Investigation of Training Multiple Instance Learning with Instance Sampling

Aliasghar Tarkhan[1] ATARKHAN@UW.EDU
Trung Kien Nguyen[2] NGUYENK8@GENE.COM
Noah Simon[1] NRSIMON@UW.EDU
Jian Dai[2] DAIJ12@GENE.COM
[1] *University of Washington, Department of Bioststistics, Seattle, USA*
[2] *PHC Imaging Group, Genenetch, South San Francisco, USA*

Editors: Under Review for MIDL 2022

Abstract

One challenge of training deep neural networks with gigapixel whole-slide images (WSIs) in computational pathology is the lack of annotation at pixel level or region (instance) level due to the high cost and time-consuming labeling effort. Multiple instance learning (MIL) as a typical weakly supervised learning method aimed to resolve this challenge by using only the slide-level label without the need for pixel or region labels. Not all instances are predictive of the outcome. The attention-based MIL method leverages this fact to enhance the performance by weighting the instances based on their contribution in predicting the outcome. A WSI typically contains hundreds of thousands of image regions. Training a deep neural network with thousands of image regions (patches) per slide is computationally expensive, and it needs a lot of time for convergence. One way to alleviate this issue is to sample a subset of instances from the available instances within each bag for training. While the benefit of sampling strategies for decreasing computing time might be evident, there is a lack of effort to investigate their performances. This paper investigates different sampling strategies from both computing time and performance points of view. We empirically show how these sampling strategies substantially reduce computation time. Moreover, we discover that random sampling can even improve performance of the attention-based MIL that uses all instances if we randomly choose enough number of instances.

Keywords: Attention, computational pathology, deep learning, multiple instance learning, prostate cancer, sampling, transfer learning, weekly supervised learning

1. Introduction

Thanks to the advancements in digital pathology, especially slide scanners, visual inspection of sampled tissues through high-resolution Whole-Slide Images (WSIs) from biopsies has become the gold standard for diagnosing many diseases in oncology, such as prostate cancer (Fraggetta et al., 2017; Epstein, 2010; Otálora et al., 2021). However, the manual inspection of the entire WSI (with a typical size $10^5 \times 10^5$ pixels) is costly and time-consuming to be done by an expert. Also, the diagnosis might differ from one expert to another, known as the observer variability (Brunyé et al., 2010).

Computational pathology aims to develop automated machine learning and artificial intelligence tools to analyze the gigapixel WSIs (Cui and Zhang, 2021). Such tools save cost and time; they also showed great accuracy and provided high-quality health care to patients with different diseases (Tarkhan et al., 2021; Nofallah et al., 2021; Molani et al., 2019). However, developing such automated tools for WSIs comes with some new challenges, especially when using complex models such as deep neural networks. WSIs are giga-pixel images, and they are too big to be fed into a deep neural network due to the memory constraint. One immediate solution is to divide a WSI into many (typically hundreds of thousands) smaller regions (with the typical size of 256×256 or 512×512), also known as patches or tiles. One can train a deep neural network by feeding a single or very few numbers of these small images. However, the main challenge is the lack of pixel-level annotation and that the labels (i.e., showing the status of disease) are only available at the slide (patient) level. A possible solution might be annotating those smaller image regions (or patches). Labeling such images by an expert at the pixel level (or in smaller image patches) is costly (labor and time) (Quellec et al., 2017).

Multiple instance learning (MIL), as a typical weakly supervised learning method, has been proposed to tackle this challenge (Dietterich et al., 1997; Maron and Lozano-Pérez, 1998). In a MIL problem, the aim is to train a model with bags of instances where the algorithm can only access the labels at the bag level. Such a scenario often happens in pathology, where one usually divides a gigapixel WSI image into many smaller image regions, known as tiles or patches. For prostate cancer, for example, each image tile can be partially related to a sub-type (the bag label), but it may not represent it by itself. Therefore, the upcoming challenge with the MIL problem is that not all instances (image tiles) are equally predictive of the bag label (class), and some of them may even relate to the other classes (Liu et al., 2012).

Some works considered combining the instance-level responses from a classifier to alleviate this challenge (Bahdanau et al., 2016; Raffel and Ellis, 2016; Ramon and Raedt, 2000; Raykar et al., 2008; Ilse et al., 2018). Among them, (Ilse et al., 2018) proposed an attention-based deep MIL framework to deal with this challenge. Their proposed framework includes two networks : (1) attention network and (2) classification network. These two networks are trained simultaneously. The attention network has parameters for updating the attention (importance) weights of different instances, while the classification network has parameters for the classification task. Although their approach increases flexibility and interpretability of MIL problems, it still has a challenge: They use all instances per bag across all iterates when training the combined network. A WSI has hundreds of thousands of image tiles (e.g., with size 256 × 256). Feeding all of these instances, regardless of their predictive information for the class label, is time-consuming and computationally expensive. An attention MIL network may not need to be trained by instances that are just noises or have little information for the class label.

This paper investigates different sampling strategies for the attention-based deep MIL framework. We consider four sampling strategies: (1) no sampling, (2) random sampling, (3) adaptive sampling, and (4) top-k sampling. We show how the sampling strategies substantially reduces computation time. Among them, we also show that random sampling strategy can improve performance compared to no sampling (i.e., using whole instances in the original work Ilse et al. (2018)) if we choose enough number of selected instances. We use the Cancer Genome Atlas (TCGA) repository of prostate adenocarcinoma (TCGA-PRAD) dataset (Zuley et al., 2016) and Camelyon16 to compare different strategies and support our discoveries.

2. MIL and Attention-based MIL Networks

In this section, we briefly explain MIL problem and its attention-based version.

