Multiple instance learning with pre-contextual knowledge

Andrea Grandi, Daniele Vellani
{275074,196186}@studenti.unimore.it

February 21, 2025

Abstract

The visual examination of histopathological images is a cornerstone of cancer diagnosis, requiring pathologists to analyze tissue sections across multiple magnifications to identify tumor cells and subtypes. However, existing attention-based Multiple Instance Learning (MIL) models for Whole Slide Image (WSI) analysis often neglect contextual and numerical features, resulting in limited interpretability and potential misclassifications. Furthermore, the original MIL formulation incorrectly assumes the patches of the same image to be independent, leading to a loss of spatial context as information flows through the network. Incorporating contextual knowledge into predictions is particularly important given the inclination for cancerous cells to form clusters and the presence of spatial indicators for tumors. To address these limitations, we propose an enhanced MIL framework that integrates pre-contextual numerical information derived from semantic segmentation. Specifically, our approach combines visual features with nuclei-level numerical attributes, such as cell density and morphological diversity, extracted using advanced segmentation tools like Cellpose. These enriched features are then fed into a modified BufferMIL model for WSI classification. We evaluate our method on detecting lymph node metastases (CAMELYON16 and TCGA lung).

1. Introduction

In recent years, computational pathology has emerged as a transformative tool for cancer research, leveraging Whole Slide Images (WSIs) to extract meaningful insights into tissue architecture and cellular composition. These large, high-resolution images are invaluable for diagnosing and prognosticating cancer, yet their sheer size, heterogeneity, and reliance on detailed annotations pose substantial challenges. One computational challenge is the large size of WSIs, of the order of 100,000 $\times$ 100,000 pixels. Processing images of such size with deep neural network directly is not possible with the GPUs commonly available. Overcoming this problem, previous work proposes to tessellate each WSI into thousands of smaller images called tiles and global survival prediction per slide is obtained in two steps. The tiles are first embedded into a space of lower dimension using a pre-trained feature extractor model, and a MIL model is trained to predict survival from the set of tiles embeddings of a WSI (Herrera et al., 2016) [1].

Multiple Instance Learning (MIL) has become a pivotal paradigm for WSI analysis. By treating a slide as a "bag" of smaller patches (instances), MIL allows slide-level predictions without the need for pixel-level annotations, streamlining the analysis pipeline (Ilse et al., 2018; Campanella et al., 2019) [2, 3]. Despite its utility, traditional MIL approaches often overlook critical contextual and numerical information that can enhance interpretability and predictive accuracy.

One limitation of MIL is the assumption that tiles from the same WSI are independent (Ilse et al., 2018) [2]. In particular, MIL models take into account only the visual knowledge comes from WSIs. In contrast, pathologists take into account also other aspects of WSIs in their analysis. Addressing these limitations requires innovative approaches capable of combining visual and numerical features from WSIs effectively (Litjens et al., 2017; Campanella et al., 2019) [3, 4].

In this work, we introduce a novel pipeline that integrates cutting-edge tools and methodologies to overcome these limitations. We preprocess WSIs using the CLAM framework (Lu et al., 2021) [5], ensuring the retention of essential visual features. To extract nuclei-specific numerical features such as cell counts and density, we utilize Cellpose (Stringer et al., 2021) [6], a state-of-the-art segmentation algorithm. Simultaneously, we employ DINO (Caron et al., 2021) [7], a self-supervised vision transformer, to generate embeddings representing the visual content of each patch. By concatenating these numerical and visual features, we construct a richer, more informative representation for each patch.

Our key innovation lies in adapting the BufferMIL (Bontempo et al, 2023) [8] framework to incorporate these enriched embeddings, enhancing interpretability through the extracted numerical features.

This paper is structured as follows: Section 2 reviews key advancements in MIL and its applications in computational pathology. Section 3 describes our methodology, detailing preprocessing, feature extraction, and the enhancements made to BufferMIL. Section 4 presents experimental results, discusses their implications, and outlines potential future directions. By combining numerical and visual features, our work seeks to advance computational pathology and provide deeper insights into the analysis of WSIs.

The source code is publicly available at `https://github.com/andrea-grandi/bio_project`.

2. Related Work

Multiple Instance Learning has revolutionized computational pathology by enabling efficient WSI classification

without exhaustive pixel-level annotations. Under MIL formulation, the prediction of a WSI label can come either directly from the tile predictions (instance-based) (Campanella et al.,2019) [3], or from a higher-level bag representation resulting from aggregation of the tile features (bag embedding-based) (Ilse et al., 2018) [2]. The bag embedding-based approach has empirically demonstrated superior performance (Sharma et al., 2021) [9]. Most recent bag embedding-based approaches employ attention mechanisms, which assign an attention score to every tile reflecting its relative contribution to the collective WSI-level representation. Attention scores enable the automatic localization of sub-regions of high diagnostic value in addition to informing the WSI level label.

One of the first important work in this field was DS-MIL (Li et al., 2021) [10]. This model utilizes a dual-stream framework, where patches are extracted from different magnifications (e.g., 5× and 20× in their study) of Whole Slide Images. These patches are processed separately for self-supervised contrastive learning. The embeddings obtained from patches at various resolutions are then concatenated to train the MIL aggregator, which assigns an importance or criticality score to each patch. The most critical patch is selected and compared to all others in a one-vs-all manner. This comparison uses a distance metric inspired by attention mechanisms, though it differs significantly by comparing two queries instead of the traditional key-query setup. Finally, the distances are aggregated to generate the final bag-level prediction.

