# Supplemental Material: Differentiable Zooming for Multiple Instance Learning on Whole-Slide Images

Kevin Thandiackal[1,2*], Boqi Chen[1,2*], Pushpak Pati[1], Guillaume Jaume[3†], Drew F. K. Williamson[3], Maria Gabrani[1], and Orcun Goksel[2,4]

[1] IBM Research Europe, Zurich, Switzerland
{kth,che,pus,mga}@zurich.ibm.com
[2] ETH Zurich, Zurich, Switzerland
[3] Brigham and Women's Hospital, Harvard Medical School, Boston, USA
{gjaume,dwilliamson}@bwh.harvard.edu
[4] Uppsala University, Uppsala, Sweden
orcun.goksel@it.uu.se

In this supplemental material, we include additional results, visualizations, and analyses. The contents of the individual sections are:

- Appendix A: Confusion matrices for ZoomMIL on all datasets. Additional results for the competing methods operating at different magnifications
- Appendix B: Analyzing the limitations of ZoomMIL
- Appendix C: Analyzing the impact of the number of selected patches in the differentiable Top-K module
- Appendix D: ZoomMIL attention maps at different magnifications
- Appendix E: Training details of the competing baselines
- Appendix F: Derivation of the Jacobian for differentiable patch selection

## A Additional Classification Analysis

In Figure 1, we present the confusion matrices of ZoomMIL on all datasets. Results are averaged over three runs with different weight initializations. On CRC, ZoomMIL performs very well and correctly classifies 95.63% and 94.55% of the non-neoplastic and low-grade cases, respectively. Out of all high-grade, cases, our model identifies 83.33% correctly. We can see that BRIGHT is the most difficult dataset due to its challenging pre-cancerous class, which is often confused with either the non-cancerous or the cancerous class. On CAMELYON16, ZoomMIL accurately identifies 97.92 % of non-metastatic cases, while correctly classifying 61.90 % of metastatic cases.

Additionally, we report the classification performance of all single-scale baselines operating at different magnifications in Table 1, 2, and 3. It can be observed in Table 1 and 2 that lower magnifications severely impact the performance of the baselines, while ZoomMIL-Eff performs significantly better. These results indicate the efficacy of our method, and conclude the benefits of zooming and combining information across magnifications.

[*] Contributed equally.
[†] Work done while at IBM Research Europe

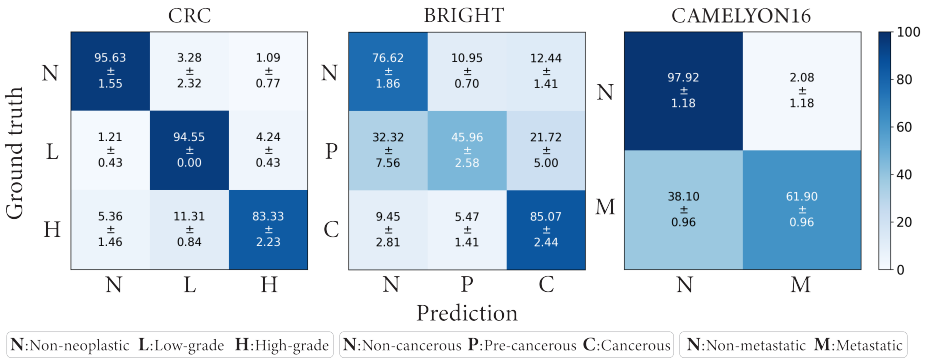


**Fig. 1.** Confusion matrices of ZoomMIL on CRC, BRIGHT, and CAMELYON16. An entry represents the corresponding fraction (%) w.r.t. all samples in the same row