2.1. MIL problem formulation

Suppose there are N subjects (or patients) with bags of images $\boldsymbol{\mathcal{X}}^{(1)}, \boldsymbol{\mathcal{X}}^{(2)}, \ldots, \boldsymbol{\mathcal{X}}^{(N)}$ and bag-level binary labels $y^{(1)}$, $y^{(2)}$, ..., $y^{(N)} \in \{0, 1, \ldots, C-1\}$ where C is number of classes. The bag for n-*th* patient (i.e., $\boldsymbol{\mathcal{X}}^{(n)}$) contains K_n instance images $\boldsymbol{X}_1^{(n)}, \boldsymbol{X}_2^{(n)}, \ldots, \boldsymbol{X}_{K_n}^{(n)}$. For instance, in computational pathology, nne can obtain such K_n instance images by sampling from different regions of a WSI (i.e., $\boldsymbol{\mathcal{X}}^{(n)}$). In the classical supervised learning, we have $K_n = 1$, i.e., there is one image per subject with corresponding label $y^{(n)}$. Note that the number of instances inside the bag can vary among different subjects. To decrease computing time and cost, it is common to use a state-of-the-art pre-trained network such as *ResNet50* (He et al., 2015) to extract a low-dimensional embedding feature $\boldsymbol{h}_k^{(n)}$ from k^{th} instance image of n^{th} subject, i.e., $\boldsymbol{X}_k^{(n)}$. After that, we have dataset $\{(\boldsymbol{h}_k^{(n)}, y^{(n)}, \text{ for } n = 1, 2, \ldots N \text{ and } k = 1, 2, \ldots, K_n\}$. The task of the neural network is to predict the label of n^{th} subject, i.e., $y^{(n)}$ through extracting features from its K_n embeddings $\boldsymbol{h}_k^{(n)}$, $k = 1, 2, \ldots, K_n$. In computational pathology applications (e.g., prostate cancer (Otálora et al., 2021)), the instance-level labels $y_k^{(n)}, k = 1, 2, \ldots, K_n$ are unknown and we only have the bag-level label $y^{(n)}$. The bag-level images (e.g., WSIs) are too big to feed into the neural networks due to the memory constraint. Multiple-instance learning (MIL) is a weakly supervised learning approach to train the neural networks using instances while only bag labels are available (Quellec et al., 2017). For binary classification task (i.e., $C = 2$), the basic assumption of a MIL problem is:

$$y^{(n)} = \begin{cases} 0, & \text{iff } \sum_{k=1}^{K_n} y_k^{(n)} = 0 \\ 1, & \text{otherwise.} \end{cases} \tag{1}$$

or equivalently,

$$y^{(n)} = \max_k \{y_k^{(n)}\}, \tag{2}$$

conveying that a bag is labeled positive if it contains at least a positive instance. The above two expressions are not appealing from the optimization perspective. One possible alternative is to consider element-wise maximum operator as

$$h_{bag,m}^{(n)} = \max_k \{h_{k,m}^{(n)}\}, \forall m = 1, 2, \ldots M \tag{3}$$

where $h_{k,m}^{(n)}$ is the m^{th} element of k^{th} instances for n^{th} patient; M is the dimension of embedding extracted from the pre-trained network (e.g., M=1024 when we use ResNet50 to extract features from its last fully-connected layer). Another alternative is to use the mean operator:

$$\boldsymbol{h}_{bag}^{(n)} = \frac{1}{K_n} \sum_{k=1}^{K_n} \boldsymbol{h}_k^{(n)}. \tag{4}$$

However, both operators in Equation (3) and Equation (4) are pre-calculated and predefined. Hence, they are non-trainable, which hinders assigning trainable weights to the embedding features based on their contribution in predicting the bag label.

In practice, not all embedding features (or in general instances) contribute to the prediction of outcome (i.e., $y^{(n)}$) equally. Some instances are just noises, some have little information, some others have information about another class, and only a few of them are well-predictive of the outcome. Therefore, there is a need to have a trainable framework to weigh different instances based on their underlying information about the outcome. Attention-based MIL framework proposed by (Ilse et al., 2018) deals with this challenge.

2.2. Attention-based MIL

Authors in (Ilse et al., 2018) proposed an attention-based MIL pooling approach that is trainable. They proposed a combined architecture of two trainable networks: attention network and classification network. The attention network is trained so that the weighted average of embedding features by their trainable attention weights represents the class at most. The classification network is trained to minimize the prediction error given the pooled embedding feature as its input. Both of these two network pieces are trained simultaneously. To allow for the element-wise non-linearity, (dis)similarities discovery, and a better expressiveness, the authors in (Ilse et al., 2018) proposed to use a gated attention mechanism for MIL pooling. They considered a single-branch attention mechanism where all classes share a shared attention branch. The MIL pooled (aggregated) feature is given as

$$\boldsymbol{h}_{bag}^{(n)} = \sum_{k=1}^{K_n} a_k^{(n)} \boldsymbol{h}_k^{(n)}, \tag{5}$$

with

$$a_k^{(n)} = \frac{exp\big\{\boldsymbol{w}^T \big(tanh(\boldsymbol{V}\boldsymbol{h}_k^{(n)}) \odot sigm(\boldsymbol{U}\boldsymbol{h}_k^{(n)})\big)\big\}}{\sum_{k'=1}^{K_n} exp\big\{\boldsymbol{w}\big(tanh(\boldsymbol{V}\boldsymbol{h}_{k'}^{(n)}) \odot sigm(\boldsymbol{U}\boldsymbol{h}_{k'}^{(n)}))\big)\big)\big\}}, \tag{6}$$

where $\boldsymbol{w} \in \mathbb{R}^{L\times 1}$, $\boldsymbol{U} \in \mathbb{R}^{L\times M}$, and $\boldsymbol{V} \in \mathbb{R}^{L\times M}$ are trainable parameters included in the attention network; $tanh(.)$ and $sigm(.)$ are the element-wise hyperbolic tangent and sigmoid functions; $\odot$ is an element-wise multiplication. Such a MIL pooling mechanism preserves flexibility and interpretability (see Section 2.4 in (Ilse et al., 2018)). Finally, the bag-level aggregated representation $\boldsymbol{h}_{bag}^{(n)}$ is fed into the classification network that includes C individual classification branches. Each classification branch estimates the predicted score of the corresponding class. The predicted $C \times 1$ score vector is given as

