Semi-supervised Segmentation of Histopathology Images with Noise-Aware Topological Consistency — Supplementary Material —

Meilong Xu[1], Xiaoling Hu[2], Saumya Gupta[1], Shahira Abousamra[1], and Chao Chen[1]

[1] Stony Brook University, Stony Brook, NY, USA
[2] Athinoula A. Martinos Center for Biomedical Imaging, Massachusetts General Hospital and Harvard Medical School, MA, USA
meixu@cs.stonybrook.edu

In the supplementary material, we begin with notations for foreground and background in Sec. 6, followed by a detailed introduction to persistent homology in Sec. 7. Then, we describe the correspondence between persistent dots and the likelihood map in Sec. 8. Next, we discuss the differentiability of the noise-aware topological consistency loss in Sec. 9. In Sec. 10, we provide detailed descriptions of the datasets, followed by implementation details in Sec. 11. We also provide the reference of our baselines in Sec. 12. In Sec. 13, we describe the evaluation metrics in detail. More qualitative results are given in Sec. 14. Finally, additional ablation studies and results are provided in Sec. 15.

6 Notes on Foreground and Background

Here, we provide some notations about foreground and background in our paper. Our algorithm uses black as the foreground and white as the background as can be seen in Fig. 2- Fig. 3 of the main paper and Fig. 6 of the Supplementary. For better visualization, however, we display the segmentation results and ground truth with white as the foreground in Fig. 1 and Fig. 5 of the main paper and Fig. 7 of the Supplementary.

7 Background: Persistent Homology

In algebraic topology [36], *homology classes* account for topological structures in all dimensions. 0-, 1-, and 2-dimensional structures describe connected components, loops/holes, and cavities/voids, respectively. For binary images, the number of d-dimensional topological structures is called the *d-dimensional Betti number*, β_d.[3] Despite the well-understood topological space for a binary image, the theory does not directly extend to real-world scenarios with continuous, noisy data. For example, in image analysis, we need a principled tool to reason about

[3] Technically, β_d counts the dimension of the d-dimensional homology group. The number of distinct homology classes/topological structures is exponential to β_d.

the topology from a continuous likelihood map. To bridge this gap, the theory of *persistent homology* was invented in the early 2000s [9].

Persistent homology has emerged as a powerful tool for analyzing the topology of various kinds of real-world data, including images. In the image segmentation task, we apply persistent homology to the likelihood map of a deep neural network to reason about its topology. Given an image in the 2D domain $I \subseteq \mathbb{R}^2$, we use a network to generate a likelihood map f. The segmentation map is obtained by thresholding f at a certain threshold c (usually 0.5). We define a *sublevel set*: $S_c := \{(m,n) \in I \mid f(m,n) \leq c\}$. With all different threshold values sorted in an increasing order ($c_1 < c_2 < \cdots < c_n$), we obtain a filtration, i.e., a series of growing sublevel sets: $\varnothing \subseteq S_{c_1} \subseteq S_{c_2} \subseteq ... \subseteq S_{c_n} = I$. As the threshold c increases, topology of the sublevel set changes. New topological structures appear while old ones disappear.

Persistent homology tracks the evolution of all topological structures, such as connected components and loops. All the topological structures and their birth/death times are captured in a so-called *persistence diagram*, providing a multi-scale topological representation (See Fig. 2).

A persistence diagram (PD) consists of multiple dots in a 2-dimensional plane. These dots are called *persistent dots*. Given a continuous-valued likelihood map function f, we have its persistence diagram $Dgm(f)$. Each persistent dot $p \in Dgm(f)$ represents a topological structure. Its two coordinates denote the birth and death filtration values for the corresponding topological structure, i.e., $p = (b, d)$, where $b = birth(p)$ and $d = death(p)$. We can calculate the persistence diagrams for outputs of both the student and the teacher models, in order to compare the two likelihood maps from a topological perspective.

8 Mapping Persistent Dots to the Likelihood

In Fig. 6, we show how persistent dots in the persistence diagram can ultimately be mapped to pixels/voxels in the likelihood map. Consequently, the loss functions defined in Eq. (4)- Eq. (8) of the main paper are differentiable: the penalty applied to the persistent dots is ultimately a penalty on the pixels/voxels of the likelihood. Hence backpropagation can take place: our proposed losses are differentiable.