**Table 1.** Classification performances on the CRC dataset [11]

| Methods | Classification | |
|---|---|---|
| | Weighted-F1(%) | Accuracy(%) |
| ABMIL [6] (5×) | 86.8±0.7 | 87.0±0.7 |
| ABMIL [6] (10×) | 88.8±0.7 | 89.0±0.6 |
| CLAM-SB [10] (5×) | 87.7±0.5 | 87.8±0.5 |
| CLAM-SB [10] (10×) | 89.5±0.5 | 89.6±0.5 |
| TransMIL [12] (5×) | 86.2±1.1 | 87.4±1.1 |
| TransMIL [12] (10×) | 88.4±1.3 | 89.1±1.1 |
| ZoomMIL-Eff (5× → 10×) | 90.3±1.3 | 90.3±1.3 |
| ZoomMIL (5× → 10× → 20×) | 92.0±0.6 | 92.1±0.7 |

**Table 2.** Classification performances on the BRIGHT dataset [3]

| Methods | Classification | |
|---|---|---|
| | Weighted-F1(%) | Accuracy(%) |
| ABMIL [6] (1.25×) | 58.4±1.0 | 58.9±1.6 |
| ABMIL [6] (2.5×) | 58.7±1.1 | 59.3±1.0 |
| CLAM-SB [10] (1.25×) | 59.9±1.3 | 60.3±1.2 |
| CLAM-SB [10] (2.5×) | 60.1±1.2 | 60.2±1.6 |
| TransMIL [12] (1.25×) | 46.1±3.8 | 47.3±2.5 |
| TransMIL [12] (2.5×) | 52.0±1.3 | 54.5±2.7 |
| ZoomMIL-Eff (1.25× → 2.5×) | 66.0±1.9 | 66.5±1.5 |
| ZoomMIL (1.25× → 2.5× → 10×) | 68.3±1.1 | 69.3±1.0 |

## B Limitations

We conjecture that our classification performance on CAMELYON16 is limited by the size of the metastatic regions. For the samples including extremely small

**Table 3.** Classification performances on the CAMELYON16 dataset [1]

| Methods | Classification | |
|---|---|---|
| | Weighted-F1(%) | Accuracy(%) |
| ABMIL [6] (10×) | 76.7±0.8 | 78.3±0.7 |
| CLAM-SB [10] (10×) | 77.5±0.6 | 79.1±0.6 |
| TransMIL [12] (10×) | 76.6±1.1 | 79.6±1.0 |
| ZoomMIL (10× → 20×) | 83.3±0.3 | 84.2±0.4 |

**Table 4.** Classification performances on the validation set of CAMELYON16 [1]. Validation set is further grouped according to the size of the metastatic regions

| Size of metastases | Classification | |
|---|---|---|
| | Weighted-F1(%) | Accuracy(%) |
| large | 96.2±1.0 | 96.1±1.1 |
| small | 86.7±1.0 | 87.1±1.0 |

metastases, it is challenging to optimize the zooming process. To validate our hypothesis, we sub-categorize the metastatic samples in the training set into "small" and "large" metastatic groups, via visual inspection, and create new stratified training and validation sets. In Table 4, we present the classification performances of ZoomMIL on the validation set, individually for the small and large metastatic samples. The results show that the performance is significantly higher for large metastases, which substantiates our hypothesis.

## C Impact of K in Differentiable Top-K Patch Selection

Here, we analyze the impact of the number of selected patches ($K$) on the classification performance of ZoomMIL. Figure 2 shows that the performance increases with increasing $K$. It peaks at $K = 12$ and then slightly drops for further increment. We reason that this behavior is caused by the average number of patches per Whole-Slide Image (WSI) being 16 in the BRIGHT dataset at the lowest magnification (1.25×). Almost all patches are selected in this case, which makes it suboptimal to learn to improve the zooming process.

## D Interpretability

We have shown the ZoomMIL attention maps at the lowest magnification, *i.e.*, 1.25×, on the BRIGHT dataset in Figure 4 of the main manuscript. Here, in Figure 3, we provide the attention maps at both low and high magnification, *i.e.*, 1.25× and 10×, as well as close-up patches with highest attention scores. We can observe in the zooming process from 1.25× to 10×, that the model pins down its focus to the most informative regions in the WSI. The observation is

