Supplementary material of "SparseConvMIL: Sparse Convolutional Context-Aware Multiple Instance Learning for Whole Slide Image Classification"

Marvin Lerousseau MARVIN.LEROUSSEAU@CENTRALESUPELEC.FR
CentraleSupélec

Maria Vakalopoulou
CentraleSupélec

Eric Deutsch
Gustave Roussy

Nikos Paragios
CentraleSupélec

Editors: M. Atzori, N. Burlutskiy, F. Ciompi, Z. Li, F. Minhas, H. Müller, T. Peng, N. Rajpoot, B. Torben-Nielsen, J. van der Laak, M. Veta, Y. Yuan, and I. Zlobec.

Appendix A. Mathematical framework of multiple instance learning

Let us consider a set X of bags (WSIs) $(x_i)_{1\leq i\leq n}$ such that each bag x_i is constituted of a set of k_i instances (tiles) $\{x_{i,1}, x_{i,2}, \cdots, x_{i,k_i}\}$ where instances are from a domain $\mathcal{D}$. In particular, bags can have a variable number of instances, or can share the same number of instances and, in that case, $\forall i, j, k_i = k_j$.

In its most general formulation, a MIL model m can be written as a combination of 3 modules:

1. An instance-embedder $f_{\theta_1} : \mathcal{D} \to \mathcal{E}$ embedding instances into a space $\mathcal{E}$

2. A pooling operator $g_{\theta_2} : \prod \mathcal{E} \to \mathcal{F}$ processing a set (of arbitrary size) of instance embeddings into a bag embedding

3. A bag classifier $h_{\theta_3} : \mathcal{F} \to \mathcal{Y}$ projecting a bag embedding

such that

$$\forall x_i \in X, m(x_i) = g_{\theta_3}\Big(h_{\theta_2}\big(f_{\theta_1}(x_{i,1}), f_{\theta_1}(x_{i,2}), \cdots, f_{\theta_1}(x_{i,k_i})\big)\Big) \in \mathcal{Y}$$

Examples of pooling functions are:

$$
\begin{aligned}
&\text{mean} && : x_i \mapsto \frac{1}{k_i}\sum_{k=1}^{k_i} x_{i,k} \\
&\text{max} && : x_i \mapsto \max\{x_1, \ldots, x_{k_i}\} \\
&\text{log-sum-exp (Ramon and De Raedt, 2000)} && : x_i \mapsto \frac{1}{M}\log\big(\sum_{k=1}^{k_i}\exp(M \times x_{i,k})\big) \\
&\text{attention (Ilse et al., 2018)} && : x_i \mapsto \sum_{k=1}^{k_i}\frac{\exp\left(w^\top \tanh(V x_{i,k}^\top)\right)}{\sum_{j=1}^{k_i}\exp\left(w^\top \tanh(V x_j^\top)\right)} \cdot x_{i,k} \\
&\text{gated-attention (Ilse et al., 2018)} && : x_i \mapsto \sum_{k=1}^{k_i}\frac{\exp\left(w^\top(\tanh(V x_{i,k}^\top) \odot \mathrm{sigm}(U x_{i,k}^\top))\right)}{\sum_{j=1}^{k_i}\exp\left(w^\top(\tanh(V x_j^\top) \odot \mathrm{sigm}(U x_j^\top))\right)} \cdot x_{i,k}
\end{aligned}
$$

where $a \in \mathbb{N}^*$, $r \in \mathbb{R}^*$, $M \in \mathbb{R}^+$, $V \in \mathbb{R}^{L \times \dim(\mathbb{H})}$, $U \in \mathbb{R}^{L \times \dim(\mathbb{H})}$, $w \in \mathbb{R}^{L \times 1}$, $L \in \mathbb{N}^*$ are parameters, $\odot$ is the elementwise multiplication, and sigm is the elementwise sigmoid function. The max operator can also be substituted or combined with the min operator. The log-sum-exp is also known as the softplus function and is considered as a smooth approximation to the max function. Attention-based approaches (Ilse et al., 2018) leverage an attention module formalized with a one hidden layer perceptron, that computes one score per input instance embedding which are then normalized such that they sum to 1, as to accommodate a potentially varying number of instances. All of these functions output a vector (or bag embedding) with the same shape as the k_i input vectors. It is possible to combine them in many ways such as to obtain output vectors of higher dimensions *e.g.* by using concatenation, summation, average or sequential combinations of themselves. These operators can be used for instance-based or embedding-based multiple instance learning.