Another work is BufferMIL, which is a notable framework that enhances MIL by incorporating explicit domain knowledge for histopathological image analysis, particularly addressing challenges like class imbalance and covariate shift. In this approach, a buffer is maintained to store the most representative instances from each disease-positive slide in the training set. An attention mechanism then compares all instances against this buffer to identify the most critical ones within a given slide. This strategy ensures that the model focuses on the most informative instances, thereby improving its generalization performance. By leveraging a buffer to track critical instances and employing an attention mechanism for comparison, BufferMIL effectively mitigates issues related to class imbalance and covariate shift. This approach enhances the model's ability to focus on the most informative patches within WSIs, leading to more accurate and reliable predictions in histopathological image analysis.

Building upon the attention-based methodologies of frameworks like BufferMIL, Context-Aware MIL (CAMIL) (Fourkioti et al., 2024) [11] extends the concept of informed instance selection by introducing neighbor-constrained attention mechanisms. CAMIL leverages spatial dependencies among WSI tiles to achieve superior performance in cancer subtyping and metastasis detection, showcasing the importance of spatial context in WSI analysis. Similarly, the Nuclei-Level Prior Knowledge Constrained MIL (NPKC-MIL) (Wang et al., 2024) [12] highlights the value of combining handcrafted nuclei-level features with deep learning, demonstrating improvements in interpretability and classification accuracy for breast cancer WSIs.

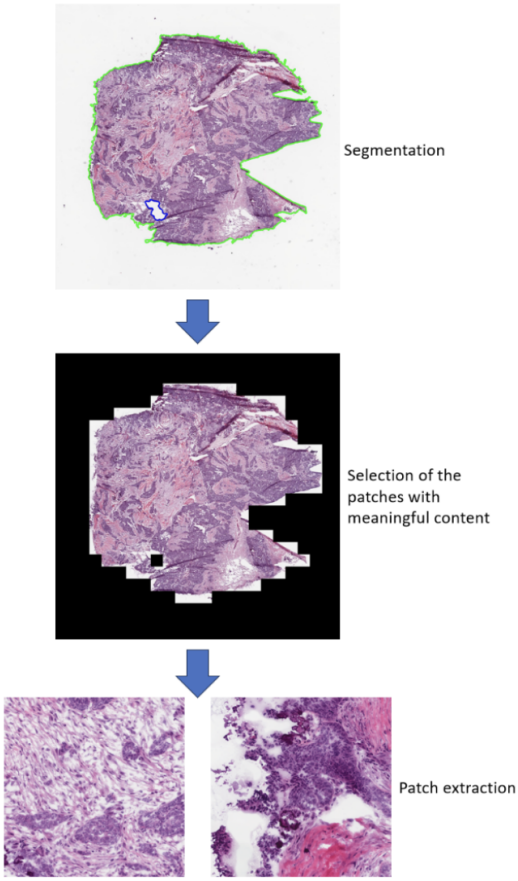


Figure 1: Whole Slide Image Preprocessing

3. Methods

In this section, we detail the methodology employed in our study, focusing on the integration of numerical and visual features into an enhanced MIL framework for WSI analysis.

3.1. Patch Extraction and Preprocessing

In our study, we employed the CLAM (Computational Pathology Learning and Analysis Methods) framework to efficiently extract patches from Whole Slide Images (WSIs) at a magnification of 20x. This magnification was chosen for its balance between detail and computational manageability, providing sufficient resolution for histopathological analysis. The extraction process involved several key steps as shown in Figure 1:

- **Patch Extraction with CLAM**: CLAM was used to divide the large WSIs into smaller, manageable patches. This framework is designed to handle the scale and complexity of WSIs by extracting patches at specified magnifications, in this case, 20x.
- **Otsu's Thresholding**: To segment the tissue areas from non-tissue regions within each patch, we applied Otsu's thresholding method. Otsu's algorithm automatically determines the optimal threshold value to separate the foreground (tissue) from the background,

