A Learning-based Framework for Topology-Preserving Segmentation using Quasiconformal Mappings
Han Zhang, Lok Ming Lui

Abstract
We propose the Topology-Preserving Segmentation Network, a deformation-based model that can extract objects in an image while maintaining their topological properties. This network generates segmentation masks that have the same topology as the template mask, even when trained with limited data. The network consists of two components: the Deformation Estimation Network, which produces a deformation map that warps the template mask to enclose the region of interest, and the Beltrami Adjustment Module, which ensures the bijectivity of the deformation map by truncating the associated Beltrami coefficient based on Quasiconformal theories. The proposed network can also be trained in an unsupervised manner, eliminating the need for labeled training data. This is achieved by incorporating an unsupervised segmentation loss. Our experimental results on various image datasets show that TPSN achieves better segmentation accuracy than state-of-the-art models with correct topology. Furthermore, we demonstrate TPSN’s ability to handle multiple object segmentation.

Index Terms
Topology-Preserving, Prior Knowledge, Deformable Model, Medical Image Segmentation

I. INTRODUCTION
Image segmentation, the process of identifying and separating regions of interest in images, is a crucial task in many applications such as object recognition, image editing and medical imaging. For instance, medical image segmentation is known to be a crucial step in many clinical applications, including disease diagnosis and monitoring of disease progression. While deep learning has revolutionized the field of image processing in recent years, developing learning-based segmentation models is still facing a significant challenge due to the limited availability of the training data. This situation is particularly common in the medical field, as medical image data is frequently subject to privacy and legal regulations that limit its availability. The lack of data poses challenges to training models that are capable of achieving high levels of accuracy in segmenting anatomical structures from medical images.

However, despite the limited availability of image data, it is worth noting that structures of interest in many applications follow strict morphological rules. These rules can be incorporated as prior knowledge into learning-based segmentation models. By using these prior rules, it is possible to guide the segmentation process and achieve better segmentation quality, even with limited data availability. For instance, anatomical structures such as the kidney or liver are known to be simply-connected, while vessel walls are typically circularly shaped. Integrating the topological rule into a learning-based segmentation model can improve the segmentation result, as well as ensure that the segmented object is consistent with the anatomical structure being imaged.

Conventional segmentation methods often fail to consider geometric information, which can result in inaccurate segmentation with topological errors. For example, mathematical models for image segmentation based on pixel-wise classification may produce isolated segmented regions and create holes in anatomical structures. These issues can be clinically problematic and impractical [4]. To address this problem, recent works have incorporated prior shape information into regularization terms, which can improve segmentation results but do not ensure topological correctness [4]. On the other hand, deformation-based segmentation models can enforce topological and geometric constraints by warping a template image to a target domain that encompasses the region of interest. For instance, topology- and convexity-preserving segmentation models have been proposed recently. These models deform a template shape by mapping it to segment the image [3], [29]. The properties of the mapping are carefully controlled.

In recent years, learning-based imaging models have gained significant attention as a means to integrate data information and enhance efficiency. By utilizing U-Net [27], dense spatial transformations can be generated using deep neural network models [16]. This opens up possibilities for developing learning-based segmentation techniques based on deformable models. However, enforcing geometric constraints on the deformation map within a deep neural network remains a challenge. Recent studies have explored various regularizations to be imposed on the output of a spatial transformer network or its sub-layers. However, these approaches often tend to be over-constrained, leading to sub-optimal results [34], or lack mathematical justifications to guarantee a solution that satisfies the prescribed geometric conditions [21].

H. Zhang is with the City University of Hong Kong, Hong Kong SAR (e-mail: hzhang863-c@my.cityu.edu.hk).
L.M. Lui is with the Chinese University of Hong Kong, Hong Kong SAR (e-mail: lmlui@math.cuhk.edu.hk). Corresponding author.
In this paper, we present a topology-preserving segmentation framework that utilizes a learning-based approach. Our proposed framework, namely Topology-Preserving Segmentation Network (TPSN), learns to segment an input image by deforming the template mask to enclose the region of interest while preserving topology. The preservation of topology is achieved by ensuring the bijectivity of the deformation map. To enforce the bijectivity of the deformation map, we introduce the Deformation Estimation Network regularized by ReLU-Jacobian and the Beltrami Adjustment Module based on Quasiconformal theories. The ReLU-Jacobian regularizer, which is the negative of the Jacobian determinant activated by the ReLU function, discourages changes in orientation under the deformation map in the segmentation model, thereby promoting bijectivity. To guarantee bijectivity, we further propose a Beltrami adjustment module based on quasiconformal theories that adjust the Beltrami coefficient of the mapping, which measures local geometric distortions. This module can guarantee the bijectivity of the deformation map, even in extreme cases, resulting in topology-preserving segmentation results. Additionally, our proposed network can be trained in an unsupervised manner, which eliminates the need for labeled training data. This is achieved by incorporating an unsupervised loss into our proposed model.