$$\boldsymbol{s}_{bag}^{(n)} = \boldsymbol{W}_c^T \boldsymbol{h}_{bag}^{(n)}, \tag{7}$$

where $\boldsymbol{W}_c \in \mathbb{R}^{M\times C}$ is the trainable classifier with c-th column corresponds to the c-th branch predicting the score of class c for the bag. Finally, one can estimate the bag label by

$$\widehat{y}^{(n)} = \arg\max_c \{\boldsymbol{s}_{bag}^{(n)}\}. \tag{8}$$

In many pathology applications, there might be many instances within each WSI which increase computing time and cost. In the next section, we present different sampling strategies to overcome these possible shortcomings.

3. Sampling Strategies for Attention-based MIL

3.1. Random sampling

With random sampling strategy, we randomly draw a limited number of instances (or images) to train the deep neural network. The main reason to use this strategy is due to memory constraint: it is not possible to bring all instances/images of a patient (bag) or a batch of patients into memory to train the deep neural network. This strategy has been used in the literature (Zhu et al., 2016; Wulczyn et al., 2020; Li et al., 2018) and showed a great success to reduce computing resources and time. However, there is a lack of investigation on the computing time and performance of random sampling in the deep attention-based MIL network. On one hand, different random subsets of instances for a patient (bag) for training the network over different iterates may increase generalizability and handle over-fitting better (Bishop, 1995). On the other hand, using a limited number of instances per iterate may not capture whole information to predict the outcome of the patient. Therefore, there it might be worth investigating such a trade-off which is one of the aims of this paper.

3.2. Adaptive sampling

In practical applications (e.g., prostate cancer (Otálora et al., 2021)), there are many instance images may not contribute to the bag (patient) class. There have been some works in the literature dealing with this issue (Williamson et al., 2021; Lu et al., 2019; Dehaene et al., 2020), but they all used whole instances. We propose to adaptively draw G well-predictive instances per subject (bag) from an empirical sampling distribution. For nth patient, we estimate the sampling distribution as a multinomial distribution with a corresponding vector of probabilities $\mathcal{P}^{(n)} = (p_1^{(n)}, p_2^{(n)}, \ldots, p_{K_n}^{(n)})$, $0 \leq p_k^{(n)} \leq 1$, $\sum_{k=1}^{K_n} p_k^{(n)} = 1$ where we choose $p_k^{(n)} = a_k^{(n)}$ (the attention weight extracted from forward attention network). We propose to draw a subset of G indices from distribution $\mathcal{P}^{(n)}$ as

$$(I_1^{(n)}, I_2^{(n)}, \ldots, I_G^{(n)}) \sim \mathcal{P}^{(n)}. \tag{9}$$

With Equation (9), instances that have higher attention weights (i.e., higher $a_k^{(n)}$ that are well-predictive of the outcome) will be chosen more often during training. After adaptively drawing the G instances over each iterate, we train the attention-based neural network by following Equation (5) to Equation (8) by replacing K_n with G. Since the estimates of the network parameters and consequently the attention weights $a_k^{(n)}$ are noisier over a couple of initial iterates (epochs), we propose to consider a few initial iterates as warm-up iterates where we use all instances to train the network. Although the estimation of instance sampling distribution using the forward attention network is faster than training the whole network, it may add overload if we do it on every iterate/epoch. Therefore, one might decide to estimate $\mathcal{P}^{(n)}$ on every e_{update} epochs. Figure A in Appendix 1 illustrates the general architecture of instance sampling strategies and Algorithm 1 in Appendix A provides more details of the implementation for attention-based MIL framework.

Note that authors in (Katharopoulos and Fleuret, 2019) compared uniform and adaptive instance sampling with other networks without sampling. But they fixed the attention network for the uniform sampling. We take a more fair approach and assume the same network architecture for all strategies we aim to compare in this paper.

3.3. Top-k sampling

As an alternative to the adaptive sampling is top-k sampling strategy which has been used in the computational pathology literature (Campanella et al., 2019; Sharmay et al., 2021). In this sampling strategy, top k instances with the highest instance-level score are selected to train the network. In this paper, such a score can be chosen as the attention weights. Therefore, we select top-k instances with the highest attention weights.

4. Dataset and Network Architecures

4.1. Datasets

We used the Cancer Genome Atlas (TCGA) repository of prostate adenocarcinoma (TCGA-PRAD) (Zuley et al., 2016) and Camelyon16 (Ehteshami Bejnordi et al., 2017) The Cancer Genome Atlas (TCGA) repository of prostate adenocarcinoma (TCGA-PRAD) dataset (Zuley et al., 2016) to compare and evaluate different sampling strategies. Please see Appendix B for more details about these datasets, deatils of pre-processing conducted on these datasets before feeing into the attention-based MIL networks.

4.2. Network architecture

The network architecture includes two sub-networks: attention backbone network and classification networks. To have a fair comparison, we consider the same network architecture (number of layers, nodes, activation function, etc.) for all of the sampling strategies. We also tune the network hyper-parameters to maximize the performance. To save computing time, we apply early stopping criterion to alleviate over-fitting problem. See Appendix C for more details on the network architecture and parameters, tuning hyper-parameters, and applied early stopping criterion.