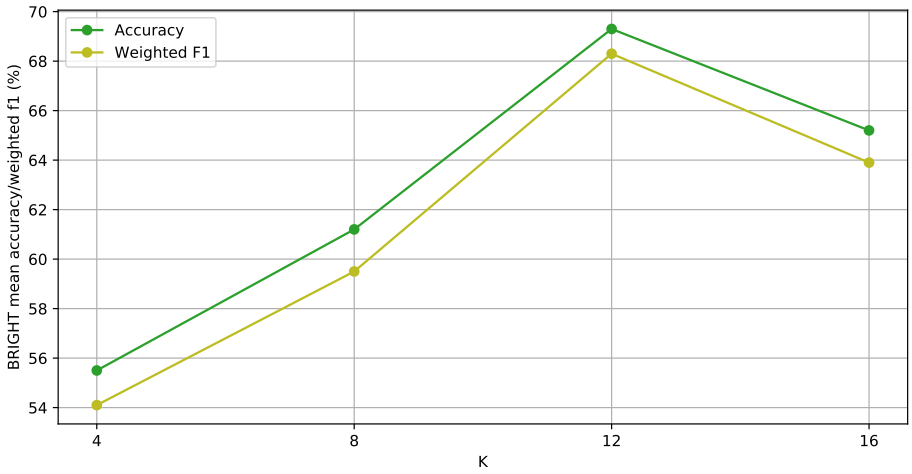


**Fig. 2.** Classification performance of ZoomMIL on BRIGHT for different values of $K$

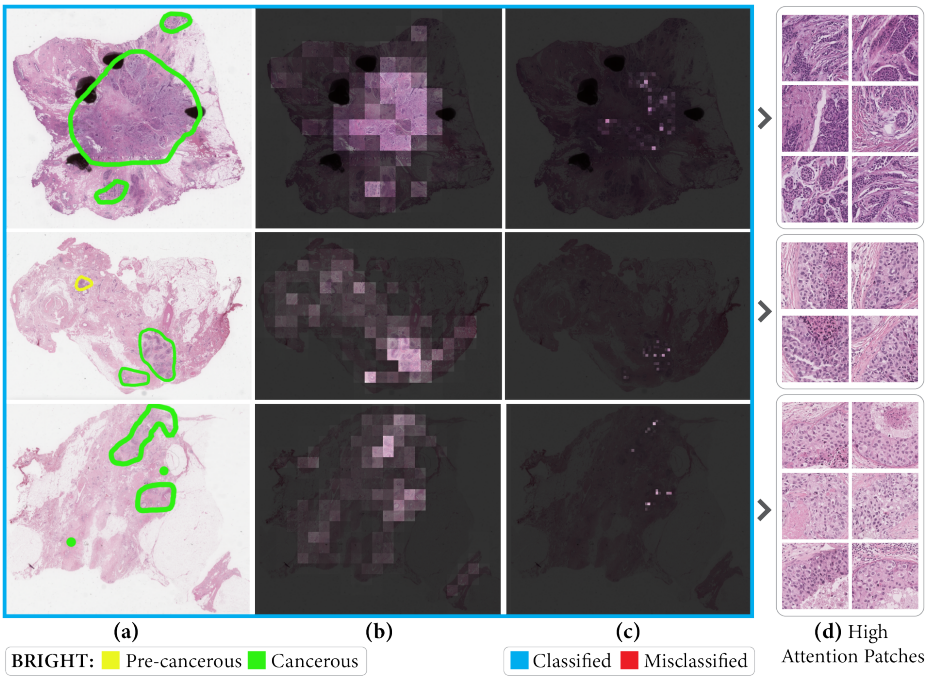


**Fig. 3.** Interpreting ZoomMIL on the BRIGHT dataset: (a) annotated tumor regions, (b) attention maps at 1.25× magnification, (c) attention maps at 10× magnification, and (d) a subset of extracted patches with high attention scores at 10× magnification

further substantiated by the high attention patches that include cancerous content, *i.e.*, invasive tumors and ductal carcinoma in situ tumors, for the presented cancerous WSIs. Figure 4 shows more examples of annotated tumor regions and attention maps for correctly classified and misclassified samples across all the classes in BRIGHT. For both the correct and incorrect classifications, we can notice that the focus of the model is aligning with the focus of the pathologist. However, the morphological ambiguities among the classes finally lead to certain misclassifications.