We conduct extensive experiments to evaluate the performance of our proposed framework. The proposed model is first applied for supervised segmentation, which involves training the segmentation network with a set of images with labeled masks. The loss function includes the proposed ReLU Jacobian regularization and a pixel-wise segmentation loss, such as the Dice loss. Our proposed network successfully produces segmentation results with the prescribed topological constraint. We evaluate the performance of the model on the KiTS21[13] and ACDC[2] dataset for simply-connected and doubly-connected object segmentation, respectively. To segment more general multiply-connected objects, we design the Fill First, Dig Second(FFDS) strategy. To further improve the segmentation accuracy, we propose a multi-level TPSN architecture that predicts the segmentation mask from a coarse-to-fine fashion. We also test our proposed model for unsupervised segmentation using an unsupervised segmentation loss in the model. Our proposed framework is evaluated on the BTCV[20] dataset extracted from MICCAI 2015 challenge, and experimental results are also promising. From the experiments on the corrupted images, our model is robust under extreme cases where very noisy images are considered, with the help of the Beltrami adjustment module.

To summarize, the contributions of this work are as follows:

- We proposed a segmentation framework called Topology-Preserving Segmentation Network (TPSN), which learns to segment an input image by deforming the template mask to enclose the region of interest while preserving topology.
- We propose a methodology for topology-preserving segmentation that combines the ReLU-Jacobian regularizer and the Beltrami adjustment module. The ReLU-Jacobian regularizer promotes bijectivity and the Beltrami adjustment module guarantees bijectivity even in extreme and noisy cases. This approach results in topology-preserving segmentation masks.
- Our proposed network can be trained in an unsupervised manner using the Mumford-Shah model’s fidelity term, which distinguishes the foreground and background based on image intensities, eliminating the need for labeled training data.

II. RELATED WORK

A. Image Segmentation

Image segmentation has been studied extensively and various segmentation models have been developed in recent years. The active contour model [18] is perhaps the earliest segmentation model, which evolves a parameterized curve iteratively to the object boundary. Later, Chan et al. [5] proposed the active contour model without edge, based on the Mumford-Shah functional and level sets, to segment an image by curve evolution. The model has an advantage of allowing topological changes. With the popularity of deep learning in recent years, a lot of segmentation models using deep architectures have been invented. Ronneberger et al. [27] proposed the learning-based segmentation model using the U-Net structure and achieved impressive results. The extension of U-Net for 3D volumetric images was later developed in [24], [8]. As learning-based segmentation models require data, multiple works have been devoted to generating meaningful training data. A series of dataset [13], [2], [13], [20] are organized and open to the public for research purposes. Some works design learning-based segmentation models by manipulating data. For instance, a segmentation framework, called nnUNet, was proposed in [15], which uses only a standard U-Net architecture with a series of practical pre- and post- processing strategies on the images. This framework achieved outstanding results on multiple segmentation datasets. More learning-based models have been invented recently by adjusting the network architectures. Jha et al. [17] proposed to use one U-Net to predict an initial mask and apply an additional one to refine the segmentation result.

Besides, unsupervised learning-based segmentation models have also been considered, which train the model by images without labeled masks as the ground truth. Ouali et al. [26] proposed the autoregressive unsupervised image segmentation model by maximizing the mutual information between different constructed views.