5. Results

We consider binary classification task and compare four smapling strategies: (1) no sampling where we use whole instances over iterates (this is the standard attention MIL in (Ilse et al., 2018) and what is called CLAM-MIL in (Williamson et al., 2021)), (2) random sampling where we randomly draw choose G instances, (3) adaptive sampling where we adaptively select G instances, and (4) top-k sampling where we choose k instances with the highest ttention weights. We evaluate different strategies with both TCGA-PRAD and Camelyon16 datasets. We use the same network architecture, hyper-parameters, and tuning procedure (as explained in Appendix C) for all methods. For all methods, we choose minimum number of epochs as $e_{min} = 50$, maximum number of epochs as $e_{max} = 300$, and patience $e_{patience} = 20$ for early stopping. For random and adaptive sampling methods, we consider ten warm-up epochs ($e_{warm} = 10$) to train the model using whole instances initially. After that we pick $G = 10$ ($\sim 0.2\%$ of all available instances), 30, 100, 300, and 1000. We conducted all experiments on AWS nodes with one NVIDIA Tesla T4 GPU node, 32 CPUs, and 235 GB memory. Figure 1 compares the testing AUC, training time, and the number of training epochs after training is stopped by the early stopping algorithm (see Appendix C.3 for more details) for TCGA-PRAD (left panel) and Camelyon16 (right panel). We consider ten repetitions of Monte Carlo simulations for splitting data into training/validation/testing and we report *mean* ± *Standard error (SE)*. We observe that all instance sampling strategies reduce the computational complexity as expected. Also, we observe that random instance sampling with enough selected instances (e.g., around $G = 100$ or more) outperforms no sampling (i.e., using whole instances) strategy. Adaptive sampling might do better than random instance sampling when the number of instances per patient (bag) is minimal (around $G = 10$ or less) due to, e.g., memory constraints. Top-k sampling strategy performs the worst. From the results, we discover an important fact about using sampling strategies for the attention-based MIL networks: instance sampling strategies (versus using all instances) not only saves computing time and resources, but also can improve the performance the patients' disease status with WSI's.

6. Discussion

We investigated different instance sampling strategies for attention-based MIL networks. Instance sampling strategies significantly reduce computing time (and hence resources). Except for fewer selected instances, random sampling outperforms both the adaptive sampling and no sampling strategies. The justification is that random sampling makes the network sees almost different subsets of instances over different iterates (epochs) and play a role of regularization avoid over-fitting.

We used the pre-processing to throw out noisy (e.g., background) tiles or less-informative tiles beforehand. Then we used a pre-trained network (e.g., Resnet50) to extract low dimensional features from remaining image tiles after pre-processing. These two steps combined results in the embedding features that have more or less the same level of information about the outcome. This may result in assigning almost the same weights to the instances by the attention network. This could be another reason why the random sampling works better if we carefully choose the number selected instances G.

We considered a binary classification problem with a small dataset (with 318 patients) to evaluate our proposed model. However, it is worth extending our investigation to multiple-class classification tasks (e.g., the framework presented in (Williamson et al., 2021)) or other tasks such as survival prediction (Tarkhan and Simon, 2020; Tarkhan et al., 2021; Yao et al., 2020) using the attention-based MIL network.

