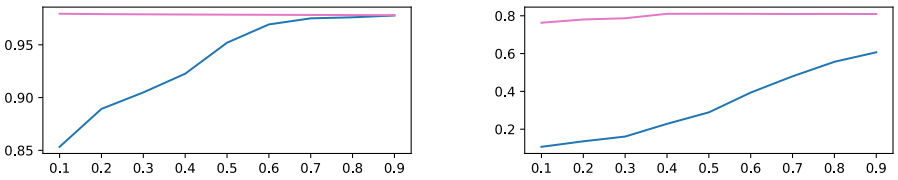
$$T \leq \frac{4}{\max(\gamma, \zeta)^2}. \tag{22}$$

Next look at the update for $\boldsymbol{a}$,

$$\frac{\partial \mathcal{L}_i}{\partial \boldsymbol{u}} = (\boldsymbol{I}_{\boldsymbol{a}} - \boldsymbol{A}^2)\boldsymbol{H}_i^T \boldsymbol{w}(\hat{Y}_i - Y_i) \approx \boldsymbol{I}_{\boldsymbol{a}} \boldsymbol{H}_i^T \boldsymbol{w}_T(\hat{Y}_i - Y_i) \tag{23}$$

$$= [a_1 \boldsymbol{h}_{i,1}^T \boldsymbol{w}_T, \cdots, a_K \boldsymbol{h}_{i,K}^T \boldsymbol{w}_T](\hat{Y}_i - Y_i). \tag{24}$$

We know that $\boldsymbol{w}_T$ converges to the optimal $\boldsymbol{w}^*$ so the $\boldsymbol{h}_{i,j}^T \boldsymbol{w}_T > 0$ if the jth patch is positive and $\boldsymbol{h}_{i,j}^T \boldsymbol{w}_T < 0$ if jth patch is negative. so this update means that the first K_p values of $\boldsymbol{u}$ will increase for positive slides while the last K_n values will decrease for any slides. This iteration leads to the desired optimal $\boldsymbol{a}^*$.



(a) Patch-level accuracy. (b) Patch-level precision.

Figure 8: The patch-level performance of positive slides of CAMELYON16 under different decision thresholds $\{0.1, 0.2, \ldots, 0.9\}$. The blue line is CLAM and the red line is Bayes-MIL.

B EXPERIMENTS

Hyper-parameters For the backbone MIL model, we use the the same parameter setup as CLAM. We use Adam with learning rate 10^{-4} and weight decay 10^{-5} for optimization. We use early stop with patient step 20 and maximum epoch 200.

Efficiency The running time information of different methods, tested on Camelyon16, are shown in Tab. 3. By using an efficient variational inference design and implementation, we keep a decent running time of Bayes-MIL between CLAM and TransMIL.

Table 3: Running time of the MIL methods.

	CLAM	TransMIL	DSMIL	Bayes-MIL
training time (minute per epoch)	0.606	1.8	0.824	1.217
testing time (second per slide)	0.062	5.857	0.045	0.185

Supplemental results on CAMELYON16 The full CRF (8) generate the best slide-level results on CAMELYON16. The incomplete results with 3-fold splits of the dataset show the testing accuracy is 0.9015, the testing AUROC is 0.9546. As the computation efficiency is low, we stopped running the experiments.

Results on normal slides of CAMELYON16 The numerical results (accuracy) on Camelyon16 are shown in Tab. 4. Bayes-MIL consistently presents high accuracy while other methods (e.g., CLAM) do not provides satisfactory results.

Table 4: Patch-level results on normal slides of CAMELYON16

Methods	Acc (threshold=0.1)	Acc (threshold=0.3)	Acc (threshold=0.5)
DSMIL	0.0058	0.1188	0.5547
CLAM	0.2758	0.6650	0.8395
TransMIL	0.9846	0.9898	0.9920
Bayes-MIL-APCRF	**0.9986**	**0.9996**	**0.9998**

Results on tumor slides of CAMELYON16 The comparisons of patch-level performance between CLAM and the data uncertainty of Bayes-MIL-APCRF are shown in Fig. 8. It could be concluded that Bayes-MIL presents consistently decent performance under different threshold while CLAM should explore its optimal threshold.

More results on CAMELYON17 The patch-level and slide-level results on CAMELYON17 are shown in Tab. 5. The ablation study shows the effectiveness of the three step model design. Bayes-MIL has better performance than the CLAM baseline.

