We thank the reviewers for constructive feedback. We will improve our manuscript accordingly. Some concerns were raised regarding the reproducibility. We will address them below. If accepted, we will release the code immediately.

Q1 (R1,R3): How is the optimal matching computed? How is $\mathrm{Dgm}(f)$ computed? Small noisy components?

A: The matching algorithm is as follows. A total of $k =$ (Betti number) dots from ground truth ($\mathrm{Dgm}(g)$) are at the upper-left corner $p = (0, 1)$ (in Fig.4 (b) of the paper). We find the k dots closest to the corner p in $\mathrm{Dgm}(f)$ and match them to the ground truth dots . The remaining dots in $\mathrm{Dgm}(f)$ are matched to the diagonal line. The algorithm computes and sorts the squared distances from all dots in $\mathrm{Dgm}(f)$ to p. The complexity is $O(n \log n)$, n = # of dots in $\mathrm{Dgm}(f)$. In a general setting, the state-of-the-art diagram matching algorithm [Kerber et al. 2017] takes $O(n^{3/2})$.

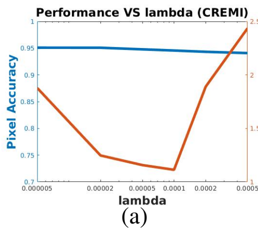
The dots matched to the diagonal line correspond to small noisy components or noisy loops. These dots will be pushed to the diagonal. And their corresponding components/loops will be removed or merged with others.

To compute the persistence diagram $\mathrm{Dgm}(f)$, we use the classic algorithm of [13,14]: We first discretize an image patch into vertices (pixels), edges and squares. The adjacency relationship between these elements and their likelihood function values are encoded in a boundary matrix, whose rows and columns correspond to vertices/edges/squares. The matrix is reduced using a modified Gaussian elimination. The pivoting entries of the reduced matrix correspond to all the dots in $\mathrm{Dgm}(f)$. Efficient software packages are publicly available (GUDHI, RIPSER, PHAT, etc).

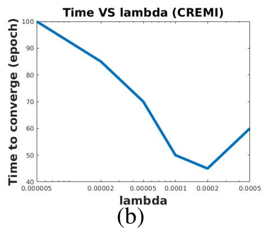
Q2 (R1,R3): How is the prediction across patches aggregated? Handle the boundaries between patches?

A: We do not partition the image into patches. Instead, we randomly and densely sample patches which can overlap. Our loss enforces correct topology within each sampled patch. These overlaps between patches propagate correct topology everywhere. Correct topology within a patch means the segmentation can be a deformation of the ground truth. But the deformation is constrained within the patch. The patch size controls the tolerable geometric deformation.

Q3 (R1,R3): Ablation study of lambda. lambda=0 should also be evaluated as a baseline. Report standard deviation.



(a)



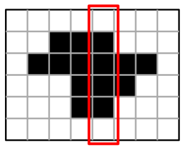
(b)

A: Figures (a)(b) show ablation studies of lambda on CREMI w.r.t. accuracy, Betti error and convergence rate. As we increase lambda, per-pixel accuracy is slightly compromised. The Betti error decreases first but increases later. One important observation is that a certain amount of topological loss improves the convergence rate significantly. These observations suggest choosing lambda = 1×10^{-4}. We will also add ablation study w.r.t. patch size and report standard deviation in the final version. Since our method is based on the DIVE architecture, the baseline DIVE is equivalent to the case when lambda = 0.

Q4 (R1): Training with a dilated ground truth mask?

A: A dilated mask cannot help. The topological loss identifies difficult locations (e.g., blurred membrane locations in Figure 1 of the paper) as critical pixels and forces the network to learn to classify them correctly. Dilated ground truth masks cannot help with this. For CREMI dataset, using a topology-agnostic model with dilated masks (1 and 2 pixel dilation), we have Betti Error 4.126 and 4.431 respectively, still significantly worse than TopoNet (Betti Error = 1.113).

Q5 (R1): Usefulness of the proof. What is the benefit of the loss or the proof?



A: The correctness proof ensures that the topological loss enforces topological constraints. Topological constraints can significantly reduce the size of the solution space, making the optimization much easier. We prove this for a simple connectivity constraint and leave general cases for future work. For an $m \times m$ image patch, the space of all binary segmentations has DoF = $m \times m$ Degrees of Freedom. But if we force the segmentation to be connected, the solution space is much smaller. See the figure on the right. Within each column (or row) of pixels, the segmentation is an interval with DoF $= 2$ (locations of two ends). Over all rows and columns, the DoF of the connected segmentation space is only $4m$.

Furthermore, in practice, the topological loss combats sample bias. As noted in Q4, a network learns to predict most pixels except for a few difficult locations. There are never sufficient samples at such difficult locations. A topological loss mitigates such sample bias by identifying these locations as critical pixels and increasing their weights in the training. Thus, training converges faster (Figure (b)) and topology-relevant metrics improve.

Q6 (R2): What if cross entropy loss is turned off or downweighted? Relative importance of the two losses.

A: Per Q5, the topological loss complements the cross-entropy loss by improving training efficiency and mitigating sampling bias. However, without cross-entropy loss, inferring topology from a completely random likelihood map is meaningless. Cross-entropy loss finds a reasonable likelihood map so that the topological loss can improve its topology.

Q7 (R2): Robust to geometry errors: help in misalignment between the ground truth and training images?

A: Absolutely. This is an additional benefit of the topological loss that increases robustness to geometry errors.

Q8 (R3): Can both types of metrics be maximized? If not, can post-processing help?

With sufficient samples for the difficult locations (blurred locations), we can maximize both types of metrics. However, this is not the case in reality. We use topological loss to force the model to memorize patterns at the difficult locations, at the expense of overfitting and consequently slightly compromised per-pixel accuracy. Post-processing at test time should help in general, but does not identify these difficulty locations for specific handling.