Appendices

A. Pseudo code for UDLS

Algorithm 1 UDLS

```
Require: MIL classifier $p(\cdot)$, PatchDropout rate $r$, label smoothing factor $\alpha$, Training epochs of Stage 1 and 2: $\{E_1, E_2\}$
Input: A train set of feature embedding bags $F = \{F_1, \cdots, F_N\}$, where $F_i = \{f_{i,1}, \cdots, f_{i,K}\}$, $F_i \in R^{K \times D}$ and $f_{i,j} \in R^{1 \times D}$ is the $j$-th instance feature in $i$-th bag
Output: Trained calibration MIL classifier and Predicted labels $\{\hat{p}(F_i) | 1 \leq i \leq N\}$

  for Bag index $i = 1$ to $N$ do
      for Dropout index $t = 1$ to $T$ do
          $F_{i,t} \leftarrow PatchFeatureDropout(F_i)$
      end for
  end for
  for epoch = 1 to $E_1$ do
      for Bag index $i = 1$ to $N$ do
          $\mathcal{L}_1^i \leftarrow \sum_{t=1}^{T} \mathcal{L}_1(y(X_i) \mid p(F_{i,t}))$
          $BackpropagateAndOptimize(p(\cdot) \mid \mathcal{L}_1^i)$
      end for
  end for
  for Bag index $i = 1$ to $N$ do
      $p_T(F_i) \leftarrow \frac{1}{T} \sum_{t=1}^{T} p(F_{i,t}), \quad p_T(F_i) \in R^{1 \times C}$
      $H(p_T(F_i)) \leftarrow -\sum_{c=1}^{C} p_T(F_i)[c] \log(p_T(F_i)[c])$
      $LS(X_i;\alpha) \leftarrow \frac{(H(p_T(F_i)) - H_{min})}{H_{max} - H_{min} + \epsilon} \cdot \alpha$, $LS(X_i;\alpha) \in R^{1 \times 1}$
      $y_c^{LS}(X_i) \leftarrow y_c(X_i)(1 - LS(X_i;\alpha)) + LS(X_i;\alpha)/C$
  end for
  for epoch = 1 to $E_2$ do
      for Bag index $i = 1$ to $N$ do
          $\mathcal{L}_2^i \leftarrow \mathcal{L}_2(y_c^{LS}(X_i) \mid p(F_i))$
          $BackpropagateAndOptimize(p(\cdot) \mid \mathcal{L}_2^i)$
      end for
  end for
```

B. Source code for UDLS

Our code can be found at: `https://github.com/parkhyeongminn/UDLS`

C. Qualitative Results on Various Models

Fig. 1, Fig. 2, and Fig. 3 present the confidence histograms and reliability diagrams that compare the calibration results of the backbone MIL models without calibration, with label smoothing, and with UDLS on the Camelyon16 image classification. The top rows show the histograms of the predicted confidence, where the dashed black lines indicate the average accuracy, and the dashed gray lines indicate the average confidence. The bottom rows show the reliability diagrams which plot the average bin accuracy against the average bin confidence of the positive labels.

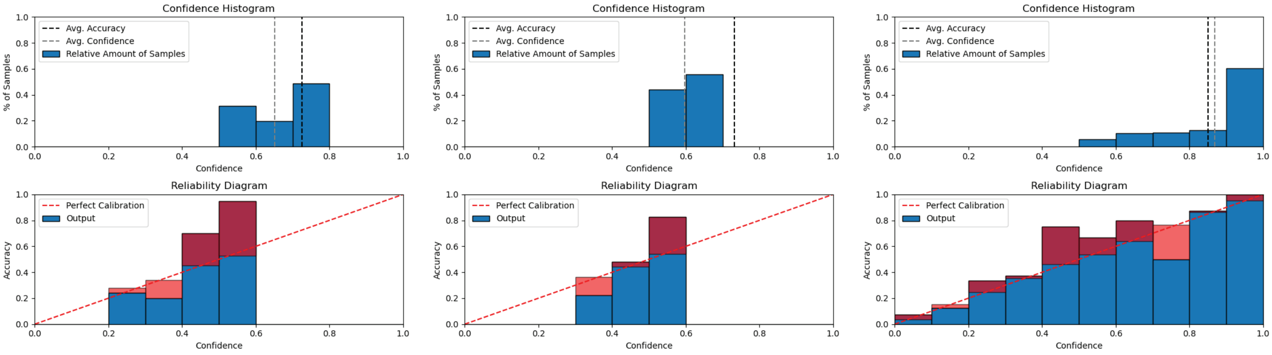
From Fig. 1, the confidence histograms and the reliability diagrams indicate that the AB-MIL is under-confident, and it becomes even more under-confident with label smoothing, with no sample possessing a predicted probability over 0.7. On the other hand, the under-confidence problem of AB-MIL is resolved using UDLS, being almost perfectly calibrated. In addition, it shows a notable improvement in classification accuracy at the same time. It suggests that while the original label smoothing makes samples harder for under-confident models, the proposed uncertainty-based data-wise label smoothing is more effective for identifying easy samples.