Note that BMIL-APCRF has a slightly lower slide-level performance in terms of accuracy and AUC, but still competitive compared with previous works. The reason could be that in order to utilize the

efficient standard computation library, $\tilde{\boldsymbol{a}}$ is reshaped to a $W \times F$ rectangle. As a result, some areas are 0 in the rectangle, which might introduce ambiguity for classification. In future work, we will implement an efficient convolution for the irregular feature map to improve the performance, which is orthogonal to this work.

Table 5: Patch-level and Slide-level results on CAMELYON17

	Patch-level			**Slide-level**		
	P-Prec. (↑)	P-FROC (↑)	P-AUC (↑)	S-Acc. (↑)	S-AUC (↑)	S-ECE (↓)
CLAM	0.3833	0.4454	0.8121	0.8055±0.03	0.8222±0.03	0.2589±0.01
Bayes-MIL-Vis	0.3875	0.4545	0.8222	0.8088±0.02	0.8276±0.05	0.2547±0.02
Bayes-MIL-SDPR	0.4001	0.4591	0.8265	**0.8185**±0.03	**0.8612**±0.01	0.2455±0.01
Bayes-MIL-APCRF	**0.4131**	**0.4600**	**0.8312**	0.8180±0.03	0.8597±0.02	**0.2453**±0.02

More results on TCGA-NSCLC Bayes-MIL uses $p(a_k|Y=1) = \mathcal{LN}(\mu_1, \sigma_1)$ as the prior for the tumor subtype classification, as no negative slides are provided. Results in Tab. 6 shows Bayes-MIL outperforms existing methods in accuracy and calibration, while providing AUC higher than CLAM baseline. Note this result is preliminary without search for optimal setup. Future work will explore more detailed setups on the tumor subtype classification dataset.

Table 6: Tumor subtype classification results on TCGA-NSCLC.

Methods	AUC	Accuracy	ECE
CLAM	0.9420 ±0.03	0.8640 ±0.04	0.1697±0.02
Scaling ViT	0.9516	0.8821	N/A
TransMIL	**0.9603**	0.8835	N/A
Bayes-MIL-Enc	0.9440±0.01	0.8893±0.04	0.1647±0.01
Bayes-MIL-APCRF	0.9451±0.01	**0.8966**±0.01	**0.1575**±0.02

Tumor stage classification results on CAMELYON17. Bayes-MIL uses $p(a_k|Y=1) = \mathcal{LN}(\mu_1, \sigma_1)$ as the prior for different stages of the tumors, while uses $p(a_k|Y=0) = \mathcal{LN}(\mu_0, \sigma_0)$. This is consistent with the original design for the SDPR. A softmax function is adopted for the slide level classification.

Table 7: Tumor stage classification results on CAMELYON17.

Methods	AUC	Accuracy	ECE
CLAM	0.7803	0.6	0.4138
Bayes-MIL-APCRF	**0.8070**	**0.64**	**0.4017**

C THE PSEUDO CODES FOR TRAINING AND INFERENCE

The training algorithm is as follows:

$\mathbf{H} \longleftarrow \text{feature\_extractor}(\mathbf{X})$
while not converged **do**
  $\boldsymbol{\pi}, \mathbf{W} \sim q_\phi(\boldsymbol{\pi}, \mathbf{W})$.
  Calculate a_k with (3).
  $\bar{a}_k = \text{APCRF}(a_k)$.
  Calculate $L_{\mathcal{D}}(\boldsymbol{\phi}), R_{\boldsymbol{\pi},\boldsymbol{W}}(\phi), R_{\bar{\boldsymbol{a}}}(\phi)$ and optimize with the objective (10).
end while

The inference algorithm includes an Monte-Carlo integration which takes S Monte-Carlo samples:

$\mathbf{H} \longleftarrow \text{feature\_extractor}(\mathbf{X})$
for s=1:S **do**
  $\boldsymbol{\pi}_s \sim q_\phi(\boldsymbol{\pi})$.

Calculate a_{ks} with (3).
end for
$a_k = \frac{1}{S}\sum_s a_{ks}$
$\bar{a}_k = \mathrm{APCRF}(a_k)$.
$\hat{y}_i = \sigma(\boldsymbol{w}^T \boldsymbol{H}_i \bar{\boldsymbol{a}} + b)$.