Appendix B. Implementation details of the experimental validation

B.1 Epithelial classification on CRCHistoPhenotype

All methods shared the same training parameters as follows:

- The instance embedding model f_{θ_1} (Table 1) proposed in Sirinukunwattana et al. (2016) and used in Ilse et al. (2018) was employed.
- Loss function was binary cross-entropy
- Optimizer was the Adam (Kingma and Ba, 2014) with default momentum values β of 0.9 and 0.999, learning rate of $1e^{-4}$, weight decay of $5e^{-4}$, batch size of 1 for 100 epochs.
- Data augmentation consisted in the next functions in that order:
  1. Random vertical and horizontal flip.
  2. Random rotation.

3. H&E color augmentation (Ruifrok et al., 2001): H&E histopathological slides are originally uncoloured. The two stains Haematoxylin and Eosin are applied which respectively color nuclei and cytoplasm. Therefore, the true color space of H&E slides is made of the two vectors H and E rather than R, G, and B. Each tile was deconvoluted in the HE space using scikit-learn (Pedregosa et al., 2011) v0.24.2 (behavior changes depending on the version for the considered functions) with H vector value of $H = [0.650, 0.704, 0.286]^\top$ and E value of $E = [0.071, 0.994, 0.112]^\top$. Then, two independent random gaussian variables with mean 1 and standard deviation of 3 were sampled, and multiplied to H and E. These product of these multiplications were used to convert the tile from the (H, E, residual) space back to the RGB space.
4. Random crop of a 128 pixel-wide region.
5. Channel-wise standard scaling with RGB mean and standard deviation extracted from the training set.

Layer ID	Layer type	Layer parameters
1	Conv	Filter width 4, stride 1, padding 0, ReLU
2	Maxpool	Filter width 2, stride 2
3	Conv	Filter width 3, stride 1, padding 0, ReLU
4	Maxpool	Filter width 2, stride 2
5	Fully connected	512 neurons, ReLU
6	Dropout	0.25
7	Fully connected	512 neurons, ReLU
8	Dropout	0.25

Table 1: Tile embedding model f_{θ_1} from Sirinukunwattana et al. (2016) used in the CRCHistoPhenotype experiment.

For SparseConvMIL, the position of the center of each tile was used to build the sparse maps before applying spatial data augmentation consisting of random flips, rotations, and per axis scaling as detailed in section B. The sparse-input CNN was made of 2 convolutional layers of 12 channels, filter size 3, stride 1, activated with ReLU. An adaptive global average pooling layer converted the second layer sparse signal into a dense signal. The implementations of other methods are detailed in Table 2 and (Ilse et al., 2018).

All approaches are trained end-to-end. The 100 power fields were split into 55, 20, 24 samples for respectively the training, validation and testing set. The validation set is used to select the snapshot with least validation error for inference on the testing set. The training/testing process was performed 5 times for each method to derive confidence intervals.

B.2 Subtype classification on The Cancer Genome Atlas

All benchmarked methods used a ResNet34 architecture (He et al., 2016) pre-trained on Imagenet (Deng et al., 2009) as the instance embedding f_{θ_1}. To obtain embeddings instead of the probabilities output of ResNet34, the last classifier layer was removed, resulting in 512

Layer ID	Layer type	Layer parameters
1	Conv	Filter width 4, stride 1, padding 0, ReLU
2	Maxpool	Filter width 2, stride 2
3	Conv	Filter width 3, stride 1, padding 0, ReLU
4	Maxpool	Filter width 2, stride 2
5	Fully connected	512 neurons, ReLU
6	Dropout	0.25
7	Fully connected	512 neurons, ReLU
8	Dropout	0.25
9	max or	
9	mean or	
9	attention module or	
9	Sparse-input CNN	
10	Fully connected	1 output neuron, Sigmoid

Layer ID	Layer type	Layer parameters
1	Conv	Filter width 4, stride 1, padding 0, ReLU
2	Maxpool	Filter width 2, stride 2
3	Conv	Filter width 3, stride 1, padding 0, ReLU
4	Maxpool	Filter width 2, stride 2
5	Fully connected	512 neurons, ReLU
6	Dropout	0.25
7	Fully connected	512 neurons, ReLU
8	Dropout	0.25
9	Fully connected	1 output neuron, Sigmoid
10	Max-MIL/Mean-MIL	

