Supplementary Material

1. Derivation of Variational Information Bottleneck with Bernoulli Prior

Variational Information Bottleneck [1]

The Information Bottleneck (IB) can work as an information compression role to intervene in DNN's training [1]. Consider the joint distribution $p(X, Y, Z)$ factors as follows:

$$\begin{aligned} p(X, Y, Z) &= p(Z|X, Y)p(Y|X)p(X) \\ &= p(Z|X)p(Y|X)p(X), \end{aligned} \tag{1}$$

and assume $p(Z|X, Y) = p(Z|X)$, corresponding to the Markov chain $Y \leftrightarrow X \leftrightarrow Z$. The objective function of IB to be maximized is given in [7] as,

$$R_{IB} = I(Z, Y) - \beta I(Z, X), \tag{2}$$

where $I(\cdot, \cdot)$ indicates the Mutual Information (MI) and β is a Lagrange multiplier.

Since the computation of MI is intractable during the training of the neural networks, the variational bound of the two term can be derived as:

$$\begin{aligned} & I(Z, Y) \\ &= \int dydzp(y, z) \log \frac{p(y|z)}{p(y)} \\ &= \int dydzp(y, z) \log \frac{p(y|z)q(y|z)}{p(y)q(y|z)} \\ &= \int dydzp(y, z)\{\log q(y|z) - \log p(y) + \log \frac{p(y|z)}{q(y|z)}\} \\ &= \int dydzp(y, z) \log q(y|z) + H(Y) \\ &\quad + KL(p(Y|Z), q(Y|Z)) \\ &>= \int dydzp(y, z) \log q(y|z) \\ &= \int dxdydzp(x)p(y|x)p(z|x) \log q(y|z), \end{aligned} \tag{3}$$

$$\begin{aligned} I(Z, X) &= \int dzdxp(x, z) \log \frac{p(z|x)}{p(z)} \\ &= \int dzdxp(x, z) \log \frac{p(z|x)r(z)}{p(z)r(z)} \\ &= \int dzdxp(x, z) \log \frac{p(z|x)}{r(z)} - KL(p(Z), r(Z)) \\ &<= \int dzdxp(x, z) \log \frac{p(z|x)}{r(z)} \\ &= \int dzdxp(x)p(z|x) \log \frac{p(z|x)}{r(z)}, \end{aligned} \tag{4}$$

Thus, the IB objective can be transferred as a variational bound of Eq.(2) as follows:

$$\begin{aligned} R_{IB} >= & \int dxdydzp(x)p(y|x)p(z|x) \log q(y|z) \\ & - \beta \int dzdxp(x)p(z|x) \log \frac{p(z|x)}{r(z)} \\ = & -\frac{1}{N} \sum_{n=0}^{N} \mathbb{E}_{z \sim p_\theta(z|x_n)}[-\log q_\phi(y_n|z)] - \\ & \beta KL[p_\theta(z|x_n), r(z)], \end{aligned} \tag{5}$$

Where the $p(x)p(y|x)$ is approximated by using the empirical data distribution during stocastic batch iteration training, N denotes the number of samples, $q_\phi(y|z)$ is a parametric approximation to the likelihood $p(y|z)$, $r(z)$ is the prior probability of z to variational approximate the marginal $p(z)$, and $p_\theta(z|x)$ is the parametric posterior distribution over z. Then, to maximize IB objective can be seen to minimize:

$$\begin{aligned} J_{IB} = \frac{1}{N} \sum_{n=0}^{N} & \mathbb{E}_{z \sim p_\theta(z|x_n)}[-\log q_\phi(y_n|z)] + \\ & \beta KL[p_\theta(z|x_n), r(z)]. \end{aligned} \tag{6}$$

Learn Sparsity via Variational Bound of IB [5]

To trade off the dilemma of computational limitation and task-specific representation learning via end-to-end back-propagation, we propose to utilize the IB module to filter most task-irrelevant instances for task-specific fine-tuning.

The above filtering process can be implemented by optimizing the second term of in Eq.(2) which controls the compression. There are two ways that compress X to Z by decreasing the KL divergence between $p(z|x)$ and $r(z)$ in Eq.(6) variational method: reducing the dimension of representation Z compared to X in [1], or converting input X into a sparse one in [5].