D DETAILS ON VARIATIONAL INFERENCE

We use the multiplicative Gaussian with log-uniform prior proposed by Kingma et al. (2015) for the MIL parameters, as the variational parameters of posterior mean could be canceled in KL term for easier optimization. Molchanov et al. (2017) revised the formulation of KL term by an approximation for sparsity of neural network. Cui et al. (2021) further extends the approximation for KL to have a flexible range. Specifically, the prior $p(\log|\pi|) = \text{const}$. The posterior is

$$\pi = \theta\eta = \theta(1+\sqrt{\phi}\epsilon_\pi),\ \epsilon_\pi \sim \mathcal{N}(0,1), \quad \pi \sim \mathcal{N}(\pi|\theta, \phi\theta^2). \tag{25}$$

By taking a mean-field approximation, the KL term is

$$-D_{\mathrm{KL}}[q(\pi|\theta,\phi)||p(|\pi|)] = \frac{1}{2}\log\alpha - \mathbb{E}_{\epsilon_\pi\sim\mathcal{N}(1,\phi)}\log|\epsilon_\pi| + C, \tag{26}$$

For a flexible range of α, the KL term is approximated by

$$R_{\boldsymbol{\pi},\boldsymbol{W}}(\phi) = \mathrm{KL}[q_{\boldsymbol{\phi}}(\boldsymbol{\pi},\boldsymbol{W})||p(\boldsymbol{\pi},\boldsymbol{W})] = \sum_{\phi\in\Phi} b_1 e^{-e^{b_4\cdot(b_2+b_3*\log\phi)^2}} - 0.5\log(1+\phi^{-1}) + C, \tag{27}$$

where $b_1 = 0.7294$, $b_2 = -0.2041$, $b_3 = 0.3492$ and $b_4 = 0.5387$. Φ is the whole set of variational parameters corresponding to $\{\boldsymbol{\pi}, \mathbf{W}\}$. Refer to Molchanov et al. (2017) and Cui et al. (2021) for more details.

As Y is a hard variable of 0 or 1 during training process, (6) only chooses one component for each input slide. The KL term of SDPR is analytically written as

$$R_{\bar{\boldsymbol{a}}}(\phi) = \sum_k \log\frac{\sigma'}{\bar{\sigma}_k} + \frac{\bar{\sigma}_k^2 + (\bar{\mu}_k - \mu')^2}{2\bar{\sigma}_k^2} - 0.5 \tag{28}$$

where $\bar{\mu}_k = C_{\boldsymbol{w},\boldsymbol{h}}(f_\mu(\boldsymbol{\pi},\boldsymbol{H}_i))_k, \bar{\sigma}_k = f_\sigma(\boldsymbol{\pi},\boldsymbol{H}_i))_k$ are conditioned on the variational parameters. μ' and σ' are based on the slide label, for selecting between $\{\mu_0,\sigma_0\}$ and $\{\mu_1,\sigma_1\}$.

E CONVOLUTIONAL CRF

Following the notation in Sec. 3.3, the pair-wise potential function is defined as

$$\psi_p(\bar{\boldsymbol{a}}_k, \bar{\boldsymbol{a}}_j|\boldsymbol{m}) = c(\bar{\boldsymbol{a}}_k,\bar{\boldsymbol{a}}_j)\sum_{l=1}^{L}\gamma^{(l)}\mathcal{K}^{(l)}(\boldsymbol{m}_i,\boldsymbol{m}_j) \tag{29}$$

the mean-field approximation Q to conditional random field (Zheng et al., 2015) can be written as

$Q_k \leftarrow \frac{1}{Z}e^{\psi_u(\bar{\boldsymbol{a}}_k|\boldsymbol{m})}$
while not converged **do**
1. $Q_k \leftarrow \sum_{k\neq j}\psi_p(\bar{\boldsymbol{a}}_k,\bar{\boldsymbol{a}}_j|\boldsymbol{m})$
2. $Q_k \leftarrow \psi_u(\boldsymbol{a}_k|\boldsymbol{m}) + Q_k$
3. $Q_k \leftarrow \mathrm{normalize}(Q_k)$

end while

We omit the label compatibility step as $a_k \in R$ is a continuous value instead of categorical variable. The last normalization step could also be omitted as we perform normalization on the output attention in (3). Thus, the cross-patch label compatibility $c(\bar{\boldsymbol{a}}_k,\bar{\boldsymbol{a}}_j)$ can also be a constant, which we set to be 1. $w^{(l)}$ is a trainable parameter and $\mathcal{K}^{(l)}$ is a Gaussian kernel, both indexed by l. L is the total number of kernels. We perform this algorithm in a online fashion thus the loop is only executed once in every training iteration.