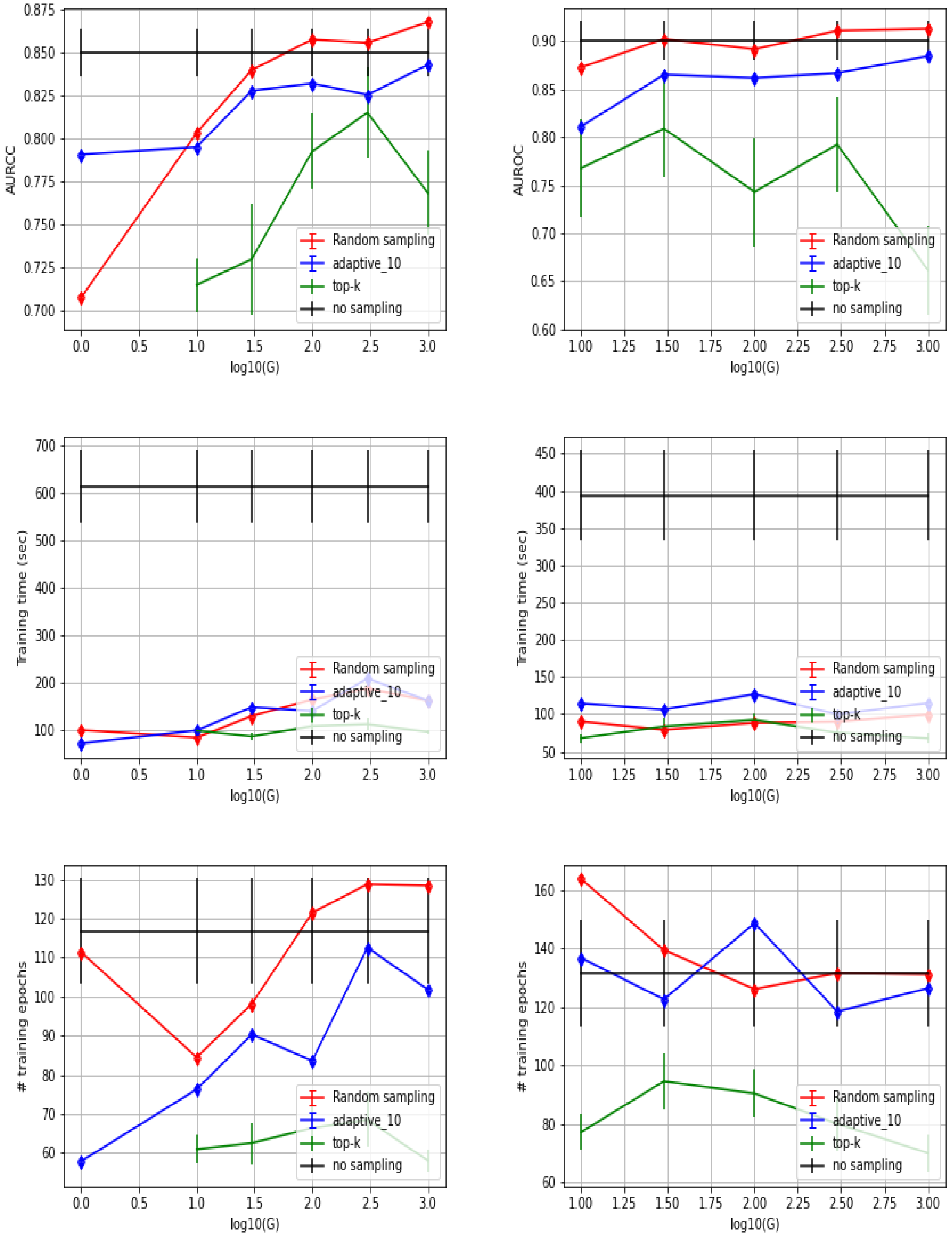

Figure 1: (left column) TCGA-PRAD (right column) Camelyon16; (top) Area under ROC curve, (middle) training time, and (bottom) number of training epochs; We compare different sampling strategies: **no sampling**, **random sampling**, **adaptive sampling**, and **top-k sampling**.

Acknowledgments

The authors would like to thank the Genentech Personalized Healthcare Digital Pathology Program for sponsoring project resourcing. The results published here are based upon data generated by the TCGA Research Network: https://www.cancer.gov/tcga. The authors declare the following competing interests: A.A and N.S. are employees of the University of Washington, and T.K.N. and J.D. are employees of Genentech.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

Chris M. Bishop. Training with noise is equivalent to tikhonov regularization. *Neural Computation*, 7(1):108–116, 1995. doi: 10.1162/neco.1995.7.1.108.

Tad T. Brunyé, Ezgi Mercan, Donald L. Weaver, and Joann G. Elmore. Accuracy is in the eyes of the pathologist: The visual interpretive process and diagnostic accuracy with digital whole slide images. *J. of Biomedical Informatics*, 66:171–179, 2010. doi: 10.1016/j.jbi.2017.01.004.

Gabriele Campanella, Matthew G. Hanna, Luke Geneslaw, Allen P. Miraflor, Vitor Werneck Krauss Silva, Klaus J. Busam, Edi Brogi, Victor E. Reuter, David S. Klimstra, and Thomas J. Fuchs. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. *Nature Medicine*, pages 1–9, 2019.

Miao Cui and David Y. Zhang. Artificial intelligence and computational pathology. *Laboratory Investigation*, 101:412–422, 2021. doi: https://doi.org/10.1038/s41374-020-00514-0.

Olivier Dehaene, Axel Camara, Olivier Moindrot, Axel de Lavergne, and Pierre Courtiol. Self-supervision closes the gap between weak and strong supervision in histology, 2020.

Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1):31–71, 1997. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(96)00034-3. URL https://www.sciencedirect.com/science/article/pii/S0004370296000343.

Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes van Diest, Bram van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen A. W. M. van der Laak, , and the CAMELYON16 Consortium. Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer. *JAMA*, 318(22): 2199–2210, 12 2017. ISSN 0098-7484. doi: 10.1001/jama.2017.14585. URL https://doi.org/10.1001/jama.2017.14585.

Jonathan I. Epstein. An update of the gleason grading system. *J. Urol*, 183(2):433–440, 2010. doi: 10.1016/j.juro.2009.10.046.

Filippo Fraggetta, Salvatore Garozzo, Gian F. Zannoni, Liron Pantanowitz, and Esther D. Rossi. Routine digital pathology workflow: the catania experience. *J Pathol Inform.*, 8 (51):1–6, Dec 2017. doi: 10.4103/jpi.jpi_58_17.

Donald F. Gleason and George T. Mellinger. Prediction of prognosis for prostatic adenocarcinoma by combined histological grading and clinical staging. *The Journal of Urology*, 111 (1):58–64, 1974. ISSN 0022-5347. doi: https://doi.org/10.1016/S0022-5347(17)59889-4. URL https://www.sciencedirect.com/science/article/pii/S0022534717598894.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

Maximilian Ilse, Jakub M. Tomczak, and Max Welling. Attention-based deep multiple instance learning, 2018.

Angelos Katharopoulos and François Fleuret. Processing megapixel images with deep attention-sampling models, 2019.

Pegah Khosravi, Maria Lysandrou, Mahmoud Eljalby, Qianzi Li, Ehsan Kazemi, Pantelis Zisimopoulos, Alexandros Sigaras, Matthew Brendel, Josue Barnes, Camir Ricketts, Dmitry Meleshko, Andy Yat, Timothy D. McClure, Brian D. Robinson, Andrea Sboner, Olivier Elemento, Bilal Chughtai, and Iman Hajirasouliha. A deep learning approach to diagnostic classification of prostate cancer using pathology-radiology fusion. *J Magn Reson Imaging*, 54(2):462–471, 2021. doi: 10.1002/jmri.27599.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Ruoyu Li, Jiawen Yao, Xinliang Zhu, Yeqing Li, and Junzhou Huang. Graph cnn for survival analysis on whole slide pathological images. In Alejandro F. Frangi, Julia A. Schnabel, Christos Davatzikos, Carlos Alberola-López, and Gabor Fichtinger, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, pages 174–182, Cham, 2018. Springer International Publishing. ISBN 978-3-030-00934-2.

Guoqing Liu, Jianxin Wu, and Zhi-Hua Zhou. Key instance detection in multi-instance learning. In *Proceedings of the Asian Conference on Machine Learning*, volume 25 of *Proceedings of Machine Learning Research*, pages 253–268. PMLR, 2012. URL https://proceedings.mlr.press/v25/liu12b.html.

Ming Y. Lu, Richard J. Chen, Jingwen Wang, Debora Dillon, and Faisal Mahmood. Semi-supervised histology classification using deep multiple instance learning and contrastive predictive coding, 2019.

Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. MIT Press, 1998. URL https://proceedings.neurips.cc/paper/1997/file/82965d4ed8150294d4330ace00821d77-Paper.pdf.

Sevda Molani, Mahboubeh Madadi, and Wesley Wilkes. A partially observable markov chain framework to estimate overdiagnosis risk in breast cancer screening: Incorporating uncertainty in patients adherence behaviors. *Omega*, 89:40–53, 2019. ISSN 0305-0483. doi: https://doi.org/10.1016/j.omega.2018.09.009. URL https://www.sciencedirect.com/science/article/pii/S030504831830001X.

NCCN. Nccn guidelines: Prostate cancer (version 4.2018). https://www2.tri-kobe.org/nccn/guideline/archive/urological2018/english/prostate.pdf, 2018. Accessed: 2021-11-11.

Shima Nofallah, Sachin Mehta, Ezgi Mercan, Stevan Knezevich, Caitlin J. May, Donald Weaver, Daniela Witten, Joann G. Elmore, and Linda Shapiro. Machine learning techniques for mitoses classification. *Computerized Medical Imaging and Graphics*, 87:101832, 2021. ISSN 0895-6111. doi: https://doi.org/10.1016/j.compmedimag.2020.101832. URL https://www.sciencedirect.com/science/article/pii/S0895611120301270.

Sebastian Otálora, Niccolò Marini, Henning Müller, and Manfredo Atzori. Combining weakly and strongly supervised learning improves strong supervision in gleason pattern classification. *BMC Med Imaging*, 21(77):1–14, 2021. doi: https://doi.org/10.1186/s12880-021-00609-0.

Sinno J. Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. doi: 10.1109/TKDE.2009.191.

Lutz Prechelt. *Early Stopping — But When?*, pages 53–67. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-35289-8. doi: 10.1007/978-3-642-35289-8_5. URL https://doi.org/10.1007/978-3-642-35289-8_5.

Gwenole Quellec, Guy Cazuguel, Beatrice Cochener, and Mathieu Lamard. Multiple-instance learning for medical image and video analysis. *IEEE Reviews in Biomedical Engineering*, 10:213–234, 2017. doi: 10.1109/RBME.2017.2651164.

Colin Raffel and Daniel P. W. Ellis. Feed-forward networks with attention can solve some long-term memory problems, 2016.

Jan Ramon and Luc D. Raedt. Multi instance neural networks. In *Proceedings of the ICML*, Workshop on Attribute-value and Relational Learning, pages 53–60, 2000.

Vikas C. Raykar, Balaji Krishnapuram, Jinbo Bi, Murat Dundar, and R. Bharat Rao. Bayesian multiple instance learning: automatic feature selection and inductive transfer. *ICML*, pages 808–815, 2008. doi: https://doi.org/10.1145/1390156.1390258.

Yash Sharmay, Lubaina Ehsany, Sana Syed, and Donald E. Brown. Histotransfer: Understanding transfer learning for histopathology. In *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, pages 1–4, 2021. doi: 10.1109/BHI50953.2021.9508542.

Aliasghar Tarkhan and Noah Simon. Bigsurvsgd: Big survival data analysis via stochastic gradient descent, 2020.

Aliasghar Tarkhan, Noah Simon, Thomas Bengtsson, Kien Nguyen, and Jian Dai. Survival prediction using deep learning. In *Proceedings of AAAI Spring Symposium on Survival Prediction - Algorithms, Challenges, and Applications 2021*, volume 146 of *Proceedings of Machine Learning Research*, pages 207–214. PMLR, 22–24 Mar 2021. URL https://proceedings.mlr.press/v146/tarkhan21a.html.

Ming Y. Luand Drew F. K. Williamson, Tiffany Y. Chen, Richard J. Chen, Matteo Barbieri, and Faisal Mahmood. Data-efficient and weakly supervised computational pathology on whole-slide images. *Nat Biomed Eng*, 5:555—-570, 2021. doi: https://doi.org/10.1038/s41551-020-00682-w.

Ellery Wulczyn, David F. Steiner, Zhaoyang Xu, Apaar Sadhwani, Hongwu Wang, Isabelle Flament-Auvigne, Craig H. Mermel, Po-Hsuan Cameron Chen, Yun Liu, and Martin C. Stumpe. Deep learning-based survival prediction for multiple cancer types using histopathology images. *PLOS ONE*, 15(6):e0233678, Jun 2020. ISSN 1932-6203. doi: 10.1371/journal.pone.0233678. URL http://dx.doi.org/10.1371/journal.pone.0233678.

J. Yao, X. Zhu, J. Jonnagaddala, N. Hawkins, and Junzhou Huang. Whole slide images based cancer survival prediction using attention guided deep multiple instance learning networks. *Medical Image Analysis*, 65:101789, 2020. ISSN 1361-8415. doi: 10.1016/j.media.2020.101789. URL http://dx.doi.org/10.1016/j.media.2020.101789.

Xinliang Zhu, Jiawen Yao, and Junzhou Huang. Deep convolutional neural network for survival analysis with pathological images. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 544–547, 2016. doi: 10.1109/BIBM.2016.7822579.

Margarita L. Zuley, Rose Jarosz, Bettina F. Drake, Danielle Rancilio, Aleksandra Klim, Kimberly Rieger-Christ, and John Lemmerman. Radiology data from the cancer genome atlas prostate adenocarcinoma [tcga-prad] collection. *Cancer Imaging Arch*, 2016.

Appendix A. Architecture and flowchart of presented sampling strategies

Figure A illustrates the general architecture, and Algorithm 1 provides more details of the implementation for the presented instance sampling strategies in Section 3.