For the setting of our long instance sequenced MIL, we reduce $I(X, Z)$ into a degree so that the gradients can be back-propagated to the backbone encoder, which needs us to convert a WSI of bag size over 10k into 1k for the sake of sparsity. Considering MIL for tumor v.s. normal binary classification without loss of generality and the latent label y_i of each instance x_i, we argue that it is sufficient enough to make the WSI level prediction if one tumor area is detected. With the above understanding, we propose to learn compressed components similar to [5] by defining a IB module as:

$$z = m \odot x, \tag{7}$$

where m is a Bernoulli(π) distributed binary mask, thus $r(z|x) = (1-\pi)\delta(z) + \pi\delta(z-x)$. and in this way $KL[p_\theta(z|x), r(z)]$ in Eq.(6) can be decomposed as,

$$\begin{aligned} &KL[p_\theta(z|x), r(z)] \\ &= (1-\theta(x)) \int \delta(z) log \frac{p_\theta(z|x)}{r(z)} dz \\ &+ \theta(x) \int \delta(z-x) log \frac{p_\theta(z|x)}{r(z)} dz \\ &= (1-\theta(x)) log \frac{1-\theta(x)}{1-\pi} + \theta(x) log \frac{\theta(x)}{\pi p(x)} \\ &= KL[p_\theta(m|x), r(m)] - \theta(x) log p(X) \\ &= KL[p_\theta(m|x), r(m)] + \pi H(X), \end{aligned} \tag{8}$$

where $H(X)$ is the entropy of X, which can be omitted during the minimization due to its constant value.

2. PyTorch Pseudocode

We show the pytorch pseudocode of the WSI sparsity training of stage-1 in Algorithm 1.

Algorithm 1: PyTorch-style pseudocode for WSI task-specific IB sparsity learning

```
# Learn sparsity of WSI with fixed
 backbone
for (X,y) in data_loader:
   with torch.no_grad():
      model.eval()
      Z_0 = model(X)
      # X = x_1,x_2,...,x_n
      # Z = z_1,z_2,...,z_n
   model.train()
   # IB is a sequential FCs
   M = IB(Z_0)
   logits = torch.sigmoid(M)
   p_z = Bernoulli(logits)
   Z_mask = p_z.sample()
   r_z = Bernoulli(π)
   # reparameterization trick for
    Bernoulli samples
   Z_1 = Z_0·(M+Z_mask)/2
   Y = model_wsi(Z_1)
   loss1 = CrossEntropyLoss(Y,y)
   loss2 = KL_divergence(p_z, r_z)
   loss = loss1+βloss2
   optimizer.zero_grad()
   loss.backward()
   optimizer.step()
```

3. Details of Datasets

Camelyon-16 [2] is a public dataset for metastasis detection in breast cancer (tumor / normal classification), including 270 training sets and 130 test sets. A total of about 1.5 million patches at ×20 magnification are obtained after preprocess.

TCGA-BRCA The Cancer Genome Atlas Breast Cancer [6] is a public dataset for breast invasive carcinoma cohort for Invasive Ductal Carcinoma (IDC) versus Invasive Lobular Carcinoma (ILC) subtyping. The WSI is segmented into non-overlapping tissue-containing patches at 20× magnification and about 2.0 million patches were curated from 1038 WSIs.

LBP-CECA The Liquid-based Preparation cytology for Cervical Cancer's early lesion screening dataset is introduced to validate the universality of our method on cytopathology. The WSIs include 4 classes (Negative, ASC-US, LSIL, ASC-H/HSIL [4]) and are segmented into patches with overlapping of 25 and size of 256 at 20× magnification and about 3.2 million patches were curated from 1393 WSIs.

Camelyon-16-C is generated with random synthetic domain shift on Camelyon-16 [2] testset for simulation. Three kind of corruptions are included: Jpeg compression, Brightness and Hue are implemented by the code in [8], all with a severity of 2.

Camelyon-17 [3] dataset is collected from five different centers. It is an offical extension challenge of Camelyon-16. In this paper we combine all tumor positive WSI and random selected negative to constitude a real domain shift test set. Finally, 164 WSIs are sampled out for test.

4. Further Ablation Experiments

Influence of Learning Rate on the Backbone

Here we show the influence of backbone learning rate on Top-512 fine-tuning results, which is performed on Camelyon-16 only once for the relatively long training time of training stage-2. The ablations results are summarized in Table 1. Since the supervision signal of WSI is too weak, we find that lower learning rate helps convergence. For learning rate of 1e-3 and 5e-4, the fine-tuning collapse quickly and diverges to Nan loss. For learning rate of 1e-5, we get the best fine-tuning results on Top-512 as a WSI distilled bag.

LR	F1	AUC
1e-3	N/A	N/A
5e-4	N/A	N/A
1e-4	0.682	0.744
5e-5	0.713	0.741
1e-5	**0.899**	**0.944**
5e-6	0.876	0.908
1e-6	0.806	0.804

Table 1. **Influence of Learning Rate on the Backbone** during fine-tuning process with weakly WSI supervision.