In Fig. 6, we give an example of a likelihood f in Fig. 6(a), and focus on the **orange** persistent dot in Fig. 6(b); let us call it p. It's coordinate in the persistence diagram $Dgm(f)$ is nothing but its birth b and death d given by $(b, d) = (0.42, 0.46)$.

There are precisely two pixels in the likelihood that capture the lifetime of this persistent dot p. We call them *critical* pixels. We denote the location of these critical pixels in f using black arrows in Fig. 6(a). These two critical pixels have the values 0.42 and 0.46 respectively. We now map the likelihood to the persistence diagram below.

In the filtration Fig. 6(c), when the threshold is 0.42, the critical pixel of the same value gets included into the binary map. It is a connected component on its

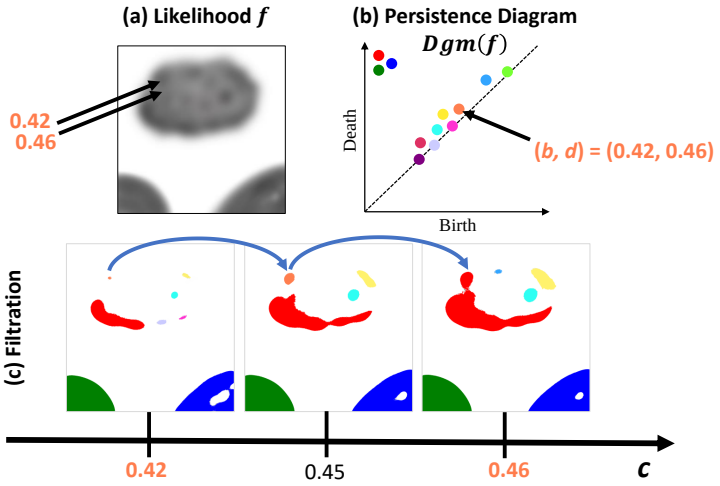


Fig. 6: **(a)** A predicted likelihood map f, and **(b)** the corresponding persistence diagram $Dgm(f)$. Consider the **orange** persistent dot having birth b and death d times as $(b, d) = (0.42, 0.46)$. We show the corresponding filtration in **(c)** for these specific birth/death times. At birth $b = 0.42$, the connected component corresponding to the **orange** is born. At death $d = 0.46$, this connected component dies as it gets absorbed into the older **red** connected component. Note that we only show 0-dim persistent dots pertaining to connected components in $Dgm(f)$.

own and is denoted by **orange** in Fig. 6(c) when $c = 0.42$. This marks the *birth* of the connected component corresponding to the persistent dot p. At threshold $c = 0.45$, we see this **orange** connected component grows larger as more pixels get introduced into the binary map. Finally, at $c = 0.46$, the second critical pixel is introduced which joins the **orange** connected component to the older **red** connected component. This marks the *death* of the connected component corresponding to p as it gets absorbed into the older **red** connected component. Hence, the persistent dot $p \in Dgm(f)$'s birth and death values each correspond to a single pixel location in the likelihood f.

Now, this persistent dot gets matched to the diagonal according to the bijection γ^* introduced in Sec. 3.3. Consequently, the loss described in Eq. (7) pushes p towards the diagonal. This means p is a noisy structure and we would like to suppress/remove it. On pushing it to the diagonal, we force the birth and death times to be the same: the moment this structure is born, it should be automatically included in the older connected component. Hence it ceases to exist as a standalone connected component across any and all filtration values and is thus effectively removed as noise.

9 Differentiability of the Topology-Aware Losses.

Both $\mathcal{L}^U_{\text{topo-cons}}$ and $\mathcal{L}^U_{\text{topo-rem}}$ are differentiable, as Eq. (5) and Eq. (7) are both written as polynomials of the likelihood map f_s at certain critical pixels. Here it is crucial to assume the critical pixels, x_p^b and x_p^d, remain constant locally. This is because the likelihood map is a piecewise linear function determined by the function values at a discrete set of pixel locations. Assuming without loss of generality that all pixels have distinct values, we can show that within a small neighborhood of the likelihood f_s, the order of all pixels in f_s remains the same. Therefore, the algorithmic computation of persistent homology will associate the same set of critical pixels with each persistent dot x in the diagram. In other words, we can assume x_p^b and x_p^d remain constant.