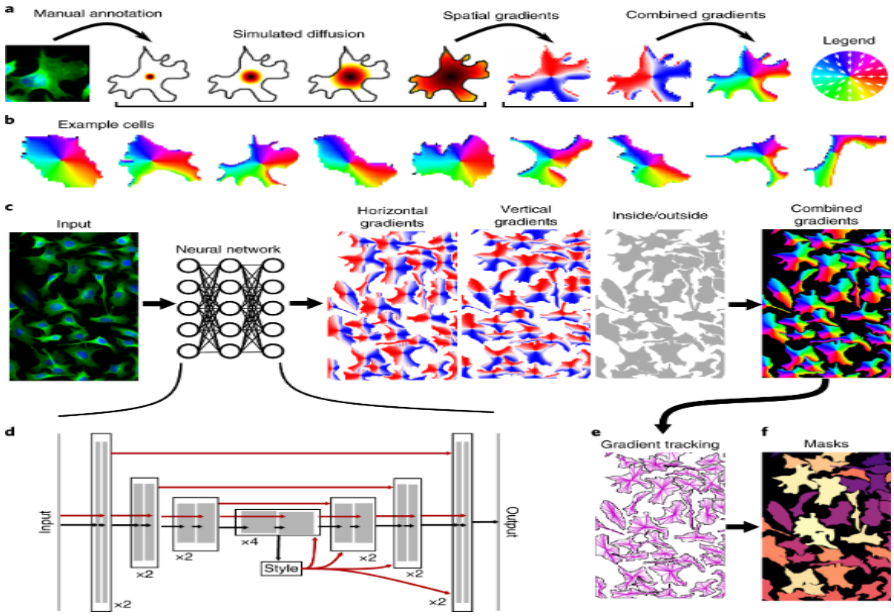


Figure 2: **Cellpose model architecture.** **a**, Procedure for transforming manually annotated masks into a vector flow representation that can be predicted by a neural network. A simulated diffusion process started at the center of the mask is used to derive spatial gradients that point towards the center of the cell, potentially indirectly around corners. The X and Y gradients are combined into a single normalized direction from 0 to 360. **b**, Example spatial flows for cells from the training dataset. **cd**, A neural network is trained to predict the horizontal and vertical flows, as well as whether a pixel belongs to any cell. The three predicted maps are combined into a flow field. **d** shows the details of the neural network which contains a standard backbone neural network that downsamples and then upsamples the feature maps, contains skip connections between layers of the same size, and global skip connections from the image styles, computed at the lowest resolution, to all the successive computations. **e**, At test time, the predicted flow fields are used to construct a dynamical system with fixed points whose basins of attraction represent the predicted masks. Informally, every pixel "follows the flows" along the predicted flow fields towards their eventual fixed point. **f**, All the pixels that converge to the same fixed point are assigned to the same mask.

based on the image's histogram. This step is crucial for focusing the analysis on relevant tissue regions and reducing noise from non-tissue areas.

- **Storage in .h5 Format**: The thresholded patches were stored in .h5 format by CLAM. This format is efficient for storing large datasets and includes the processed images along with any associated metadata.
- **Conversion to .jpg Format**: For compatibility with standard image processing pipelines and ease of use in downstream processing, we converted the .h5 files to .jpg format. This conversion ensures that the patches can be easily integrated into various image processing libraries and neural network models.

The choice of Otsu's thresholding was motivated by its effectiveness in segmenting histopathological images, while CLAM was selected for its efficiency in handling large WSIs and extracting patches at different magnifications. The conversion to .jpg format was necessary to maintain compatibility with widely used image processing tools, with minimal impact on the quality of the patches for feature extraction.

3.2. Feature Extraction

Our approach involves the extraction of both visual and numerical features from the patches.

3.2.1 Visual Feature Extraction with DINO

We utilize DINO (Data-Independent Neighborhood Occupancy) for visual embeddings because is particularly suited for this task due to its ability to capture rich visual information without requiring labeled data. The architecture of DINO is based on the ViT (Vision Transformer) (Dosovitskiy et al., 2021) [13], which processes images by dividing them into patches and passing them through a series of transformer encoder layers.

DINO enhances the self-supervised learning process by introducing teacher and student networks, where the teacher network provides pseudo-labels for the student network. This approach allows the model to learn robust representations by minimizing the distance between the predictions of the student and teacher networks.

To extract the visual embeddings, we first preprocessed the image patches by resizing them to a fixed resolution compatible with the DINO model. We then fed these patches into the pre-trained DINO model to obtain the embeddings from a specific layer, which were used as the primary input for our MIL model. These embeddings capture the intricate visual details within each patch, providing a robust representation for subsequent analysis.

3.2.2 Numerical Feature Extraction with Cellpose

To incorporate numerical features, we employed Cellpose to extract nuclei-level attributes from the patches. Cellpose is designed to segment cyto and nuclei in histopathological images with high accuracy, enabling the computation of numerical features such as cell density and morphological diversity.

As we can see in Figure 2, the segmentation process involves several steps. First, the image patches are preprocessed to enhance contrast and remove noise. Cellpose then applies a U-Net architecture (Ronneberger et al., 2015) [14] to predict cell boundaries and nuclei centers. Additionally, Cellpose predicts flow vectors, which are crucial for accurately segmenting overlapping or touching cells. These flow vectors represent the direction and magnitude from cell centers to the edges, aiding in the precise identification of individual cells.

In this work, we utilize the pretrained cyto3 model, as it is the most general and achieves the best performance on our dataset as we can see in Figure 3. Additionally, we explore Cellpose 2.0 (Stringer et al., 2021) [6] to train a custom model; however, it does not provide a significant improvement over the original model.

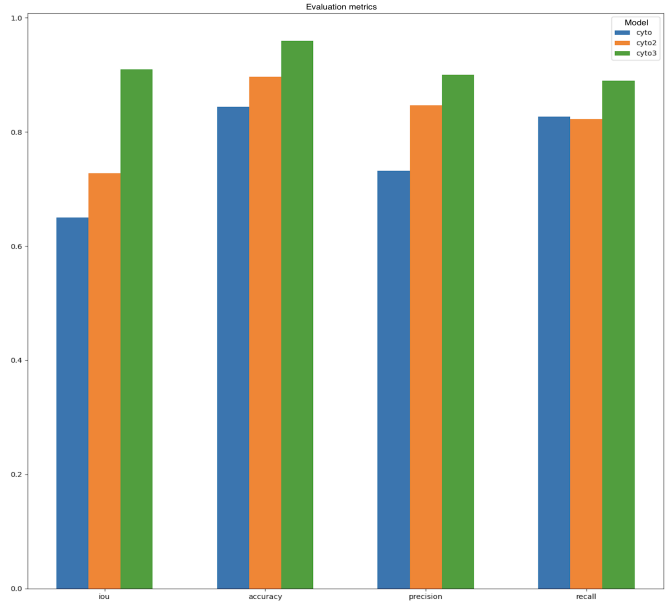


Figure 3: Cellpose models comparison

From these predictions, we extracted numerical features including cell density (number of cells per unit area), average nucleus area, and morphological diversity (measured using shape descriptors such as circularity and eccentricity). To further enhance the feature set, we derived features from the flow vectors, such as the mean and variance of flow directions and magnitudes within each patch. These flow-based features provide additional context about the cellular arrangement and organization.