B. Deformable Model

Deformable models for image segmentation have also been explored by different research groups. The main idea of deformable segmentation models is to obtain the result by searching for a suitable deformation. The active contour model [18] can be considered as a deformable model, which deforms a collection of points discretizing a curve to enclose the boundary of the
Figure 1: The TPSN network architecture consists of two components: the Deformation Estimation Network (represented by the yellow box) and the Beltrami Adjustment Module (represented by the blue box). For a segmentation task for \( q \) classes of objects, the Deformation Estimation Network utilizes an encoder-decoder architecture that takes in a prior \( M_{temp} \) and an image \( I \) with \( C \) channels, and produces \( q \) mappings for each class. This mapping is then used to deform prior masks to enclose regions of interest \( M_{pred} \). The component is designed with ReLU-Jacobian regularization, which promotes the output of a topology-preserving mapping. The Beltrami Adjustment Module consists of two convolutional layers activated by a \( \text{Tanh} \). This component ensures that the mapping is topology-preserving by truncating \( \mu \) into \( \tilde{\mu} \). The Beltrami Solver Network [7] reconstructs the quasiconformal mapping \( \tilde{f} \) that corresponds to the truncated \( \tilde{\mu} \). The corresponding prediction mask \( f_{M_{pred}} \) is then obtained.

Learning-based models that utilize deformable models have gained significant attention in recent years [11], [36] since the introduction of spatial transformer networks [16]. To improve the preservation of topology in the final output, Lee et al. [21] applied Laplacian regularization. Wyburd et al. [34] proposed a topology-preserving deep segmentation network by compositing a series of bijective deformations. However, these methods may not have a mathematical guarantee of topological preservation or may produce suboptimal results due to over-constraints, particularly for structures with complex geometry.

C. Prior Information In Neural Network

Recent research has focused on incorporating prior information into deep neural networks by designing an appropriate loss function that can integrate prior knowledge to regulate predicted masks. Yan et al. [35] proposed a convexity-preserving method for variational level-set-based image segmentation. Mosinska et al. [25] defined a topological loss that considers higher-order topological features of linear structures. Zhou et al. [40] proposed a segmentation model that statistically considers the relative location and size, performing well in multi-organ segmentation. Hu et al. [14] designed a continuous-valued loss function to enforce topological correctness using computational topology theories. Shit et al. [28] introduced cIDice to measure the Dice of skeletons between the prediction and the ground truth for tubular structures. Clough et al. [9] proposed using differentiable persistent homology to drive the segmentation process to produce an output with a prescribed Betti number.

III. PROPOSED METHOD

In this section, we describe our proposed topology-preserving segmentation network (TPSN) in detail.

A. Overall framework

Our proposed TPSN is a deep neural network that takes an image and a template mask as input and outputs the segmentation mask. Figure 1 depicts the overall framework, which is composed of two components: (1) the Deformation Estimation Network (DEN) and (2) the Beltrami Adjustment Module (BAM).

The DEN takes the image and the template mask as input, where the template mask encodes the desired topological properties. Based on the input, a deformation mapping used to warp the template mask into the region of interest is generated. The
ReLU-Jacobian term is included in the loss function of the DEN as a soft regularisation term to encourage the deformation mapping to be bijective. Under normal circumstances, training the DEN with the ReLU-Jacobian regularization enables it to generate bijective mappings. However, in challenging situations like working with noisy images, there is no assurance that the deformation mapping produced by the DEN will maintain bijectivity. To tackle this challenge, the Beltrami Adjustment Module is introduced as a solution to ensure the bijectivity of the generated mapping. This module, which functions as a post-processing operation, does not learn from data. It is optimized by minimizing \([3]\), which relies solely on the predicted mapping and the corresponding transformed mask obtained from DEN. By employing this approach, BAM is able to produce a bijective quasi-conformal mapping for each input while preserving the region enclosed by DEN to the greatest extent possible.

Finally, the bijective quasiconformal map is used to warp the template mask to construct the final segmentation result, which is guaranteed topology-preserving. In the following subsection, we will explain every component in our framework.

### B. Deformation Estimation Network (DEN)

**Network architecture:** The architecture of the Deformation Estimation Network (DEN) is depicted in Figure [1] where the channels are delineated beneath each layer. Each convolutional operation within the network is performed using a \(3 \times 3\) kernel for 2D images and a \(3 \times 3 \times 3\) kernel for 3D images. Subsequently, batch normalization is applied, followed by activation through the rectified linear unit (ReLU) function. Throughout the process, three max-pooling and three bilinear upsampling operations are incorporated for feature extraction and resolution enhancement, respectively. The network takes the image \(I\) to be segmented and concatenates it with a pre-defined template mask \(M_{\text{temp}}\), which is chosen based on the topological prior of the object being segmented. This concatenated input is then fed through an encoder-decoder network, which generates a deformation mapping \(f = \mathcal{N}(I, M_{\text{temp}})\) that warps the template mask into a predicted mask \(M_{\text{pred}} = M_{\text{temp}} \circ f\). To ensure that the orientation of the deformation mapping is preserved as much as possible, a suitable loss function is minimized during the training process. The specific details of this loss function will be described in the next paragraph.