Appendix B. More details on datasets

B.1. TCGA-PRAD (prostate cancer) dataset

The Cancer Genome Atlas (TCGA) repository of prostate adenocarcinoma (TCGA-PRAD) dataset (Zuley et al., 2016) to evaluate our proposed approach. The Gleason score (GS) from the biopsied tissue is the common method to measure the cancer status (Gleason and Mellinger, 1974). The GS is the sum of primary and secondary scores, and each ranges from 3 to 5. Therefore, the GS ranges from 6 (3+3) to 10 (5+5). Another alternative and commonly-used scoring system is Grade Group (GG) which divides the prostates cancer patients into five groups based on the pathological patterns. Table 1 summarizes GS, GG, and corresponding risk levels based on *NCCN Clinical Practice Guidelines in Oncology* (NCCN, 2018). Both GG and GS have been widely used in prostates cancer studies (Khosravi et al., 2021).

We followed the same procedure for sampling (with 20× magnification) image tiles (with size 256×256) from WSIs and the same procedure for pre-processing image tiles as explained and used in Williamson et al. (2021). The bag (patient) size varies among patients, with a minimum of 1,308, a maximum of 130,752, and an average of 49,811 image tiles. To reduce computing time and cost, we used pre-trained network Resnet50 (He et al., 2015) to extract features from image tiles (instances) into one-dimensional embedding features with size 1,024 (this procedure is known as transfer learning (Pan and Yang, 2010)). We consider binary classification where we divide patients into two classes: class 0 includes *low risk* (grade group 1) and *favorable intermediate* (grade group 2); and class 1 includes *unfavorable intermediate risk* (grade group 3), *high risk* (grade group 4), and *very high risk* (grade group 5). The resulted dataset has 318 patients with 129 patients with class 0 and 189 patients with class 1.

B.2. Camelyon16 (breast cancer) dataset

The Camelyon16 dataset is about breast cancer (Ehteshami Bejnordi et al., 2017). It is difficult and time-consuming to detect lymph node metastases with the gigapixel sized images. An automated detection of breast cancer metastases in lymph node pictures is of interest. We use the same pre-processing as we used for TCGA-PRAD dataset. We also used pre-trained network Resnet50 (He et al., 2015) to extract features from image tiles (instances) into one-dimensional embedding features with size 1,024 (this procedure is known as transfer learning (Pan and Yang, 2010)). After pre-processing and feature extraction, we are left with 80 patients (bags) with cancerous tissue and 123 patients with normal tissue.