Number selection of Top-K

Here we show the influence of IB module training in stage-1, which is performed on Camelyon-16 with five runs. The ablations results are summarized in Table 2. Generally, with the increasement of K, less essential instances would be neglected, resulting in better performace. However, most of WSIs in the Camelyon-16 dataset are with only a few tumor area, thus the less Top-K somehow fit better this dataset property. So we find that top-2048 shows the best results and even higher than all instances used for WSI decision. However for the computational limitation, we finally select top-512 for fine-tuning of stage-2.

Top-K	F1	AUC
128	0.840±0.011	0.870±0.010
256	0.843±0.009	0.870±0.010
512	0.843±0.005	0.866±0.011
1024	0.845±0.007	0.864±0.011
2048	**0.846±0.004**	**0.875±0.010**
all	0.839±0.018	0.875±0.028

Table 2. **Number selection of Top-K.**

Value selection of Lagrange multiplier

Here we show the influence of Lagrange multiplier during training stage-1, which is performed on Camelyon-16 with five runs. Definitely, the Lagrange multiplier β works as a trade off factor of the two task: if we care more about WSI training loss with a low β, then the ranking or sparsity properties of IB module may not be well learned. On the contrary, a large β will influence the training of WSI classifier. The ablations results are summarized in Table 3 and we find that the best selection of β is 1e-1.

β	F1	AUC
Upper bound	0.839±0.018	0.875±0.028
1e-3	0.835±0.008	0.860±0.012
1e-2	0.833±0.006	0.860±0.028
1e-1	**0.849±0.010**	**0.865±0.014**
1	0.839±0.015	0.852±0.018
10	0.838±0.016	0.862±0.020
100	0.828±0.010	0.853±0.007

Table 3. **Value selection of Lagrange multiplier.**

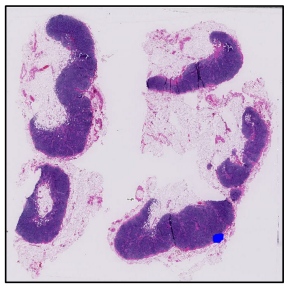
5. Result Analysis of the 3 Stages

There is a probability that the top-K instances may not contain at least one tumor patch for extreme cases, e.g. some Camelyon-16 WSIs contain very few tumors in Fig.2. Thus stage-3 is needed for covering all instances to get WSI result equipped with fine-tuned backbone, which shows further improvement compared to stage-2 in Fig.1. We also show that with random k instances, the model in stage-2 cannot converge, in Fig.1.

Figure 1. Performance of three stages on Camelyon-16, most can be found from the prior submission material.

Method	AUC
CLAM-SB	0.875
stage-1	0.865
stage-2	0.944
stage-3	0.956
stage-2 random	0.731

Figure 2. A WSI with very few tumor areas (blue).



References

[1] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017. 1, 2

[2] 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. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*, 318(22):2199–2210, 2017. 2

[3] Geert Litjens, Peter Bandi, Babak Ehteshami Bejnordi, Oscar Geessink, Maschenka Balkenhol, Peter Bult, Altuna Halilovic, Meyke Hermsen, Rob van de Loo, Rob Vogels, et al. 1399 h&e-stained sentinel lymph node sections of breast cancer patients: the camelyon dataset. *GigaScience*, 7(6):giy065, 2018. 2

[4] Ritu Nayar and David C Wilbur. *The Bethesda system for reporting cervical cytology: definitions, criteria, and explanatory notes*. Springer, 2015. 2

[5] Bhargavi Paranjape, Mandar Joshi, John Thickstun, Hannaneh Hajishirzi, and Luke Zettlemoyer. An information bottleneck approach for controlling conciseness in rationale extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1938–1952, Online, Nov. 2020. Association for Computational Linguistics. 1, 2

[6] Nicholas A Petrick, Shazia Akbar, Kenny HH Cha, Sharon Nofech-Mozes, Berkman Sahiner, Marios A Gavrielides, Jayashree Kalpathy-Cramer, Karen Drukker, Anne LL Martel, et al. Spie-aapm-nci breastpathq challenge: an image analysis challenge for quantitative tumor cellularity assessment in breast cancer histology images following neoadjuvant treatment. *Journal of Medical Imaging*, 8(3):034501, 2021. 2

[7] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000. 1

[8] Yunlong Zhang, Yuxuan Sun, Honglin Li, Sunyi Zheng, Chenglu Zhu, and Lin Yang. Benchmarking the robustness of deep neural networks to common corruptions in digital pathology. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 242–252. Springer, 2022. 2