We concatenated these numerical features with the visual embeddings from DINO to create a comprehensive representation of each patch, enhancing the discriminative power of our model. To ensure effective integration, we normalized the features, allowing them to contribute equally to the model's performance.

In summary, Cellpose not only segments cells but also provides flow vector information, which we leveraged to extract additional numerical features. This combined approach offers a more holistic representation of the cellular composition within each patch, complementing the visual information extracted from the images.

3.2.3 Geometry Dataset Conversion

To integrate the extracted features into the BufferMIL framework, we converted the data into a geometry dataset format, specifically into a DataBatch structure. This conversion is essential for ensuring compatibility with the input requirements of BufferMIL, which expects data in a specific format that includes both visual and numerical features.

The DataBatch structure organizes the data into batches, where each batch contains the concatenated features of multiple patches. We preprocessed the features by normalizing the numerical attributes to have zero mean and unit variance, ensuring that they are on a similar scale to the visual embeddings. We also ensured that the data is appropriately shuffled and split into training, validation, and test sets.

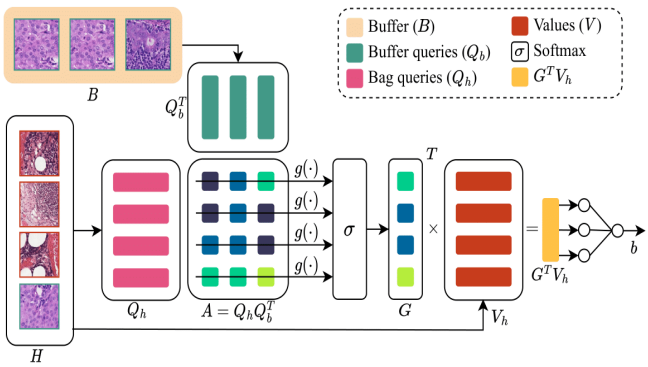


Figure 4: BufferMIL architecture

3.3. Buffer Adaptation

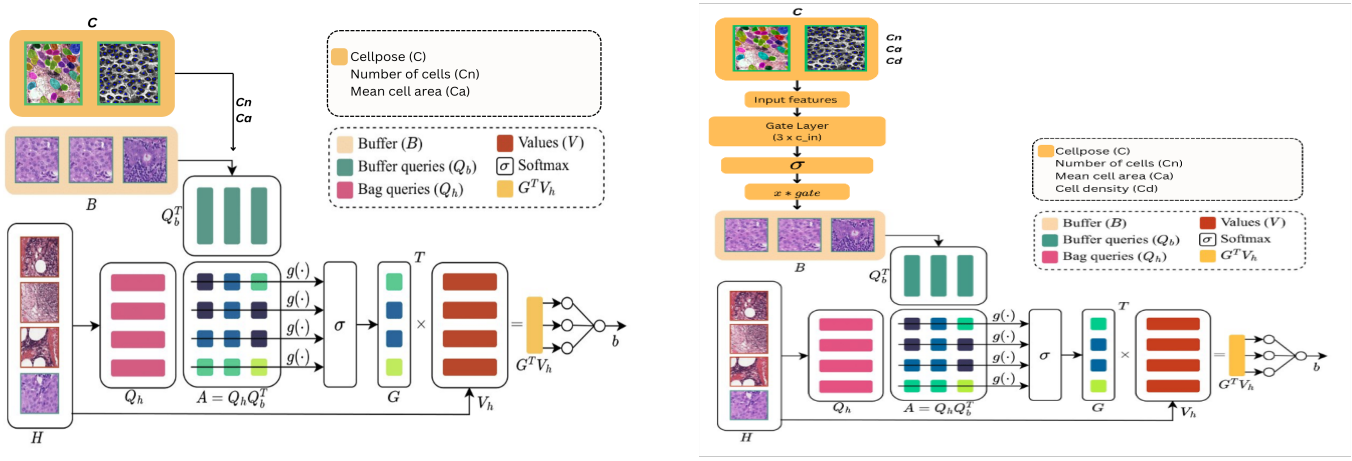
To adapt BufferMIL, particularly the buffer selection of critical patches, we have implemented an embedding concatenation approach before incorporating them into the attention matrix.

Let $A \in \mathbb{R}^{N \times N}$ be the original attention matrix used in the attention mechanism, where N represents the number of instances in a bag. We define the normalized morphological features as follows:

$$\tilde{C} = \frac{C - \min(C)}{\max(C) - \min(C)}, \tag{1}$$

$$\tilde{Ac} = \frac{Ac - \min(A_c)}{\max(A_c) - \min(A_c)}, \tag{2}$$

where C is the number of detected cells per patch and A_c is the mean cell area. The normalized versions, $\tilde{C}$ and $\tilde{A}_c$,



(a) Cellpose feature injection

(b) Cellpose feature gating

Figure 5: Our architectures

are then incorporated into the modified attention matrix using a weighted sum:

$$A' = w_1 A + w_2 \tilde{C} + w_3 \tilde{A}_c, \tag{3}$$

where w_1, w_2, w_3 are tunable hyperparameters that balance the contribution of the original attention matrix and the new morphological features.

Instead, the second approach follows a gate mechanism to integrate additional input features into the model. Specifically, we incorporate three features derived from the segmentation process (`num_cells`, `cell_density`, `mean_cell_area`), which are stacked together using `torch.stack`. These features provide valuable information regarding the segmentation, such as the number of cells, their density, and their average area.