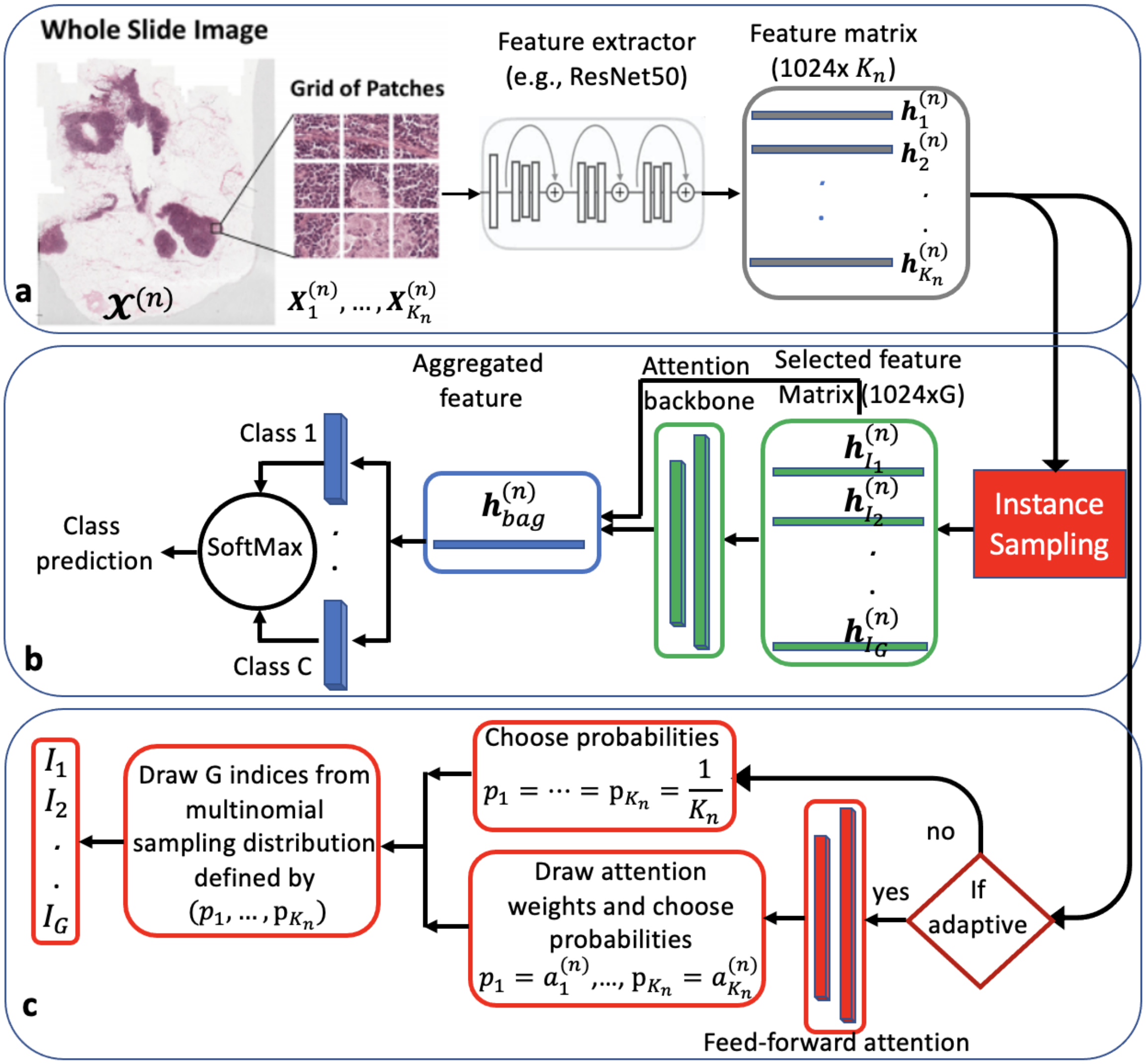

Figure 2: **Overview of our proposed adaptive architecture. a: pre-processing** We use segmentation to sample patches from the WSI $\boldsymbol{X}^{(n)}$. Then, we pass the patches through the pre-trained network (e.g., ResNet50) to extract lower-dimensional features vectors $\boldsymbol{h}_1^{(n)}$, ..., $\boldsymbol{h}_{K_n}^{(n)}$. **b: training procedure** We use the sampling strategy either random or adaptive (see panel **c** for more details) to sample a subset of G instances out of K_n instances per bag (patient). Then, we aggregate G selected instances using the attention network to get a single bag-level representative feature. Finally, we predict the class label using the aggregated feature as the input of the classification network. **c: instance sampling procedure** We consider all instances for subject n and feed them into the trained (fixed) feed-forward attention network to estimate the sampling distribution $\mathcal{P}^{(n)}$. Finally, we draw G instances out of K_n instances from distribution $\mathcal{P}^{(n)}$.

Algorithm 1: Instance sampling for attention-based MIL network.

```
Result: Test AUC
Initialization:
  Number of selected instances for training: $G$
  Number of warm-up iterates/epochs : $e_{warm}$
  Number of epochs for updating sampling distribution: $e_{update}$
  Minimum number of epochs: $e_{min}$
  Maximum number of epochs: $e_{max}$
  Number of epochs before early stop: $e_{patience}$
Start with e=1
while $e \leq e_{max}$ do
    if No improvement on validation loss for at least $e_{patience}$ and $e \geq e_{min}$ then
        Stop training
    else
        for $(n = 1, 2, ..., N)$ do
            if Uniform sampling then
                Choose
                    $$p_k^{(n)} = \frac{1}{K_n}, \quad k = 1, 2, \ldots, K_n$$
            else
                if $(e \leq e_{warm})$ then
                    Use all instances to train the model
                else
                    if $(e \bmod e_{update} == 0)$ then
                        Update sampling distribution $\mathcal{P}^{(n)}$ using the attention weights
                          extracted from the forward attention network as
                            $$p_k^{(n)} = a_k^{(n)}, \quad k = 1, 2, \ldots, K_n$$
                end
            end
            Sample $G$ instances from $\mathcal{P}^{(n)}$:
                $$(I_1^{(n)}, I_2^{(n)}, \ldots, I_G^{(n)}) \sim \mathcal{P}^{(n)}$$
            Train the attention MIL network using selected $G$ instances.
        end
        if The validation loss is improved then
            Save the current model
        e=e+1
    end
end
Use the saved model and report test AUC
```

Table 1: Grade Group, Gleason score, and their association with the risk level

Grade Group	Gleason score	Combined Gleason Score	Risk level
1	3+3	6	Low risk
2	3+4	7	Favorable intermediate
3	4+3	7	Unfavorable intermediate
4	4+4, 3+5, 5+3	8	High risk
5	4+5, 5+4, 5+5	9 and 10	Very high risk

Appendix C. Network architecture, tuning hyper-parameters and early stopping

C.1. Network architecture

First, we consider a fully connected layer $\boldsymbol{W}_d \in \mathbb{R}^{1024\times 512}$ with ReLU activation function to reduce the dimension feature embedding space from 1024 to 512. For the attention network, we consider the gated attention with $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{256\times 512}$, each followed by single shared branch $\boldsymbol{w} \in \mathbb{R}^{256\times 1}$. For the classification network, we choose a fully connected layer $\boldsymbol{W}_c \in \mathbb{R}^{512\times C}$ where we choose $C = 2$ for binary classification. We use the Adam algorithm (Kingma and Ba, 2017) to optimize the parameters of deep neural network for all methods. To find the best possible model for classification, we consider different hyper-parameters for all methods evaluated in this paper: initial learning rate with values $(10^{-4}, 10^{-3})$, regularization rate $(10^{-5}, 10^{-3})$, and dropout rate $(0.2, 0.5)$. See Appendix C.2 for more details about hyper-parameters tuning and early-stopping procedures used in this paper.

C.2. Tuning hyper-parameters

To find the best possible model for classification, we consider different hyper-parameters for all methods evaluated in this paper: initial learning rate with values $(10^{-4}, 10^{-3})$, regularization rate $(10^{-5}, 10^{-3})$, and dropout rate $(0.2, 0.5)$. We consider the following steps for tuning these hyper-parameters:

- We randomly split data into training/validation/testing datasets (80% training, 10% validation, and 10% training),
  - For each combination of hyper-parameters, we do the following,
    - We train the model on training dataset until we are confident that there will be no improvement of the validation AUC by further training. We use a stopping criterion (see Appendix C.3) to determine when to stop training.
    - We save the trained model at epoch maximizing the validation AUC
    - With the saved model and testing dataset, we calculate the testing AUC.

Finally, we report the average testing AUC over repetitions of randomly split datasets.

C.3. Early stopping criterion

It is crucial to determine the ideal training length for a neural network: While too little training gives an under-fit model, too much training over-fits and results in poor performance on the test dataset. One common approach is to train the model on the training dataset until the performance on a validation dataset stops improving. This widely used approach to training a neural network is known as *early stopping* (Prechelt, 2012). In practice, the validation error curve is not usually smooth and has some stochastic behavior due to stochastic optimization. Therefore, there might have several nearby local minima (Prechelt, 2012). To deal with this, we can continue training for a few epochs past where the local minimum is initially identified to increase confidence that there will be no later improvement in performance on the validation set. This extra number of epochs approach is known as *patience* (Prechelt, 2012). We consider *patience*=10 for all sampling methods we investigate in this paper.