**Loss:** The main goal of our segmentation framework is to obtain an accurate segmentation result that preserves the prescribed topology. To achieve this, our strategy is to obtain a bijective deformation mapping that transforms a template mask with the prescribed topological prior to obtain an accurate segmentation mask. The bijectivity of the deformation mask ensures that the topology of the segmentation mask is consistent with that of the template mask. To accomplish this, a suitable loss function is necessary. In this work, we propose a loss function that consists of the fidelity term and the regularization term. The regularization term aims to regularize the deformation mapping. To encourage the bijectivity of the deformation mapping, the ReLU Jacobian regularizer is proposed. A map is homeomorphic if its Jacobian is positive everywhere. Motivated by this, the following ReLU Jacobian regularizer is proposed:

\[
\mathcal{L}_{\text{Jac}}(f) = \|\text{ReLU}(- \text{det} \nabla(f))\|_1
\]

where \(\text{det} \nabla(f)\) denotes the Jacobian determinant of \(f\), \(\text{ReLU}(\cdot)\) represents the ReLU activation function that sets negative values to zero while keeping positive values. \(\mathcal{L}_{\text{Jac}}(f)\) penalizes the portion of the deformation map with negative Jacobian.

Minimizing this loss function encourages the deformation mapping to preserve the orientation as much as possible.

Besides, we introduce an additional Laplacian regularization term aimed at smoothing the deformation mapping:

\[
\mathcal{L}_{\text{Lap}}(f) = ||\Delta(f)||_1
\]

where \(\Delta(f)\) denotes the Laplacian of the mapping \(f\). This Laplacian regularization promotes smoothness in the deformation field, thereby mitigating abrupt changes and preserving the overall structural integrity of the segmented objects. By incorporating this regularization technique alongside the ReLU-Jacobian regularization, we enhance the accuracy of the segmentation output.

Next, the fidelity loss aims to promote segmentation results that accurately match and faithfully represent the input image data. For supervised learning where the training data has labeled masks, the Dice loss is used. The overall loss function is then given by:

\[
\mathcal{L} = \mathcal{L}_{\text{Dice}}(M_{\text{pred}}, M_{\text{label}}) + \lambda_{\text{Jac}} \mathcal{L}_{\text{Jac}}(f) + \lambda_{\text{Lap}} \mathcal{L}_{\text{Lap}}(f)
\]

where \(\mathcal{L}_{\text{Dice}}\) is the Dice loss between the predicted mask \(M_{\text{pred}}\) and label mask \(M_{\text{label}}\). \(\lambda_{\text{Jac}}\) and \(\lambda_{\text{Lap}}\) are the weighting parameters for \(\mathcal{L}_{\text{Jac}}(f)\) and \(\mathcal{L}_{\text{Lap}}(f)\) respectively. It’s also worth to note that the label mask \(M_{\text{label}}\) are given while the predicted mask \(M_{\text{pred}}\) are transformed from a given template mask \(M_{\text{temp}}\) by the predicted mapping \(f\).

For unsupervised learning without the need for labeled training data, the following unsupervised segmentation loss is used as the fidelity term:

\[
\mathcal{L}_{\text{seg}}(I, M_{\text{pred}}) = \int_D (I - c_1 M_{\text{pred}} - c_2 (1 - M_{\text{pred}}))^2 dx
\]

where \(c_1 = \int_D I M_{\text{pred}} dx / \int_D M_{\text{pred}} dx\) and \(c_2 = \int_D I (1 - M_{\text{pred}}) dx / \int_D (1 - M_{\text{pred}}) dx\), \(D\) is the whole image domain. In fact, \(c_1\) denotes the mean pixel value of the foreground (region covered by the predicted mask) while \(c_2\) is the mean pixel value of the background. \(M_{\text{pred}} = M_{\text{temp}} \circ f\) represents the final segmentation mask. \(M_{\text{pred}}\) is derived by deforming the binary template mask \(M_{\text{temp}}\) using the mapping \(f\). Consequently, with nearest sampling method, \(M_{\text{pred}}\) is also a binary image, where the interior region has an intensity value of
Quasi-conformal maps infinitesimal circles to ellipses. The Beltrami coefficient measures the distortion or dilation of the ellipse under the QC map.