## E Implementation Details

**ABMIL:** We follow the network architecture proposed in [6]. The model comprises of a gated attention module consisting of three 2-layer Multi-Layer Perceptrons (MLPs), where the first two are followed by Hyperbolic Tangent and Sigmoid activations, and a 2-layer MLP classifier with ReLU activation.

**CLAM-SB:** We implement CLAM-SB [10] with the code[1] provided by the authors. We use the Adam optimizer [7] with 0.0001 learning rate. The maximum and minimum epochs are set to 100, 50, respectively, and use early stopping with patience=20 epochs based on the validation weighted F1 score (for BRIGHT) and validation loss (for CRC and CAMELYON16). We use cross-entropy loss for both bag loss and instance loss. The weights of bag-level losses are set to $0.7, 0.5, 0.7$, and the number of positive/negative patches sampled for instance loss are set to $8, 32, 32$ for CRC, BRIGHT and CAMELYON16, respectively. For all three datasets we use a weighted sampler due to class imbalance.

**TransMIL:** We adopt the original implementation[2] of TransMIL [12]. We use Lookahead optimizer [13] with learning rate 0.0002 and weight decay 0.00001. The maximum epochs are set to 200 and early stopping is used with a *patience* of 10 epochs based on validation weighted F1 score (for BRIGHT) and validation loss (for CRC and Camelyon). We use cross-entropy loss as training loss.

**MSMIL:** We adopt the model architecture from the original implementation[3] of MSMIL [5]. We construct the multi-scale feature matrix $f \in \mathbb{R}^{(N_0+\cdots+N_{n-1})\times d}$ by concatenating the feature matrices $f_0, \cdots, f_{n-1}$ at all magnifications $m_0, \cdots, m_{n-1}$ where $f_n \in \mathbb{R}^{N_n \times d}$ and $N_n = N_{n-1}(\frac{m_n}{m_{n-1}})^2$ is the number of features at magnification $m_n$. Here, $d$ is the feature dimension and $n = \{3, 3, 2\}$ is the number of considered magnifications on CRC, BRIGHT, and CAMELYON16, respectively. We use the Adam optimizer [7] with learning rate 0.0001 and plateau scheduler (patience= 5 epochs, decay rate= 0.8). The experiments are run for 100 epochs with a batch size of one. The models with the best weighted F1-score (for BRIGHT) and best loss (for CRC & CAMELYON16) on the validation set are saved for testing.

**DSMIL:** We adopt the model architecture from the original implementation[4] of DSMIL [9]. We concatenate the $d$-dimensional feature vector of each patch

[1] `https://github.com/mahmoodlab/CLAM`
[2] `https://github.com/szc19990412/TransMIL`
[3] `https://github.com/takeuchi-lab/MS-DA-MIL-CNN`
[4] `https://github.com/binli123/dsmil-wsi`