Fig. 2 and Fig. 3 show that Trans-MIL and DTFD-MIL are over-confident, and both the original label smoothing and the proposed uncertainty-based data-wise label smoothing are effective for calibrating the over-confident models. After calibration, the gaps between the average accuracy and the average confidence in the confidence histograms are reduced, and the samples appear in the bins of the intermediate positive confidence values of the reliability diagrams, which indicates that both the label smoothing and UDLS are capable of identifying hard samples from over-confident models.

Notably, the confidence histograms indicate that the average accuracy of the label smoothing is always lower than the average accuracy of the UDLS, indicating that UDLS is effective in both calibration and classification capability. Further analysis is explained in 4.4.

D. Sensitivity Analysis

We conducted a sensitivity analysis to analyze the main hyper-parameters of the UDLS, PatchDropout rate r, and the global label smoothing factor α. We conducted experiments by varying the hyper-parameters.

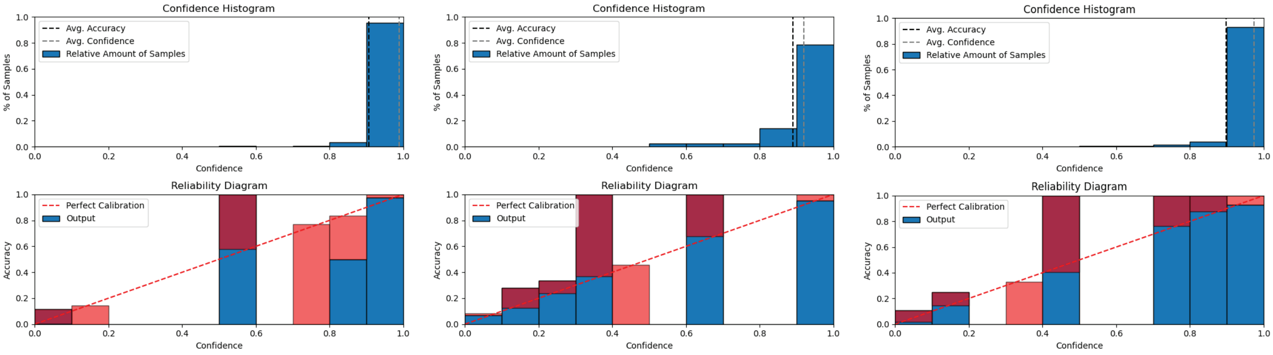


(a) Without Calibration

(b) Label Smoothing

(c) UDLS

Figure 1. The confidence histograms and the reliability diagrams of AB-MIL on Camelyon16

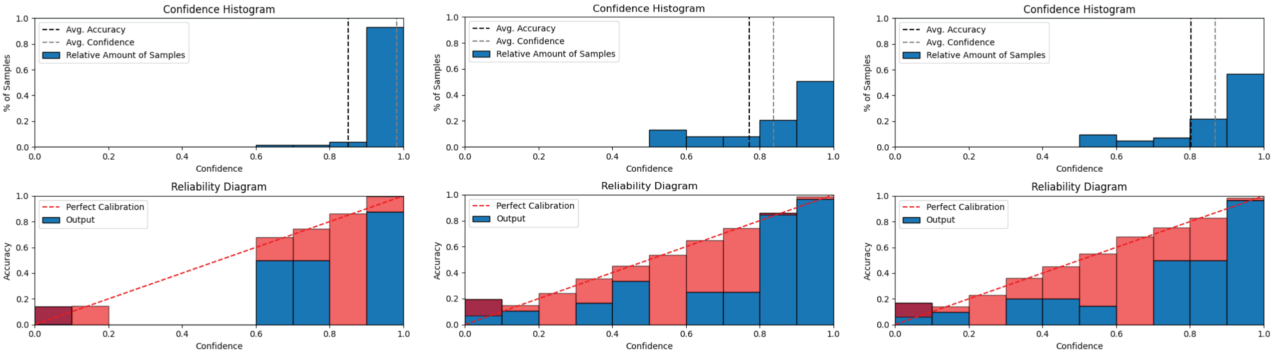


(a) without calibration

(b) Label Smoothing

(c) UDLS

Figure 2. The confidence histograms and the reliability diagrams of Trans-MIL on Camelyon16



(a) without calibration

(b) Label Smoothing

(c) UDLS

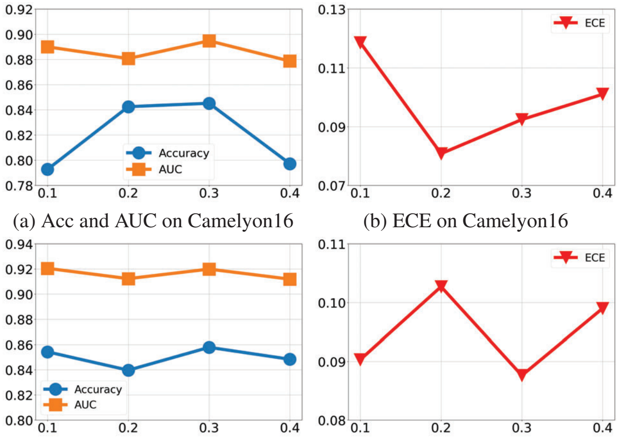
Figure 3. The confidence histograms and the reliability diagrams of DTFD-MIL on Camelyon16