1, and the exterior region has an intensity value of 0. The purpose of Equation (4) is to generate a segmentation mask that ensures the intensity values of the input image $I$ within the interior region of the mask are close to the average intensity of that region. Similarly, the intensity values of $I$ within the exterior region of the mask should be close to the average intensity of the exterior region. This objective aims to enable accurate segmentation of a piecewise constant image, where the segmentation mask aligns well with regions of consistent intensity. Thus, the overall loss function for the unsupervised case is given by:

$$L = L_{\text{seg}}(I, M_{\text{pred}}) + \lambda_{\text{Jac}}L_{\text{Jac}}(f) + \lambda_{\text{Lap}}L_{\text{Lap}}(f)$$

Overall, given a template mask $M_{\text{temp}}$ and label mask $M_{\text{true}}$, the predicted mask can be computed from $M_{\text{pred}} = M_{\text{temp}} \circ f$, where $f$ is the predicted output from a deformation estimation network.

### C. Beltrami Adjustment Module (BAM)

With the ReLU Jacobian regularizer, DEN can produce a deformation map that preserves the orientation as much as possible. In the case where the input data is close to the distribution of the training dataset, the output deformation mapping from DEN is mostly bijective, resulting in a topology-preserving segmentation mask. However, when a new input is highly noisy, which makes it significantly different from the training data distribution, the resulting mapping may not be perfectly bijective. To address this issue and ensure that the deformation mapping produced by our segmentation framework is bijective, we propose the implementation of a Beltrami Adjustment Module (BAM). This module adjusts the deformation mapping to ensure bijectivity. Every mapping is associated with a geometric quantity called the Beltrami coefficient, which is defined as follows:

$$\mu = \frac{\partial f}{\partial \bar{z}} / \frac{\partial f}{\partial z},$$

where $\frac{\partial}{\partial \bar{z}} = \frac{1}{2}(\frac{\partial}{\partial x} - i \frac{\partial}{\partial y})$ and $\frac{\partial}{\partial z} = \frac{1}{2}(\frac{\partial}{\partial x} + i \frac{\partial}{\partial y})$. $i$ is the imaginary unit.

Based on $\mu$, it is possible to determine that the angle of maximum magnification is $\arg(\mu)/2$, accompanied by a magnification factor of $|\frac{\partial f}{\partial z}|(1 + |\mu|)$. Similarly, the angle of maximum contraction is the orthogonal angle $(\arg(\mu) - \pi)/2$, with a contraction factor of $|\frac{\partial f}{\partial z}|(1 - |\mu|)$ [19]. This implies that the Beltrami coefficient $\mu$ measures the local geometric distortion under $f$, as illustrated in Figure 2.

The Beltrami coefficient $\mu$ is a complex-valued function defined on the image domain. A Beltrami coefficient $\mu$ in 2 dimensions can be expressed as a 2-channel image of the same size as the corresponding input images. In the discrete case, the image domain is triangulated and each deformation mapping is treated as a piecewise linear function over each triangular face. The first derivatives of a piecewise linear function are constant on each triangular face. Consequently, the Beltrami coefficient can be regarded as a complex-valued function defined over each triangular face. The discretization details will be described in the next section. $\mu$ measures the local geometric distortion under the mapping. If $\mu(T) = 0$ on a triangular face $T$, the three angles of the triangle will be preserved under the deformation.

Importantly, the magnitude of the Beltrami coefficient is strictly less than 1 everywhere for an orientation-preserving homeomorphism. Leveraging this fact, the Beltrami Adjustment Module (BAM) is designed to control geometric distortion by adjusting the Beltrami coefficient associated with the deformation mapping. After obtaining $f$ from the DEN, we calculate its
Theorem 1. The Beltrami Adjustment Module outputs a bijective mapping \( \hat{f} \) if and only if its associated Beltrami coefficient \( \mu \) is well-defined and satisfies \( |\hat{\mu}(T)| < 1 \) for all triangular faces \( T \).