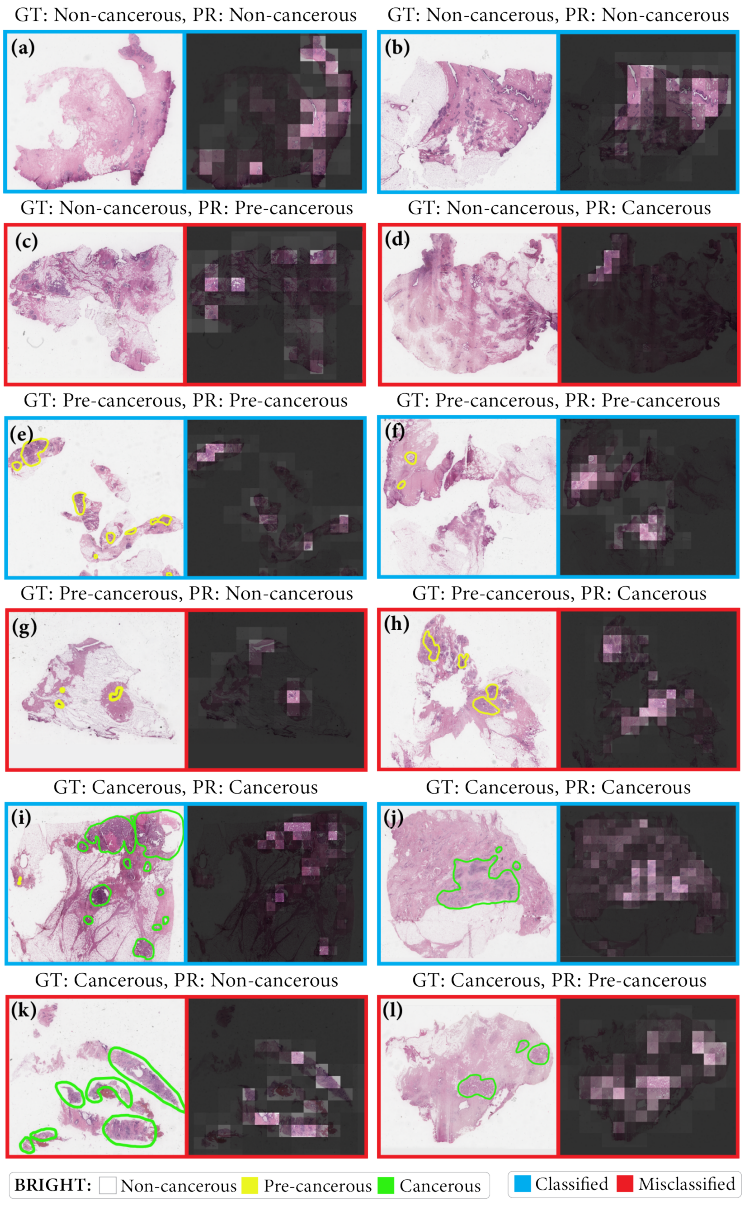


**Fig. 4.** Examples of BRIGHT WSIs with annotated tumor regions and attention maps from ZoomMIL at 1.25× magnification

at low-magnification $m_0$ with $d$-dimensional feature vectors of its corresponding sub-patches at higher magnifications $m_1, \cdots, m_{n-1}$ to form an $nd$-dimensional multi-scale feature vector where $n = \{3, 3, 2\}$ is the number of considered magnifications on CRC, BRIGHT, and CAMELYON16, respectively. Each feature vector from a lower magnification patch is replicated and concatenated to each of its higher magnification counterparts. We use the Adam optimizer [7] with learning rate 0.0001 and plateau scheduler (patience= 5 epochs, decay rate= 0.8). The experiments are run for 100 epochs with a batch size of one. The models with the best weighted F1-score (for BRIGHT) and best loss (for CRC & CAMELYON16) on the validation set are saved for testing.

**SparseConvMIL:** We adopt the original implementation[5] of SparseConvMIL [8]. Unlike other baselines, we run the model on a V100 GPU with 32GB RAM, as it depends on SparseConvNet[6]. Therefore, we limit the batch size and the number of sampled patches to 8 and 100, respectively. In the model, we set the number of sparseconv channels to 32, the downsampling factor of the sparse map to 128, and the neurons in the MLP classifier to 128. We use ResNet34 for extracting patch features, and finetune it with learning rate 0.00001. The learning rate for the rest of the model is 0.001, and weight decay is 0.0001.

**MaxMIL & MeanMIL:** We use the formulation presented in [8] for Max and Mean MIL. We set the same values for the hyperparameters as in SparseConvMIL, but freeze the patch feature extractor, *i.e.*, ResNet34.