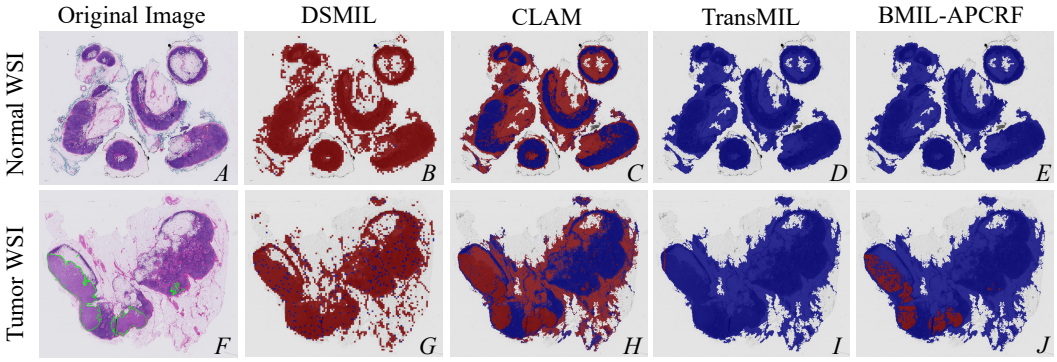


Figure 9: The segmented results using threshold=0.4. Red regions are positive and blue regions are negative.

As shown in Teichmann & Cipolla (2018), if we only consider local dependency, the kernel matrix could be defined as

$$\mathcal{K}(d\omega, d\eta, \omega, \eta) = exp(-\frac{|\vartheta(\omega,\eta) - \vartheta(\omega - d\omega, \eta - d\eta)|^2}{2\beta_k^2}), \tag{30}$$

where β_i is a learnable parameter. ϑ is the feature used for the kernel. A merged kernel matrix could be defined as $\kappa = \sum_l \gamma_l \mathcal{K}_k$. The message passing step (1) could be written as $Q(\omega,\eta) = \sum_{d\omega,d\eta} \mathcal{K}(d\omega, d\eta, \omega, \eta) \cdot Q(\omega + d\omega, \eta + d\eta)$. This is equivalent to a convolution operation (Teichmann & Cipolla, 2018). Thus, the step is simplified into the following function $\bar{\boldsymbol{a}} = C_{\boldsymbol{w},\boldsymbol{h}}(\boldsymbol{a})$. Specifically, $\tilde{\boldsymbol{a}} = \mathrm{softmax}(\boldsymbol{a})$, $\hat{\boldsymbol{a}} = \mathrm{reshape}(\boldsymbol{w}, \boldsymbol{h}, \tilde{\boldsymbol{a}})$, $\bar{\boldsymbol{a}} = \mathrm{convolution}(\hat{\boldsymbol{a}}, \mathcal{K})$. In this paper, we further take a non-parametric 3×3 convolutional kernel over the input, which is a Gaussian smoothing operation over the attention.

Next we use first-order Taylor expansion to further simplify the expectation in (8). If we perform the Taylor-expansion at the expectation of $\mathbf{a}$:

$$\mathbb{E}[\bar{\boldsymbol{a}}] = \mathbb{E}_{q(\boldsymbol{a}|\boldsymbol{\mu},\boldsymbol{\sigma})}[C_{\boldsymbol{w},\boldsymbol{h}}(\boldsymbol{a})] \approx \mathbb{E}C_{\boldsymbol{w},\boldsymbol{h}}(\mathbb{E}[\mathbf{a}]) + \mathbb{E}C'_{\boldsymbol{w},\boldsymbol{h}}(\mathbb{E}[\mathbf{a}])(\mathbf{a} - \mathbb{E}[\mathbf{a}]), \tag{31}$$

where the second term could be cancelled as $C'_{\boldsymbol{w},\boldsymbol{h}}$ is deterministic. Thus, the expectation is pushed to the mean $\mathbb{E}[\mathbf{a}] = \boldsymbol{\mu}$. The posterior form in (9) is obtained.

F Related works on uncertainty in neural network

The mutual information based uncertainty decomposition (4) is suggested by Houlsby et al. (2011). This tractable view of uncertainty is broadly used in active learning, uncertainty-critic computation vision tasks (Gal et al.; 2017b; Kendall & Gal, 2017). The followup works study how to extract distributional uncertainty, originated from the mismatch between training and testing data (Malinin & Gales, 2018). The fast approximation of expressive predictive uncertainty is studies in by Malinin et al. (2019); Cui et al. (2019). Note that the extraction of distributional uncertainty could be interesting extension to Bayes-MIL.

G More visualization results

The segmentation of Fig. 2 using threshold=0.4 is shown in Fig. 9.

More visualization results are shown in Fig. 10.