Proof. Denote the image domain as \( \Omega \). \( \hat{f} \) is piecewise linear on every triangular face \( T \). In \( T \), \( \hat{f} \) can be written as \( a_1^x x + a_2^y y + b^x + i(a_1^x x + a_2^y y + b^y) \). The Jacobian \( J_T \) of \( \hat{f} \) is a constant on each \( T \). Also, \( J_T = (a_1^x a_2^y - a_2^x a_1^y) + i((a_1^x a_2^y - a_2^x a_1^y)^2 - \frac{1}{4}((a_1^x - a_2^x)^2 + (a_2^y - a_1^y)^2)) = \frac{\partial f}{\partial T}(T)^2(1 - |\hat{\mu}(T)|^2) \). If \( \hat{f} \) is bijective and orientation-preserving, the Jacobian of \( \hat{f} \) is positive everywhere and \( \frac{\partial \hat{f}}{\partial T}(T) > 0 \) for all \( T \) since \( \hat{\mu} \) is well-defined. Also, \( (1 - |\hat{\mu}(T)|^2)^2 > 0 \) and \( |\hat{\mu}(T)| < 1 \) for all \( T \). Conversely, since \( \hat{\mu}(T) = \frac{\partial f}{\partial T}(T) \frac{\partial \hat{f}}{\partial T}(T) \) is well-defined for all \( T \), \( \frac{\partial f}{\partial T}(T) \neq 0 \) and \( |\frac{\partial f}{\partial T}(T)| > 0 \). If \( |\hat{\mu}(T)| < 1 \) for all \( T \), then \( J_T = \frac{\partial f}{\partial T}(T)^2(1 - |\hat{\mu}(T)|^2) > 0 \) for all \( T \). Hence, \( \hat{f} \) is a local homomorphism. Also, by construction, \( \hat{f} \) is an identity map on the boundary \( \partial \Omega \) and thus \( \hat{f}|_{\partial \Omega} \) is a homomorphism.

Let \( \phi: \Omega \to \hat{D} \) be a homeomorphic parameterization of \( \Omega \) onto a unit disk \( \hat{D} \). Then, \( \hat{f} = \phi \circ \hat{\phi} \circ \phi^{-1} : D \to \hat{D} \) is a local homeomorphism and \( \hat{f}|_{\partial \hat{D}} \) is homeomorphic. We proceed to show that \( \hat{f} \) is a global bijection, which implies \( \hat{f} = \phi^{-1} \circ \hat{\phi} \circ \phi \) is a global bijection.

We first prove that \( \hat{f} \) is surjective. Suppose not. We can find a point \( p \) in the interior of \( D \), which is not in the image \( \hat{f}(D) \) of \( \hat{f} \). Let \( \mathcal{P}: D \to \partial D \) be the projection map from \( D \) to the unit circle \( \partial D \), as depicted in Figure 3. Basically, for each \( q \in D \setminus \{p\} \), join \( p \) and \( q \) to get a line. Find the intersection point of the line with \( \partial D \) closest to \( q \) and define it as \( \mathcal{P}(q) \). Then, we have \( G = (f|_{\partial D})^{-1} \circ \mathcal{P} \circ \hat{f} : D \to \partial D \) is an identity map on \( \partial D \). Note that \( (f|_{\partial D})^{-1} \) is well-defined since \( f|_{\partial D} \) is homeomorphic by the no retraction theorem, such a map does not exist. We conclude that \( \hat{f} \) is surjective.

To prove that \( \hat{f} \) is injective, it suffices to show that \( f|_{\partial D} : D \setminus \partial D \to \partial D \) is injective. Note that \( f|_{\partial D} \) is surjective local homeomorphism and a covering map. Since \( D \setminus \partial D \) is simply-connected, \( f|_{\partial D} \) is a universal covering map. In our case, the cardinality of the fibers of the universal cover is 1. Hence, \( f|_{\partial D} \) is injective.

As \( \hat{f} \) is bijective, the final segmentation mask obtained by transforming the template mask by \( \hat{f} \) is topology-preserving. Note that the Beltrami Adjustment Module is a simple CNN, whose parameters are optimized to minimize \( \mathcal{L}_{\text{trun}} \).
Figure 4: Illustration of the multi-level TPSN. The original image is first downsampled to a low-resolution image, which is then inputted into a TPSN with a prior of the same size. The predicted mask is subsequently upsampled and used as the prior mask in the next layer of the TPSN.