## F Derivation of Jacobian for Differentiable Patch Selection

**Perturbed Maximum** As described in [2], given a set of distinct points $\mathcal{T} \subset \mathbb{R}^d$ and its convex hull $\mathcal{C}$, a discrete optimization problem with inputs $\mathbf{a}_m \in \mathbb{R}^d$ can generally be formulated as:

$$\max_{\hat{\mathbf{t}} \in \mathcal{C}} \langle \hat{\mathbf{t}}, \mathbf{a}_m \rangle \qquad \mathbf{t} = \arg\max_{\hat{\mathbf{t}} \in \mathcal{C}} \langle \hat{\mathbf{t}}, \mathbf{a}_m \rangle \ . \tag{1}$$

As per Definition 2.1 in [2], we can obtain a smoothed $\mathbf{t}$ by adding a random noise vector $\sigma\mathbf{Z} \in \mathbb{R}^d$ with distribution $\mathrm{d}\mu(\mathbf{z}) \propto \exp(-\nu(\mathbf{z}))\mathrm{d}\mathbf{z}$, where $\sigma > 0$ is a scaling parameter. The perturbed version of the maximizer in Eq. (1) then becomes:

$$\mathbf{t} = \mathbb{E}\left[ \arg\max_{\hat{\mathbf{t}} \in \mathcal{C}} \langle \hat{\mathbf{t}}, \mathbf{a}_m + \sigma\mathbf{Z} \rangle \right] . \tag{2}$$

According to Proposition 3.1 from [2], the associated Jacobian matrix of $\mathbf{t}$ at $\mathbf{a}_m$ can then be computed as follows:

$$J_{\mathbf{a}_m}\mathbf{t} = \mathbb{E}\left[ \arg\max_{\hat{\mathbf{t}} \in \mathcal{C}} \langle \hat{\mathbf{t}}, \mathbf{a}_m + \sigma\mathbf{Z} \rangle \nabla_{\mathbf{z}} \nu(\mathbf{Z})^\top / \sigma \right] . \tag{3}$$

[5] https://github.com/MarvinLer/SparseConvMIL

[6] https://github.com/facebookresearch/SparseConvNet

We choose our noise to have a normal distribution, *i.e.*, $\mathbf{Z} \sim \mathcal{N}(0, \mathbb{1})$. We can thus plug $\nabla_{\mathbf{z}} \nu(\mathbf{Z})^\top = \mathbf{Z}^\top$ into Eq. (3) and obtain:

$$J_{\mathbf{a}_m} \mathbf{t} = \mathop{\mathbb{E}}_{\mathbf{Z} \sim \mathcal{N}(0,\mathbb{1})} \left[ \arg\max_{\hat{\mathbf{t}} \in \mathcal{C}} \langle \hat{\mathbf{t}}, \mathbf{a}_m + \sigma \mathbf{Z} \rangle \mathbf{Z}^\top / \sigma \right] . \tag{4}$$

**Differentiable Top-K Operator** As shown in [4], the Top-K selection with sorted indices can be converted into the same form as Eq. (1). To this end, the constraint set $\mathcal{C}$ for the indicator matrix $\hat{\mathbf{T}}$ should first be defined as:

$$\mathcal{C} = \Big\{ \hat{\mathbf{T}} \in \mathbb{R}^{N \times K} : \quad \hat{\mathbf{T}}_{n,k} \geq 0 \tag{5}$$

$$\sum_{j=1}^{N} \hat{\mathbf{T}}_{j,k} = 1 \quad \forall k \in \{1, \dots, K\} \tag{6}$$

$$\sum_{k=1}^{K} \hat{\mathbf{T}}_{j,k} \leq 1 \quad \forall j \in \{1, \dots, N\} \tag{7}$$

$$\sum_{i=1}^{N} i \hat{\mathbf{T}}_{i,k} < \sum_{j=1}^{N} j \hat{\mathbf{T}}_{j,k'} \quad k < k' \Big\} , \tag{8}$$

where Eq. (6) ensures that each column-wise sum in the indicator matrix is equal to one and Eq. (7) constrains each row-wise sum to be at most one. Lastly, Eq. (8) enforces that the indices of the attention weights selected by $\hat{\mathbf{T}}$ are sorted. With these constraints, the general linear program formulation in Eq. (1) can be used to describe the Top-K selection problem:

$$\max_{\hat{\mathbf{T}} \in \mathcal{C}} \langle \hat{\mathbf{T}}, \mathbf{a}_m \mathbf{1}^\top \rangle \qquad \mathbf{T} = \arg\max_{\hat{\mathbf{T}} \in \mathcal{C}} \langle \hat{\mathbf{T}}, \mathbf{a}_m \mathbf{1}^\top \rangle , \tag{9}$$

where $\mathbf{1}^\top = [1 \cdots 1] \in \mathbb{R}^{1 \times K}$ and thus $\mathbf{a}_m \mathbf{1}^\top \in \mathbb{R}^{N \times K}$ is a matrix containing the attention vectors $\mathbf{a}_m$ repeated $K$ times as its columns. Note that here, $\langle \cdot \rangle$ computes the scalar product after vectorizing the matrices.