10 Details of the Datasets

1. **Colorectal Adenocarcinoma Gland (CRAG)** [13] is a collection of 213 H&E stained colorectal adenocarcinoma image tiles captured at 20× magnification, with full instance-level annotation. Most of the images are of the size 1512×1516. It is officially divided into a training set with 173 samples and a test set with 40 samples. In our experiments, we separate the training set into 153 images for training and 20 images for validation. For 10% and 20% labeled data splitting, we randomly select 16 and 31 images with labels respectively, for training.
2. **Gland Segmentation in Colon Histology Images Challenge (GlaS)** is introduced in [42] and comprises of 165 images derived from 16 H&E stained histological sections of stage T3 or T4 colorectal adenocarcinoma. The dataset is officially separated into a training set with 85 samples and a test set with 80 samples. In our experiments, we divide the training set into 68 images for training and 17 images for validation. For 10% and 20% labeled data splitting, we randomly select 7 and 14 images with labels for training.
3. **Multi-Organ Nuclei Segmentation (MoNuSeg)** [28] contains 44 H&E stained images of size 1000×1000 from seven organs. It consists of two sets, 30 images containing $21,623$ nuclei for training and 14 images for testing. In our experiments, we choose 20% training data (6 images) as the validation set, and for 10% and 20% labeled data splitting, we randomly select 3 and 5 images with labels respectively for training.

11 Implementation Details

We train our model in two stages. The first stage is pre-training, using only $\mathcal{L}^S$ and $\mathcal{L}^U_{pixel}$ to train the network for several iterations. For CRAG and GlaS, we pre-train the model for 12000 iterations; for MoNuSeg, we pre-train the model for 2000 iterations. The second stage is fine-tuning using our topological consistency loss. We fine-tune the model for 500 epochs using Eq. (1) as the overall training

objective. While training, we use UNet++ [65] as our backbone for both student and teacher networks, and we adopt the Adam optimizer solver to train the model. The proposed algorithm is implemented on the PyTorch platform. The training hyper-parameters are set as follows: for CRAG and GlaS, the batch size is 16, and the learning rate is $5e-4$. For MoNuSeg, the batch size is 8, and the learning rate is $1e-4$. We first apply random cropping on both labeled and unlabeled data. The cropping size is 256×256 for CRAG and GlaS and 416×416 for MoNuSeg. After random cropping, we apply random rotation and flipping for weak augmentations, and for strong augmentations, we apply color change and morphological shift. The EMA decay rate α and λ_2^U are set to 0.999 and 0.002 respectively. Introduced in [31], the weight factor of pixel-wise consistency loss is calculated by the Gaussian ramp-up function $\lambda_1^U = k * e^{-5*(1-\frac{\tau}{T})^2}$, where $k = 0.1$ and T is the total number of iterations. λ_1^L and λ_2^L in $\mathcal{L}^S$ are all set to 0.5. The persistence threshold ϕ for decomposing the persistence diagrams is 0.7. All the experiments are conducted on an NVIDIA RTX $A6000$ GPU with 48 GB RAM.

12 Baseline Reference

In our experiments, some baselines are based on the implementations of publicly available repositories. Here, we provide our baselines' source for reference and appreciate their efforts on the public code.

MT [46], EM [48], UA-MT [60], and URPC [33] are based on the implementations from: https://github.com/HiLab-git/SSL4MIS.

XNet [64] is based on the implementations from: https://github.com/guspantanadi/XNetfromYanfeng-Zhou.

CCT [37] is based on the implementations from:
https://github.com/yassouali/CCT.

HCE [25] is implemented by ourselves due to the lack of code.