Tab. 1 shows the results on different values of the label smoothing factor α. Since the original label smoothing is usually implemented with a factor of 0.05 or 0.1, we used the same values. Results demonstrate that AB-MIL and Trans-MIL show better performance when $\alpha = 0.05$ while DTFD-MIL performs better when $\alpha = 0.1$.

Since there is no rule of thumb regarding the PatchDropout rate r, the experiments were performed with the variations in this hyper-parameter. The average accuracy, AUC, and ECE values were computed from the classification results across the 3 MIL frameworks on Camelyon16 and TCGA, respectively.

	AB-MIL [1]						**Trans-MIL** [2]						**DTFD-MIL** [4]					
	Camelyon16			**TCGA**			**Camelyon16**			**TCGA**			**Camelyon16**			**TCGA**		
Alpha	Acc ↑	AUC ↑	ECE ↓	Acc ↑	AUC ↑	ECE ↓	Acc ↑	AUC ↑	ECE ↓	Acc ↑	AUC ↑	ECE ↓	Acc ↑	AUC ↑	ECE ↓	Acc ↑	AUC ↑	ECE ↓
0.05	0.842	**0.927**	0.078	0.859	0.895	**0.084**	0.866	**0.924**	0.121	0.863	**0.947**	0.095	**0.803**	**0.857**	**0.115**	**0.824**	**0.920**	0.076
0.1	**0.850**	0.922	**0.060**	**0.863**	**0.904**	0.093	**0.898**	0.914	**0.064**	**0.886**	0.945	**0.082**	0.695	0.762	0.152	0.736	0.812	**0.057**

Table 1. Sensitivity analysis on the impact of the label smoothing factor α on classification results across MIL models



(a) Acc and AUC on Camelyon16

(b) ECE on Camelyon16

(c) Acc and AUC on TCGA

(d) ECE on TCGA

Figure 4. Sensitivity analysis on the impact of the PatchDropout rate. The x-axis represents the PatchDropout rate. The y-axis in (a) and (c) are the average accuracy and AUC from the classification results of the 3 MIL models on Camelyon16 and TCGA, respectively, and the y-axis in (b) and (d) are the average ECE from the classification results of the 3 MIL models on Camelyon16 and TCGA, respectively.

Results in the Fig. 4 indicate that the optimal PatchDropout rate differs slightly between the two datasets. The average accuracy and AUC do not vary greatly depending on the PatchDropout rate, and they show the highest values when the PatchDropout rate is 0.3 on both datasets.

The best-performing PatchDropout rate on calibration in terms of ECE is 0.2 for Camelyon16 and 0.3 for TCGA. Since the TCGA dataset contains more positive instances in the positive bags than the Camelyon16, a higher PatchDropout rate is required to generate enough variations from the multiple predictions of a single input WSI to further extract the predictive entropy estimates for smoothing factor, resulting in a better-calibrated model.

E. Implementation Details for Different Data Augmentations

We provide the implementation details of the experiments with different data augmentations presented in 4.5.1. All augmentation methods were conducted 10 times for each input data.

The Gaussian noise method is implemented by selecting random noise from a Gaussian distribution with a mean of zero and a standard deviation calculated across instance features in a bag. Then, the noise is scaled to 0.1 and added to patch feature c_j.

$$\bar{c_j} = c_j + \gamma X, \quad X \sim \mathcal{N}(0, \sigma_i^2), \ \gamma = 0.1 \tag{1}$$

The ReMix [3] consists of two steps: reduce and mix. First, it reduces the number of instances in WSI bags by substituting instances with patch cluster centroids. Then, a ”Mix-the-bag” augmentation includes four online, stochastic, and flexible latent space augmentations: Append, Replace, Interpolate, and Covary. Among these, Covary-augmentation was chosen for its superior performance in the original paper. Covary-augmentation creates a new representation from the key covariance matrix by

$$\bar{c_j} = c_j + \lambda \cdot \delta, \quad \delta \sim \mathcal{N}(0, \Sigma_{j^*}^k) \tag{2}$$

where λ is a strength hyper-parameter and $\Sigma_{j^*}^k$ is the covariance matrix corresponding to the closest patch center centroid $c_{j^*}^k$. Following the implementation of the original work, we set the number of cluster centroids as $K = 8$, and the augmentation probability as $p = 0.5$, and uniformly sample λ from the range $(0, 1)$ in each augmentation.

References

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

[2] 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.

[3] 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 *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 35–45. Springer, 2022.

[4] 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.