The gating mechanism is implemented through a dedicated layer:

```
self.gate_layer = nn.Sequential(
    nn.Linear(3, self.c_in),
    nn.Sigmoid()
)
```

Mathematically, the gate function can be expressed as:

$$G(f) = \sigma(W_g f + b_g)$$

where $G(f)$ is the gate output, σ is the sigmoid activation function, W_g and b_g are the weight matrix and bias of the linear transformation, and f represents the input feature vector [num_cells, cell_density, mean_cell_area].

The output of the gate is then used to modulate x via an element-wise operation:

```
x = x * gate
```

which corresponds to the mathematical operation:

$$X' = X \odot G(f)$$

where X' is the modulated feature tensor, X is the original input tensor, and $\odot$ denotes element-wise multiplication.

Regarding its implications within the store buffer, when the forward method is invoked with `cellpose_feats` as input, the tensor x is modulated by the gating mechanism. The gated features are then used to compute the final predictions. Depending on the outcome (after applying a sigmoid function to part of the results), a decision is made on whether to store certain features in the buffer. Essentially, the gating mechanism enables the model to emphasize or down-weight specific characteristics relevant to the task before selecting which features to store in the buffer for inference.

This gating mechanism introduces a modulation process that allows external information (cellpose features) to be seamlessly integrated into feature extraction, ultimately influencing the selection of stored features for downstream inference.

3.4. Implementation

The implementation of our model follows a structured approach proposed in the BufferMIL paper with the additional extracted features. The key steps are:

1. **Model Initialization:** The model initializes the MIL layers using a fully connected layer (`FCLayer`) and a bag classifier (`BClassifierBuffer`). The MIL network is initialized with pretrained weights.

2. **Critical Instance Selection:** A patch-level classifier clspatch(·) is used to find the index of the most critical patch:

$$\begin{aligned} \text{crit} &= \arg\max\left(\text{cls}_{\text{patch}}(f(x))\right) \\ &= \arg\max\{W_p f(x_0), \ldots, W_p f(x_n)\} \end{aligned} \tag{4}$$

where W_p is a weight vector.

3. **Instance Embedding Aggregation:** Instance embeddings are aggregated into a single bag embedding by computing a linear projection into a query q_i and a value v_i using weight matrices W_q and W_v:

$$q_i = W_q h_i, \quad v_i = W_v h_i \tag{5}$$

4. **Attention-Based Scoring:** The query of the most critical instance, q_{crit}, is compared with all other queries using a distance measure $U(\cdot, \cdot)$:

$$U(h_i, h_{\text{crit}}) = \frac{\exp(\langle q_i, q_{\text{crit}} \rangle)}{\sum_{k=0}^{N-1} \exp(\langle q_k, q_{\text{crit}} \rangle)} \tag{6}$$

5. **Bag-Level Embedding:** The final bag score is computed as:

$$c_b(B) = W_b \sum_{i=0}^{N-1} U(h_i, h_{\text{crit}}) v_i \tag{7}$$

where W_b is a weight vector.

6. **Buffer Storage and Selection:** The buffer is updated every f_{req} epochs, selecting the top-k instances per slide.

7. **Final Bag Embedding Calculation:** The buffer is introduced in the attention mechanism. Given a bag $H = \{h_1, ..., h_N\}$ and buffer $B = \{b_1, ..., b_M\}$, the attention matrix A is computed:

$$A = Q_h Q_b^T \tag{8}$$

where Q_h and Q_b contain row-wise concatenated projections of H and B. An aggregation function $g(\cdot)$ is then applied to obtain a refined embedding:

$$G_i = g(\{A_{ij} : j \in [1, M]\}) \tag{9}$$

$$b = W_b G^T V_h \tag{10}$$

where G is computed using mean or max aggregation.

4. Experiments and Results

To demonstrate the performance of our method in capturing informative contextual knowledge, various experiments were performed on CAMELYON16 (Ehteshami Bejnordi et al., 2017) [15] and TCGA Lung. To measure our performance we used the following metrics:

- Accuracy
- Precision
- Recall
- AUC: area under the curve

We performed a great number of tests to find the best mix of inputs and hyperparameters, we tested:

- Different types of pretrained Cellpose models like cyto, cyto2, cyto3 and nuclei
- Apply the mean or max buffer aggregation
- Using different number of critical patches

The result in which we got the best performance out of our architecture was achieved using three inputs: "cyto3" for Cellpose model, "mean" buffer aggregation and select "10" critical patches to store in the buffer, with "10" ephocs buffer frequency update.

As shown in Table 1, the original BufferMIL model consistently outperforms our custom models across both CAMELYON16 and TCGA datasets.

- *CAMELYON16:* BufferMIL achieves an accuracy of 0.92 ± 0.09 and an AUC of 0.91 ± 0.02, while our custom model reaches 0.89 ± 0.01 in accuracy and 0.90 ± 0.012 in AUC.
- *TCGA:* BufferMIL obtains an accuracy of 0.85 ± 0.013 and an AUC of 0.86 ± 0.007, compared to 0.82 ± 0.09 in accuracy and 0.83 ± 0.011 in AUC for our custom model.

These results highlight the robustness of BufferMIL's architecture, particularly in effectively capturing relevant features from the dataset. Despite this, our custom models remains competitive, with only a marginal decrease in performance. This suggests that further optimization, such as fine-tuning hyperparameters, exploring advanced aggregation techniques, or integrating additional contextual features, could narrow the gap in performance.