13 Evaluation Metrics

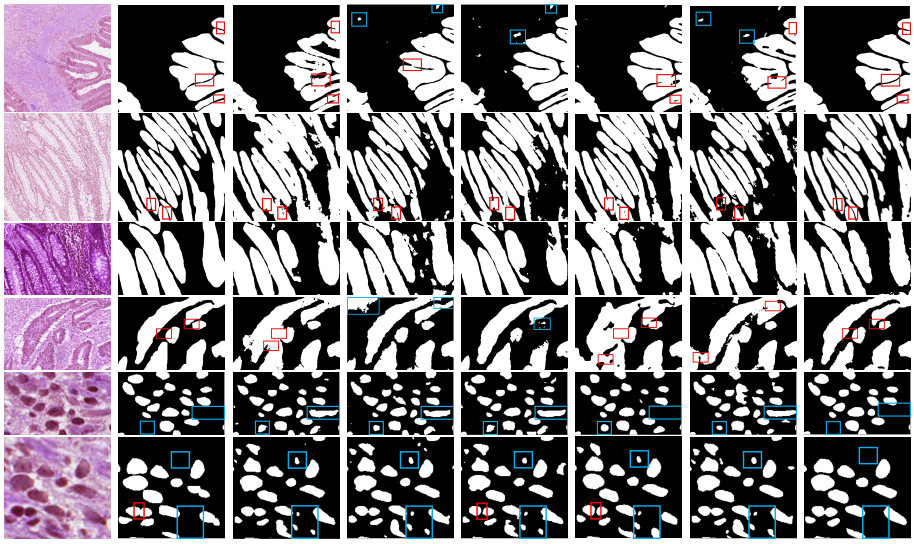
We select three widely used pixel-wise evaluation metrics, **Object-level Dice coefficient (Dice_Obj)** [55], **Intersection over Union (IoU)** and **Pixel-wise accuracy**. Object-level Dice coefficient mainly measures the similarity between two segmented objects, and this is especially useful in pathology imaging, where accurately segmenting individual anatomical structures is crucial. IoU provides a measure of how well the predicted segmentation or detected object aligns with the ground truth. Pixel-wise accuracy evaluates how many pixels in the segmentation maps are correctly classified. The larger these three metrics are, the better the segmentation performance is.

Topology-relevant metrics mainly measure the structural accuracy. We also select three topological evaluation metrics, **Betti Error** [18], **Betti Matching Error** [44], and **Variation of Information (VOI)** [34]. For the Betti error,

we split the prediction and ground truth into patches in a sliding-window fashion and calculate the average absolute discrepancy between their 0-dimensional Betti number. The size of the window is 256×256. Betti matching error considers the spatial location of the features within their respective images and can be regarded as a variant of Betti error. VOI mainly measures the distance between two clusterings. The smaller these metrics are, the better the segmentation performance is.

14 Additional Qualitative Results

Here, we provide more qualitative results in Fig. 7 further to verify the effectiveness and superiority of our proposed method.



(a) Image **(b)** GT **(c)** MT **(d)** EM **(e)** UAMT **(f)** HCE **(g)** URPC **(h)** Ours

Fig. 7: Additional qualitative results. The **red** boxes indicate the regions that are prone to topological errors such as incorrect merging or separating adjacent glands; the **blue** boxes indicate false positive gland predictions or missing glands. Rows 1-2: CRAG. Rows 3-4: GlaS. Rows 5-6: MoNuSeg. Zoom in for better views.

15 Additional Ablation Study

In this section, we conduct additional ablation studies to further demonstrate the effectiveness of our TopoSemiSeg.

Ablation study on both $\mathcal{L}_{\text{topo}}^{\text{S}}$ and $\mathcal{L}_{\text{topo}}^{\text{U}}$. To further validate the effectiveness of our method, we conduct additional experiments on 100% and 20%

labeled data with supervised and unsupervised topological constraints. $\mathcal{L}^{\mathrm{S}}_{\mathrm{topo}}$ and $\mathcal{L}^{\mathrm{U}}_{\mathrm{topo}}$ resp. denote the topology-based loss for labeled and unlabeled data (λ^{S}_{topo} and λ^{U}_{topo} are resp. weights). The results are shown in Tab. 6. In the top half of Tab. 6, most of the topology-wise performance has improved, with a slight loss in pixel-level performance. Also, with sufficient labeled data, only adding the $\mathcal{L}^{\mathrm{S}}_{\mathrm{topo}}$ performs better. In the bottom half of Tab. 6, applying $\mathcal{L}^{\mathrm{S}}_{\mathrm{topo}}$ and $\mathcal{L}^{\mathrm{U}}_{\mathrm{topo}}$ simultaneously gives mixed results on the metrics, without any significant change in overall performance. It can also be seen that the method we proposed makes good use of unlabeled data.