Table 2: Complete models from the CRCHistoPhenotype experiment. The top table displays architectures for embedding-level approaches, while the bottom row displays architectures for instance-level approaches.

output channels per tile instead of 1 probability. All benchmarked approaches shared the same MIL classifier h_{θ_3} which was made of one 512-neurons ReLU activated fully connected layer followed by a 32-output fully connected layer (there are 32 classes). During training, 200 randomly cropped 128×128 pixel tiles were randomly sampled within each WSI. Hyper-parameters were shared across all benchmarked approaches and were:

- Loss function was binary cross-entropy.
- Optimizer was the Adaptive Momentum (Kingma and Ba, 2014) with default momentum values, learning rate of $1e^{-4}$, weight decay of $1e^{-4}$, batch size of 10 (or 2000 taking into account the number of tiles per WSI) for 200 epochs.
- Due to significant imbalance in the class distribution, oversampling was employed during training with frequencies equal to the inverse of the class counts.
- Data augmentation was the same as in the CRCHistoPhenotype experiment (see subsection B.1).

Project ID	Description	**Location**	**# WSI**
TCGA-ACC	Adrenocortical carcinoma	Adrenal gland	96
TCGA-BLCA	Bladder Urothelial Carcinoma	Bladder	298
TCGA-BRCA	Brain Lower Grade Glioma	Breast	1052
TCGA-CESC	Breast invasive carcinoma	Cervix	190
TCGA-CHOL	Cervical squamous cell carcinoma and endocervical adenocarcinoma	Bile ducts	46
TCGA-COAD	Cholangiocarcinoma	Colon	508
TCGA-DLBC	Colon adenocarcinoma	Lymph nodes	39
TCGA-ESCA	Esophageal carcinoma	Esophagus	130
TCGA-GBM	Glioblastoma multiforme	Brain	647
TCGA-HNSC	Head and Neck squamous cell carcinoma	Head and Neck	412
TCGA-KICH	Kidney Chromophobe	Kidney	104
TCGA-KIRC	Kidney renal clear cell carcinoma	Kidney	773
TCGA-KIRP	Kidney renal papillary cell carcinoma	Kidney	242
TCGA-LGG	Liver hepatocellular carcinoma	Brain	509
TCGA-LIHC	Lung adenocarcinoma	Liver	287
TCGA-LUAD	Lung squamous cell carcinoma	Lung	514
TCGA-LUSC	Lymphoid Neoplasm Diffuse Large B-cell Lymphoma	Lung	511
TCGA-MESO	Mesothelioma	Mesothelium	55
TCGA-OV	Ovarian serous cystadenocarcinoma	Ovary	477
TCGA-PAAD	Pancreatic adenocarcinoma	Pancreas	147
TCGA-PCPG	Pheochromocytoma and Paraganglioma	Adrenal gland	132
TCGA-PRAD	Prostate adenocarcinoma	Prostate	426
TCGA-READ	Rectum adenocarcinoma	Rectum	180
TCGA-SARC	Sarcoma	Soft tissues	292
TCGA-SKCM	Skin Cutaneous Melanoma	Skin	336
TCGA-STAD	Stomach adenocarcinoma	Stomach	383
TCGA-TGCT	Testicular Germ Cell Tumors	Testicular	138
TCGA-THCA	Thymoma	Thyroid	384
TCGA-THYM	Thyroid carcinoma	Thymus	105
TCGA-UCEC	Uterine Carcinosarcoma	Uterus	465
TCGA-UCS	Uterine Corpus Endometrial Carcinoma	Uterus	60
TCGA-UVM	Uveal Melanoma	Skin	62
Total	Vitually all solid cancer subtypes	Pan-location	10000

Table 3: Distribution of cancer subtypes (classes), locations, and number of WSI in the total cohort of 10000 slides involved in our experiments. The first column indicates the official TCGA project ids which groups all cases from the same cancer subtype. The second columns shows the location of each cancer subtype. The third displays the total number of WSI for each cancer subtype. The last line indicates the total of the cohort.

