# torchmil: A PyTorch-based library for deep Multiple Instance Learning

F. M. Castro-Macías  F. J. Sáez-Maldonado
P. Morales-Álvarez  R. Molina

University of Granada, Spain

## Multiple Instance Learning (MIL)

**What is MIL?** It is a type of weakly supervised learning that is particularly useful when obtaining fine-grain annotations is expensive. It has applications in different areas, including medical imaging, drug repositioning, and video-event detection.

**The data:** pairs of the form $(\mathbf{X}, Y)$ where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times P}$ is a bag, and $\mathbf{x}_n \in \mathbb{R}^P$ are the instances. Usually, the instances have labels $\{y_1, \ldots, y_N\}$, but they are not observed. Only the bag label $Y$ is observed.

- In the binary case, both the instance labels and the bag labels are in $\{0, 1\}$, and it holds $Y = \max\{y_1, \ldots, y_N\} \in \{0, 1\}$.
- The bag may contain additional information, such as an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ representing the topological structure.

**At test time:** given a new bag, we want to predict the bag label, and the instance labels.

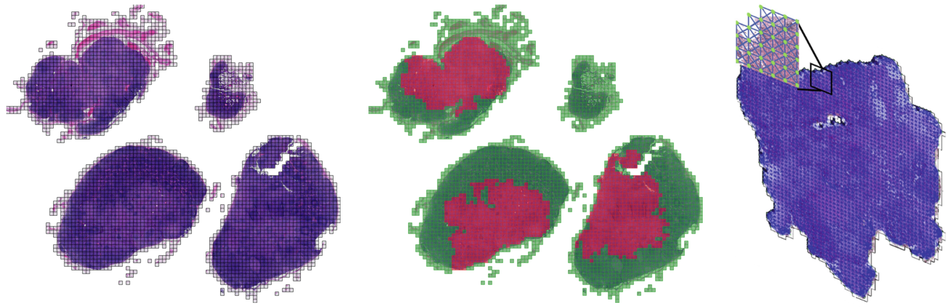

Figure 1. Whole Slide Image (WSI, bag), labeled patches (instances), and graph construction [4].

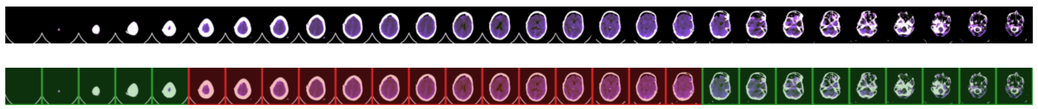

Figure 2. Computerized Tomography (CT) scan (bag) and labeled slices (instances).

## Challenges in modern MIL

**Models.** Diverse architectures capture local and global interactions – including Transformers, GNNs, and hybrid approaches [1, 2, 3, 4].

**Data.** High complexity; preprocessing critically affects performance, and in-memory representation impacts training efficiency.

**Evaluation and reproducibility.** Inconsistent evaluation protocols make it difficult to reproduce results across papers.

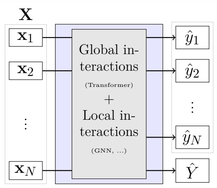


Figure 3. Architecture of deep MIL models.

### What if we could...

- **Standardize** common MIL workflows,
- Provide SOTA **models** through a **unified interface**,
- Offer ready-to-use **datasets** to enable reliable **benchmarking**?

## torchmil

**torchmil** is an open-source Python library built on top of PyTorch. It provides a flexible and extensible framework for building, training, and evaluating deep MIL models. Its is available via **pip** and we would love that you contribute to it!

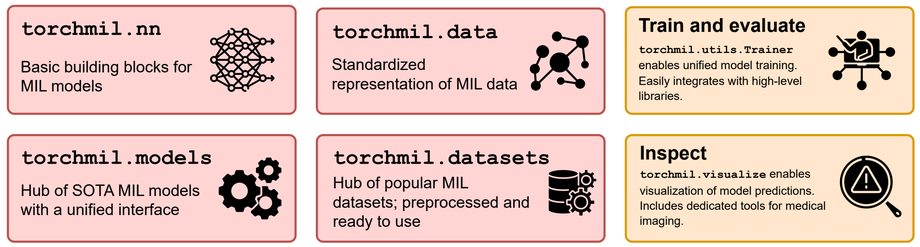


Figure 4. An overview of **torchmil**.

## Preliminary experiments

We validate the correctness of **torchmil**'s implementation of various MIL models. We consider the CAMELYON16 benchmark dataset, which involves detecting breast cancer metastases from WSIs.

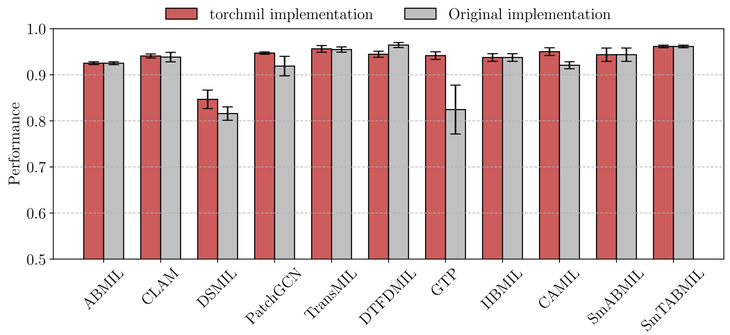


Figure 5. Performance of different MIL models in the CAMELYON16 classification task, measured as the mean of the accuracy, AUROC, and F1 metrics.

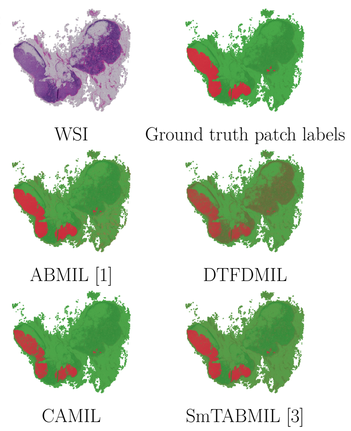


Figure 6. Attention maps generated with **torchmil.visualize**.

## References

[1] Ilse, M., Tomczak, J., and Welling, M. *Attention-based Deep Multiple Instance Learning*. ICML, 2018.

[2] Shao, Z., Bian, H., Chen, Y., Wang, Y., Zhang, J., Ji, X., *et al.* *TransMIL: Transformer-based Correlated Multiple Instance Learning for Whole Slide Image Classification*. NeurIPS, 2021.

[3] Castro-Macías, F. M., Morales-Álvarez, P., Wu, Y., Molina, R., Katsaggelos, A. K. *Sm: Enhanced Localization in Multiple Instance Learning for Medical Imaging Classification*. NeurIPS, 2024.

[4] Chen, R. J., Lu, M. Y., Shaban, M., Chen, C., Chen, T. Y., *et al.* *Whole Slide Images Are 2D Point Clouds: Context-Aware Survival Prediction Using Patch-Based Graph Convolutional Networks*. MICCAI, 2021.

## Check it out!

github.com/Franblueee/torchmil



fcastro@ugr.es