Moreover, while BufferMIL shows superior performance, our custom models may offer advantages in terms of computational efficiency or adaptability to specific tasks, which can be further explored in future work.

In our best results, we trained for 200 epochs using an ADAM optimizer. The learning rate was set to 0.001 and halved after 10 epochs. Another important detail is that in the original architecture the classifier was trained using BCEWithLogitsLoss (like in the original BufferMIL), this loss combines a Sigmoid layer and the BCELoss in one single class. This version is more numerically stable than using a plain Sigmoid followed by a BCELoss as, by combining the operations into one layer, we take advantage of the log-sum-exp trick for numerical stability.

The unreduced (i.e., with reduction set to `none`) loss can be described as:

$$\begin{aligned} \ell(x, y) = L = \{l_1, \ldots, l_N\}^\top, \\ l_n = -w_n \left[y_n \cdot \log \sigma(x_n) + (1 - y_n) \cdot \log(1 - \sigma(x_n)) \right], \end{aligned}$$

where N is the batch size. If the reduction is not `none` (default is `mean`), then:

$$\ell(x, y) = \begin{cases} \text{mean}(L), & \text{if reduction} = \text{'mean'}; \\ \text{sum}(L), & \text{if reduction} = \text{'sum'}. \end{cases}$$

This is used for measuring the error of a reconstruction, for example in an autoencoder. Note that the targets $t[i]$ should be numbers between 0 and 1.

The model learned really well the distribution of the input training set, reaching a train accuracy as high as 0.93, the features learned from the training set are indeed well representative of our classification task, but this is not

Table 1: Performance comparison between our custom model and the original BufferMIL on CAMELYON16 and TCGA LUAD (lung) datasets. Metrics are reported as mean ± standard deviation.

Model	**CAMELYON16**		**TCGA**	
	Accuracy	**AUC**	**Accuracy**	**AUC**
Feature Injection	0.89785 ± 0.010	0.90209 ± 0.012	0.82606 ± 0.016	0.80345 ± 0.010
Feature Gating	0.84341 ± 0.013	0.87709 ± 0.012	0.82654 ± 0.011	0.83345 ± 0.009
BufferMIL	$\mathbf{0.92248} \pm 0.09$	$\mathbf{0.91645} \pm 0.02$	$\mathbf{0.85989} \pm 0.013$	$\mathbf{0.86643} \pm 0.007$

true when it comes to generalization, as our performance drops in the inference scenario. Moreover, we observe that, due to the scarcity of available data and the complexity of the disease, detecting cancer in hystopatological images seems to be a difficult case to study, as the performance results are consistently worse if methods are applied to this scenario.

5. Conclusions

In this paper, we explored an alternative approach to perform the integration of numerical, more comprehensive data and WSI images in order to improve the performance on the prediction of tumoral WSIs affected by breast and lung cancer. The results we obtained are comparable with other state-of-the-art methods, but they are heavily affected by severe problems related to the scarcity of data and abundance and noisiness of the features. The advantage of our architecture with respect to the others available lies in our unique approach to integrating a-priori knowledge, inside a Multiple Instance Learning architecture.

Moreover, the proposed model is easily configurable and adaptable to process different types of input data, also varying the number of modalities. Future developments of this architecture could focus on better feature selection methods for reducing redundancy and mitigating the impact of noisy data, which could significantly enhance model robustness and generalization capabilities. Additionally, exploring advanced data augmentation techniques could help address the issue of data scarcity, thereby improving the reliability of the model's predictions in real-world clinical scenarios.

We could also extend our project by introducing a Graph Neural Network (GNN) component that would increase the wisdom coming from interconnections between patches like DAS-MIL (Bontempo et al., 2023) [16]. This integration could enable the model to better capture spatial dependencies and contextual information inherent in WSIs, potentially leading to more accurate diagnostic insights.

Ultimately, our work lays the groundwork for future research in MIL data integration within the medical imaging field. By continuously refining the model architecture and leveraging emerging deep learning techniques, we aim to contribute to the development of more effective, interpretable, and clinically relevant diagnostic tools.

References

[1] Herrera, Francisco and Ventura, Sebastián and Bello, Rafael and Cornelis, Chris and Zafra, Amelia and Sánchez-Tarragó, Dánel and Vluymans, Sarah. *Multiple instance learning : foundations and algorithms*. Springer, 2016.

[2] Maximilian Ilse, Jakub M Tomczak, and Max Welling. Attention-based deep multiple instance learning. *arXiv preprint arXiv:1802.04712*, 2018.

[3] Gabriele Campanella, Matthew G Hanna, Liron Geneslaw, et al. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. *Nature medicine*, 25(8):1301–1309, 2019.

[4] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, et al. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[5] Ming Y Lu, Drew FK Williamson, Tiffany Y Chen, et al. Data-efficient and weakly supervised computational pathology on whole-slide images. *Nature Biomedical Engineering*, 5(6):555–570, 2021.

[6] Carsen Stringer, Tim Wang, Michael Michaelos, and Marius Pachitariu. Cellpose: a generalist algorithm for cellular segmentation. *Nature methods*, 18(1):100–106, 2021.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, et al. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

[8] Gianpaolo Bontempo, Luca Lumetti, Angelo Porrello, Federico Bolelli, Simone Calderara, and Elisa Ficarra. Buffer-mil: Robust multi-instance learning with a buffer-based approach. In *Image Analysis and Processing – ICIAP 2023: 22nd International Conference, ICIAP 2023, Udine, Italy, September 11–15, 2023, Proceedings, Part II*, page 1–12, Berlin, Heidelberg, 2023. Springer-Verlag.