H Limitations

A potential limitation is the fixed feature extraction network - normally ResNet50, which is not extensively trained on the WSI data. When dealing with larger scale datasets, the features for positive patches and negative patches may not be separable by the MIL framework. Under such case, Bayes-MIL might provide marginal improvement. Future works will explore pretrained foundation models for better feature extraction.

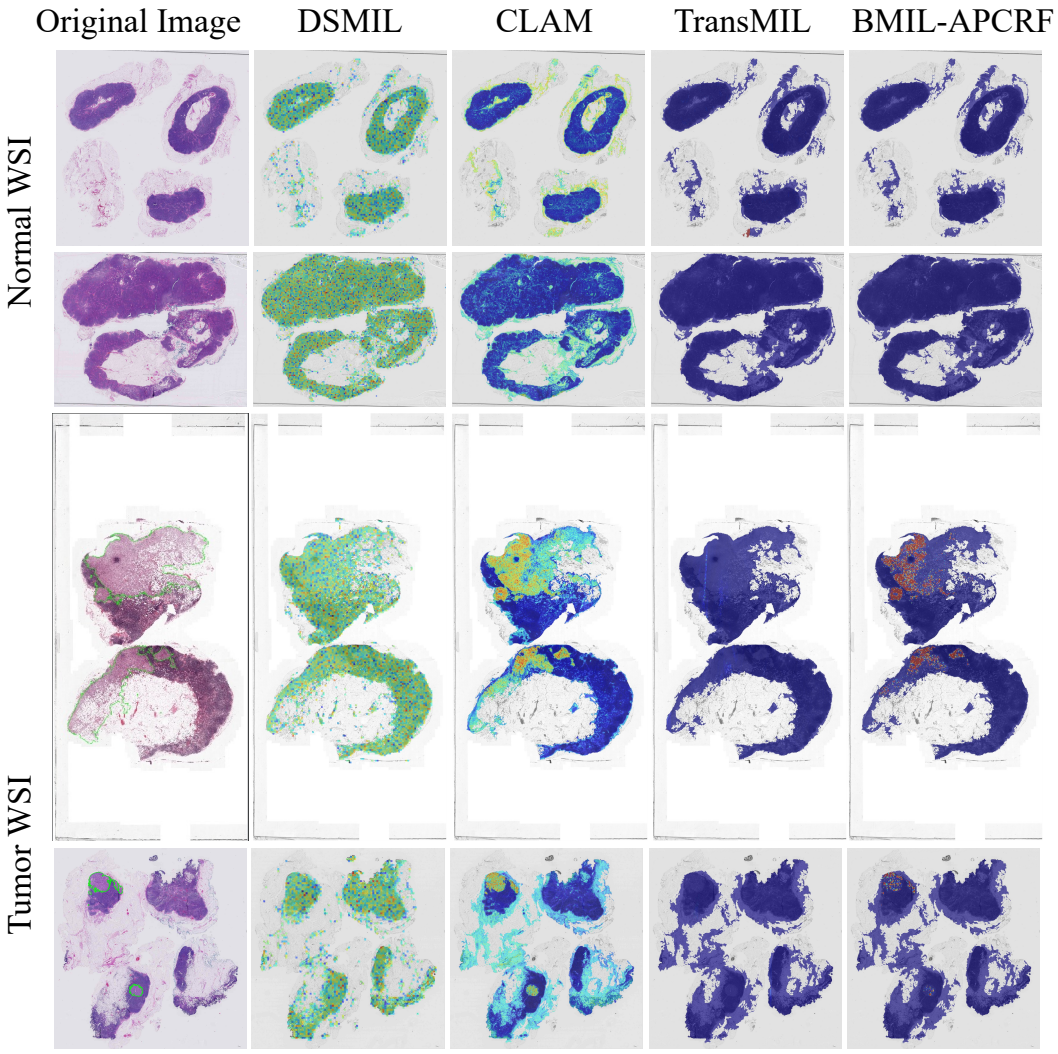


Figure 10: The visualization of normal and tumor slides and the ROIs provided by different models. The patch-level annotations for the tumor image are shown in green color in (F). The attention values $\boldsymbol{a}$ are normalized to the same range by $\frac{\boldsymbol{a}-\min_{\boldsymbol{a}}}{\max_{\boldsymbol{a}}-\min_{\boldsymbol{a}}}$. The $\min_{\boldsymbol{a}}$ and $\max_{\boldsymbol{a}}$ are the same for all methods for better visualization.

As noted in Sec. B, the implementation of APCRF sacrifices performance for efficiency. Future works include implementation of convolution for irregular feature maps for both efficiency and performance.