Table 6: Ablation study on $\mathcal{L}^{\mathbf{S}}_{\mathbf{topo}}$ and $\mathcal{L}^{\mathbf{U}}_{\mathbf{topo}}$.

Labeled Ratio (%)	λ^{S}_{topo}	λ^{U}_{topo}	Dice_Obj ↑	BE ↓	BME ↓	VOI ↓
100%	0	0	**0.928**	0.149	5.650	0.547
100%	0.002	0	0.913	**0.141**	**5.150**	**0.532**
100%	0	0.002	0.912	0.146	5.178	0.539
100%	0.002	0.002	0.922	0.153	5.239	0.543
20%	0.001	0.001	0.893	0.218	12.850	0.727
20%	0.002	0.002	0.895	**0.189**	8.725	0.723
20%	0.005	0.005	0.876	0.246	10.825	0.787
20%	0	0.002	**0.898**	0.226	**8.575**	**0.709**

Ablation Study on EMA decay α. The EMA decay α plays an important role in the teacher-student framework where it provides a smoothing effect over the parameters of the model. A higher decay (closer to 1) gives more weight to the historical parameters, leading to a more stable representation of the student model's knowledge over time. However, too high EMA decay may result in the teacher model lagging too far behind the student model due to the rapidly changing or non-stationary environments, failing to capture the latest patterns of the data. So to verify the effectiveness of our selected α, we conduct an ablation study on EMA decay. The results are shown in Tab. 7. From the results we can see when $\alpha = 0.999$, our model performs the best.

Table 7: Ablation study on EMA decay α.

α	Dice_Obj ↑	BE ↓	BME ↓	VOI ↓
0.99	0.887	0.249	11.525	0.734
0.999	**0.898**	**0.226**	**8.575**	**0.709**
0.9999	0.873	0.252	11.850	0.752

Ablation study on data augmentation. Our method relies on the assumption that, for the model to be robust, its predictions should not change significantly for small perturbations of the input data in terms of topology. Hence, data augmentation and its hyper-parameter selections are crucial for our method. In Tab. 8, we report the results of the ablation study on data augmentations, and

in the last row we also report using strong augmentations on labeled data. The *italicized numbers* are our selected hyper-parameters. We conduct experiments on hyper-parameters of strong augmentations, specifically, brightness and contrast. We provide results on several combinations of hyper-parameter values. The results indicate that our method is robust to the choice of data augmentations' hyper-parameters.

Table 8: Ablation study on data augmentations.

Brightness	Contrast	Dice_Obj ↑	BE ↓	BME ↓	VOI ↓
0.3	*0.1*	0.898	**0.226**	8.575	**0.709**
0.3	0.5	0.897	0.233	**8.000**	0.720
0.1	0.1	0.887	0.255	11.550	0.736
0.5	0.1	**0.900**	0.227	8.237	0.715
strong aug. on labeled data		0.883	0.238	8.025	0.717

Ablation study on labeled sampling bias & Retain noise and remove signal. Here, we conduct the experiments to alleviate the potential sampling bias and report the results in Tab. 9.. On GlaS dataset, 20% labeled samples do perform better than 10%. In addition, we provide the results that we retain the noise part and remove the signal part in the last 2 rows of Tab. 9. As expected, the performance drops significantly. Removing the signal dots causes the prediction to intentionally overlook the true structures while retaining the noisy dots causes it to include erroneous structures. This result, together with our ablation study (Tab. 5), shows how our signal/noise decomposition helps the model learn even without GT annotation.

Table 9: The first 2 rows: the results that rerun the experiments 5 times with different labeled training samples on GlaS dataset. The last 2 rows: the ablation study on retaining the noise and removing the signal.

Method	Dice_obj ↑	BME ↓
Ours (10%)	0.876±0.035	9.885±0.825
Ours (20%)	**0.893±0.007**	**9.384±0.479**
Noise✓Signal ✗	0.866	21.325
Ours	**0.898**	**8.575**

Consistent Comparisons. To ensure a consistent comparison, we add the results of XNet [64] for MoNuSeg dataset, CCT [37] for CRAG and GlaS dataset and FixMatch [43] for CRAG dataset in Tab. 10. Our method consistently outperforms these 3 methods. Noted that FixMatch simply selects trustworthy pseudo-labels by thresholding the classifier's confidence. Many samples are discarded. Instead, we use all images, using persistence thresholding to select true topology signal to learn (with theoretical and empirical guarantees).