[9] Yash Sharma, Aman Shrivastava, Lubaina Ehsan, Christopher A. Moskaluk, Sana Syed, and Donald E. Brown. Cluster-to-conquer: A framework for end-to-end multi-instance learning for whole slide image classification. In Mattias P. Heinrich, Qi Dou, Marleen de Bruijne, Jan Lellmann, Alexander Schlaefer, and Floris Ernst, editors, *MIDL*, volume 143 of *Proceedings of Machine Learning Research*, pages 682–698. PMLR, 2021.

[10] Bin Li, Yin Li, and Kevin W. Eliceiri. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning, 2021.

[11] Olga Fourkioti, Matt De Vries, Chen Jin, Daniel C. Alexander, and Chris Bakal. Camil: Context-aware multiple instance learning for cancer detection and subtyping in whole slide images, 2023.

[12] Xunping Wang and Wei Yuan. Nuclei-level prior knowledge constrained multiple instance learning for breast histopathology whole slide image classification. *iScience*, 27(6):109826, 2024.

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[15] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen A.W.M. van der Laak, Meyke Hermsen, Quirine F. Manson, Maschenka Balkenhol, Oscar Geessink, Nikolaos Stathonikos, Marcory C.R.F. Van Dijk, Peter Bult, Francisco Beca, Andrew H. Beck, Dayong Wang, Aditya Khosla, Rishab Gargeya, Humayun Irshad, Aoxiao Zhong, Qi Dou, Quanzheng Li, Hao Chen, Huang Jing Lin, Pheng Ann Heng, Christian Haß, Elia Bruni, Quincy Wong, Ugur Halici, Mustafa Ümit Öner, Rengul Cetin-Atalay, Matt Berseth, Vitali Khvatkov, Alexei Vylegzhanin, Oren Kraus, Muhammad Shaban, Nasir Rajpoot, Ruqayya Awan, Korsuk Sirinukunwattana, Talha Qaiser, Yee Wah Tsang, David Tellez, Jonas Annuscheit, Peter Hufnagl, Mira Valkonen, Kimmo Kartasalo, Leena Latonen, Pekka Ruusuvuori, Kaisa Liimatainen, and CAMELYON16 Consortium. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *JAMA Neurology*, 318(22):2199–2210, December 2017.

[16] Gianpaolo Bontempo, Angelo Porrello, Federico Bolelli, Simone Calderara, and Elisa Ficarra. Das-mil: Distilling across scales for mil classification of histological wsis. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023: 26th International Conference, Vancouver, BC, Canada, October 8–12, 2023, Proceedings, Part I*, page 248–258, Berlin, Heidelberg, 2023. Springer-Verlag.

A. Additional Plots

Here we provide additional visualizations supporting our analysis.

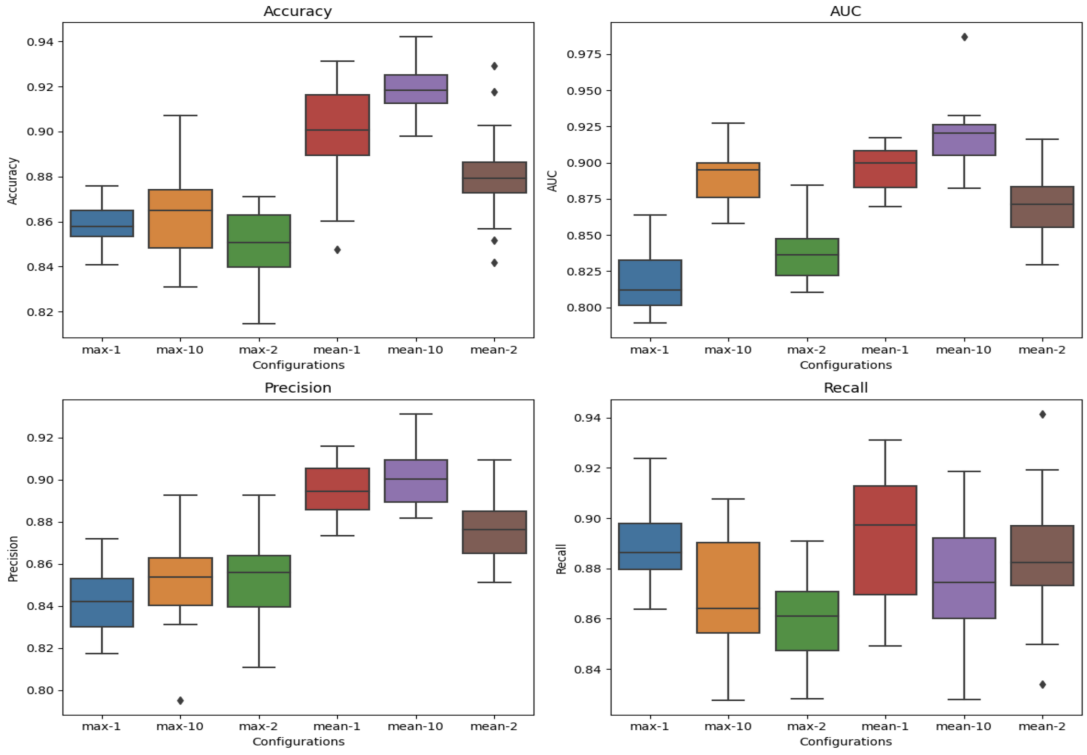


Figure 6: How our custom model (feature injection) works with different buffer aggregation techniques and different number of selected patches.