Cancer subtype (class)	**# training WSI**
lymphoid neoplasm diffuse large b-cell lymphoma	17
cholangiocarcinoma	20
mesothelioma	24
uterine carcinosarcoma	26
uveal melanoma	27
adrenocortical carcinoma	42
kidney chromophobe	46
thymoma	46
esophageal carcinoma	57
pheochromocytoma and paraganglioma	58
testicular germ cell tumors	61
pancreatic adenocarcinoma	65
rectum adenocarcinoma	79
cervical squamous cell carcinoma and endocervical adenocarcinoma	83

Table 4: Number of training whole slide images for the 14 cancer subtypes (classes) with less than 100 training samples. This table illustrates that some classes are heavily under-represented in the training set, which can challenge the accurate and efficient learning of features discriminative for subtype classification.

Additionally, SparseConvMIL and the graph-based approaches have the following data augmentation directly performed on tiles coordinates as follows (no effect on other approaches):

- random vertical and horizontal coordinates flips
- random coordinates rotation with an angle uniformly sampled within $[0, 2\pi]$
- random zoom for both x and y axes by sampling one value per axis in range $[0.7, 1.3]$

All approaches were trained end-to-end. For each method, the training process lasted approximately 1 week on 2 Nvidia V100. For fairness of comparisons, all approaches shared the same tile embedding function and classifier function: the only varying method was the pooling operator which can scale with the number of parameters for attention-based, graph-based and sparse-convolutional-based approaches but not for non-parametric approaches of max, mean and LSE.

References

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

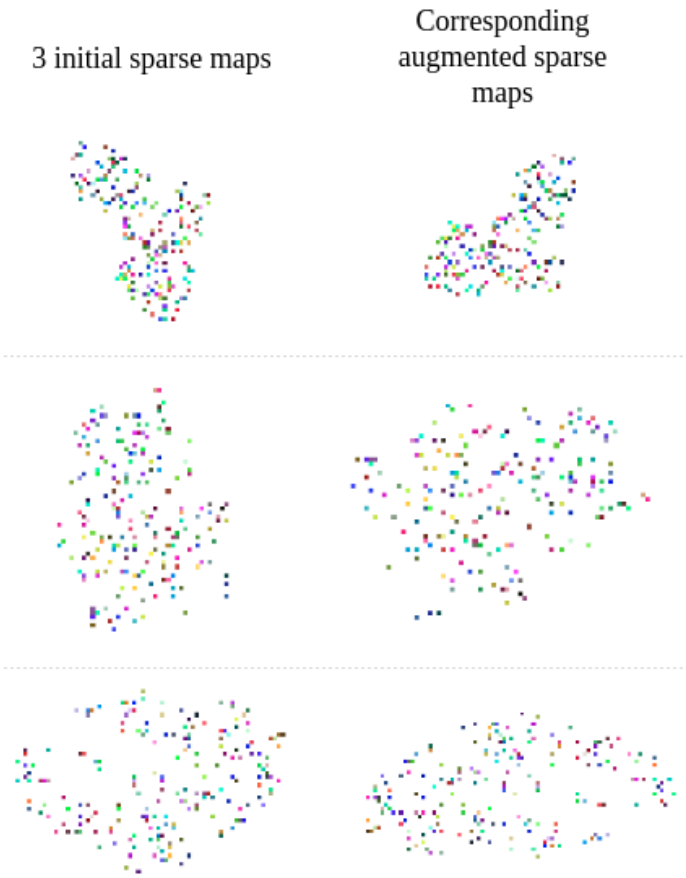

Figure 1: Illustration of SparseConvMIL specific data augmentation. 3 sparse maps are represented in the first column, 1 per line. For each sparse map, 300 512 pixel wide tiles were randomly sampled from the tissue section of WSI, and were spatially represented as coordinates within sparse maps. Each sparse map was spatially data augmented with random flips, rotations, scaling per axis and is displayed in the second column. Color for tiles is used for tracking purposes.

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

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Jan Ramon and Luc De Raedt. Multi instance neural networks. In *Proceedings of the ICML-2000 workshop on attribute-value and relational learning*, pages 53–60, 2000.

Arnout C Ruifrok, Dennis A Johnston, et al. Quantification of histochemical staining by color deconvolution. *Analytical and quantitative cytology and histology*, 23(4):291–299, 2001.

Korsuk Sirinukunwattana, Shan E Ahmed Raza, Yee-Wah Tsang, David RJ Snead, Ian A Cree, and Nasir M Rajpoot. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE transactions on medical imaging*, 35(5):1196–1206, 2016.