Table 10: The results of XNet, CCT and FixMatch.

Dataset	Labeled Ratio (%)	Method	Dice_Obj ↑	BE ↓	BME ↓	VOI ↓
CRAG	10%	CCT	0.853	1.954	40.210	0.864
		Ours	**0.884**	**0.227**	**10.475**	**0.758**
	20%	CCT	0.872	1.262	25.420	0.773
		FixMatch	0.868	1.706	30.680	0.855
		Ours	**0.898**	**0.226**	**8.575**	**0.709**
GlaS	10%	CCT	0.864	0.862	16.645	0.932
		Ours	**0.878**	**0.551**	**8.300**	**0.811**
	20%	CCT	0.876	0.761	13.125	0.834
		Ours	**0.895**	**0.510**	**9.825**	**0.808**
MoNuSeg	10%	XNet	0.762	7.152	220.405	0.842
		Ours	**0.783**	**6.661**	**196.357**	**0.789**
	20%	XNet	0.776	6.750	198.525	0.831
		Ours	**0.793**	**4.250**	**188.642**	**0.787**

Comparison to fully-sup. baselines. To better demonstrate that our method can effectively unearth the topological information from the unlabeled data, we make a comparison with two fully-supervised methods: [18] and [6]. We use these two losses only on 20% labeled training data and report the results in Tab. 11. Our TopoSemiSeg consistently outperforms these baselines because we utilize the topological information from the unlabeled data.

Table 11: Comparison to fully-sup. baselines.

Method	Dice_obj ↑	BME ↓
TopoLoss [18]	0.865	19.925
TopoLoss [6]	0.857	24.625
Ours	**0.898**	**8.575**

Accuracy/guarantee of the decomposition strategy. Using a persistence threshold to filter out topological noise is theoretically supported. The stability theorem of persistent homology [7,8] guarantees that topology due to small perturbation has small persistence. This is also demonstrated in Fig. 2. We observe that a proper persistence threshold ensures the model learns true structures and eliminates noise. To validate this, we compare the selected signal topology with the ground truth (GT) topology. On CRAG unlabeled training set, we compare the number of selected signal topology of the teacher with the Betti number of the GT. Fig. 8 shows the mean absolute difference between the two decreases during training. Thus, as training continues, the teacher's signal topology approaches GT's. This empirically shows that the decomposition picks up true topology signals, which the student learns from.

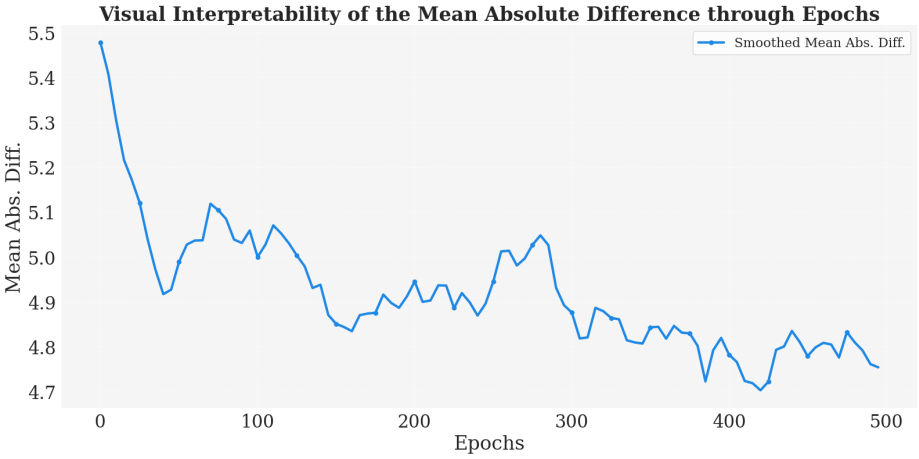


Fig. 8: Visual interpretation of the decomposition strategy.