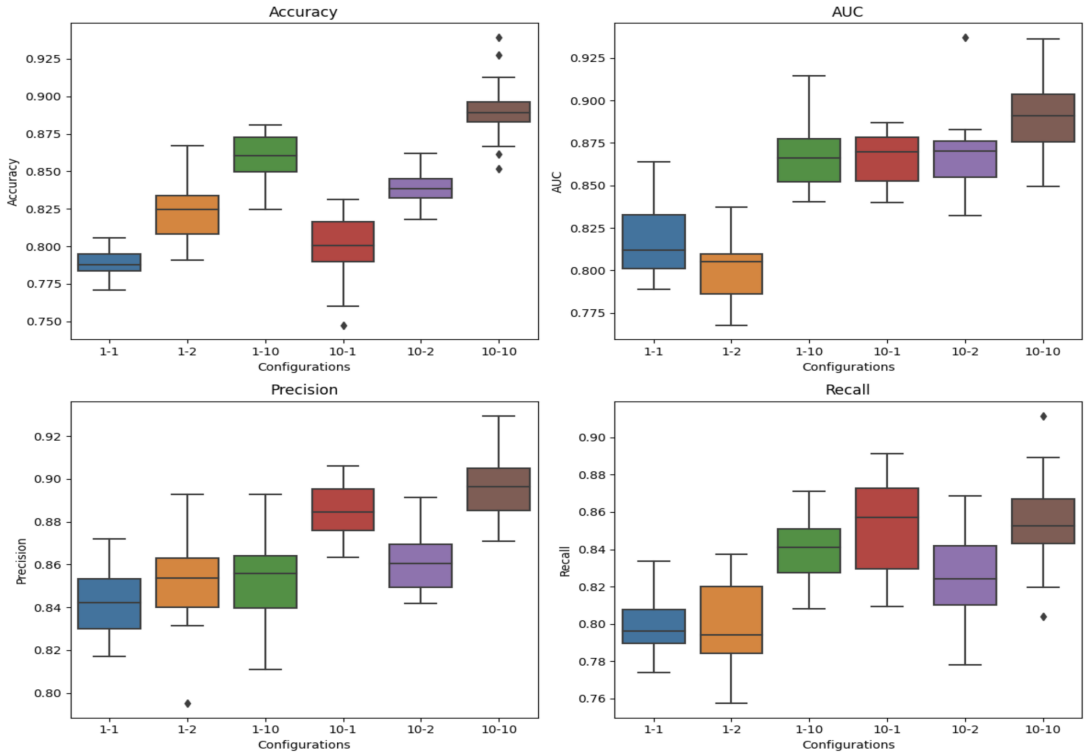


Figure 7: How our custom model (feature injection) works with different number of selected patches and the buffer update frequency.

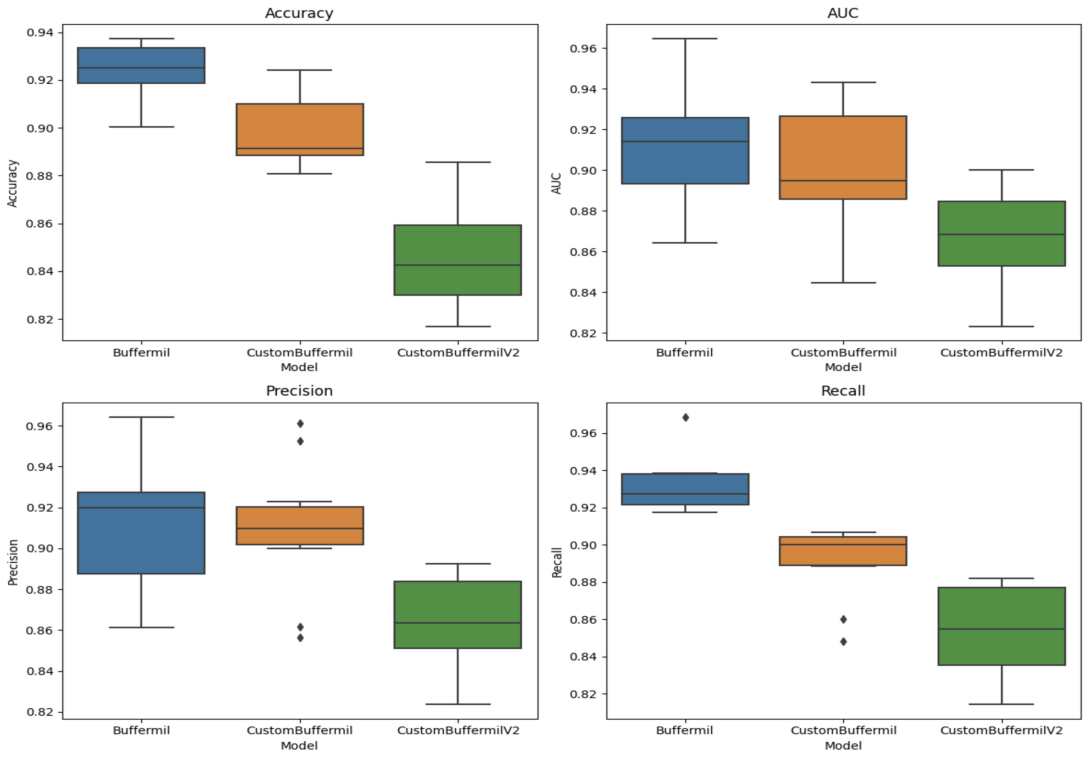


Figure 8: Original BufferMIL vs Feature injection vs Feature gating.