Now, the perturbed maximizer $\mathbf{T}$ and the corresponding Jacobian $J_{\mathbf{a}_m} \mathbf{T}$ can be computed analogously to Eq. (2) and Eq. (4), as presented in the main paper. In contrast to [2], however, for the Top-K selection problem, it is required to apply the same noise vector $\sigma \mathbf{Z}$ to each column in $\mathbf{a}_m \mathbf{1}^\top$. Following the insights from [4], we therefore apply in practice the noise directly to $\mathbf{a}_m$, *i.e.*, we employ $\big(\mathbf{a}_m + \sigma \mathbf{Z}\big) \mathbf{1}^\top$ instead of $\mathbf{a}_m \mathbf{1}^\top + \sigma \mathbf{Z}$.

## References


1. Bejnordi, B., Veta, M., van Dienst, P., van Ginneken, B., Karssemeijer, N., Litjens, G., van der Laak, J., et al.: Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. JAMA **318**, 2199–2210 (2017)
2. Berthet, Q., Blondel, M., Teboul, O., Cuturi, M., Vert, J., Bach, F.: Learning with differentiable pertubed optimizers. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 34, pp. 9508–9519 (2020)
3. Brancati, N., Anniciello, A., Pati, P., Riccio, D., Scognamiglio, G., Jaume, G., De Pietro, G., Di Bonito, M., Foncubierta-Rodríguez, A., Botti, G., et al.: Bracs: A dataset for breast carcinoma subtyping in h&e histology images. arXiv:2111.04740 (2021)
4. Cordonnier, J., Mahendran, A., Dosovitskiy, A.: Differentiable patch selection for image recognition. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2351–2360 (2021)
5. Hashimoto, N., Fukushima, D., Koga, R., Takagi, Y., Ko, K., Kohno, K., Nakaguro, M., Nakamura, S., Hontani, H., Takeuchi, I.: Multi-scale domain-adversarial multiple-instance cnn for cancer subtype classification with unannotated histopathological images. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3852–3861 (2020)
6. Isle, M., Tomczak, J., Welling, M.: Attention-based deep multiple instance learning. In: International Conference on Machine Learning (ICML). vol. 35 (2018)
7. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015)
8. Lerousseau, M., Vakalopoulou, M., Deutsch, E., Paragios, N.: Sparseconvmil: Sparse convolutional context-aware multiple instance learning for whole slide image classification. In: MICCAI Workshop on Computational Pathology. pp. 129–139 (2021)
9. Li, B., Li, Y., Eliceiri, K.: Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14318–14328 (2021)
10. Lu, M., Williamson, D., Chen, T., Chen, R., Barbieri, M., Mahmood, F.: Data efficient and weakly supervised computational pathology on whole slide images. Nature Biomedical Engineering **5**, 555–570 (2021)
11. Oliveira, S., Neto, P., Fraga, J., Montezuma, D., Monteiro, A., Monteiro, J., Ribeiro, L., Gonçalves, S., Pinto, I., Cardoso, J.: Cad systems for colorectal cancer from wsi are still not ready for clinical acceptance. Scientific Reports **11** (2021)
12. Shao, Z., Bian, H., Chen, Y., Wang, Y., Zhang, J., Ji, X., Zhang, Y.: Transmil: Transformer based correlated multiple instance learning for whole slide image classification. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 35 (2021)
13. Zhang, M., Lucas, J., Hinton, G., Ba, J.: Lookahead optimizer: k steps forward, 1 step back. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 33 (2019)