Speaking, the Beltrami Adjustment Module (BAM) acts as a post-processing step, ensuring the bijectivity of the mapping for each image while preserving the region of interest. If the mapping \( f \) from the Deformation Estimation Network (DEN) is already bijective, BAM remains inactive since \( L_{\text{trun}} = 0 \). The Beltrami Solver Network (BS-Net) within BAM remains fixed after pre-training to efficiently solve for the mapping \( \tilde{f} \) from \( \tilde{\mu} \). In the case of 3D, the generalized conformity distortion \( K(\tilde{f}) \) in an \( n \)-dimensional space can be used. This geometric quantity is analogous to measuring quasiconformality in higher dimensions.[37]

D. Multi-level TPSN

To enhance the robustness and accuracy of our proposed model, we utilize a multi-level strategy to construct the "multi-level topology-preserving segmentation network (ml-TPSN)". The pipeline and architecture of ml-TPSN are depicted in Figure 4. We first downsample the original image \( I \) into \( I' \) and \( I'' \), with dimensions 1/2 and 1/4 of the original, respectively. \( I'' \) is then fed into an encoder-decoder network with a coarse template mask of the same size to output a predicted segmentation mask. This coarse predicted mask is upsampled and employed as the template mask \( M' \) in the next level for \( I' \). \( I' \) and \( M' \) are then fed into another encoder-decoder network, and the process is repeated. The final encoder-decoder network outputs a mask \( M \) with the same resolution as the original image \( I \). This multi-level strategy enables our model to extract features in a coarse-to-fine manner, with the predicted mask used as the template mask in the next TPSN layer, leading to a more robust and accurate segmentation, particularly for structures with complex geometry. Additionally, this strategy effectively prevents sub-optimal solutions. The proposed ml-TPSN architecture thus offers improved segmentation results, as demonstrated in our experiments.

In our implementation, we utilize the nearest interpolation for both downsampling and upsampling. This choice ensures that the template mask at each level remains binary, containing only integer values.

E. Fill First Dig Second (FFDS) strategy

Our proposed framework has demonstrated exceptional efficacy in segmenting multiply-connected objects. To further enhance its performance, we introduce a novel strategy termed Fill First Dig Second (FFDS), depicted in Figure 5. This method involves initially utilizing a simply-connected shape and deforming it to encompass the target region. This process initially fills holes in the label mask, simplifying the segmentation task. During the initial phase, known as the First Fill, the filled template mask is transformed to cover the region of interest whose holes are also filled. Subsequently, in the second phase, holes are excavated from the template to match the prescribed topology of the transformed template mask. The original labeled mask with holes serves as the training label in this phase. Following sufficient training, the transformed mask with holes undergoes further adjustments to ensure its interior edges align with the inner contour of the target region.

The FFDS strategy is crucial. It is important to note that the overlap between the template mask and the label mask plays a crucial role in computing the gradient for training. Without this overlap, the gradient would be null, impeding the training process. By employing the FFDS strategy and filling the holes in the template mask during the first stage, we ensure that there is sufficient overlap between the template mask and the label mask. This allows the model to compute meaningful gradients. As a result, even complex, multiply-connected objects, including those found in other deformable models such as [34], [21], can be effectively segmented, as shown by our extensive testing.

F. Multi-object Segmentation

Multi-object segmentation is a process that involves the extraction of multiple objects from an image. This technique has found important applications in medical imaging, where the detection and segmentation of multiple anatomical structures can
be vital for diagnosis and treatment planning. However, segmenting each individual object using different networks can be both time-consuming and highly costly. While pixel-wise segmentation networks for multi-object segmentation are available in the literature [12], the use of deformable models for this task has been relatively unexplored. Our proposed framework can be naturally applied for multi-object segmentation. For single-object segmentation, the output of the encoder-decoder module is typically two or three channels, depending on the dimension of the image. However, for segmenting multiple objects, the number of channels in the output of the TPSN is designed to be $2q$ or $3q$, where $q$ is the number of objects to be segmented. Each group of three channels is then used to give the deformation map necessary to deform the associated template, shown with different colors in the template mask in Figure 1. The deformation map is then fed into the Beltrami Adjustment Module to guarantee the bijectivity and is further used to segment the corresponding class of objects in the image.

IV. DISCRETIZATION OF DEFORMATION MAPPINGS

Our proposed network controls the Jacobian and the Beltrami coefficient of the deformation mapping. In this work, the computation of the Jacobian and Beltrami coefficient is based on the finite element method. The image domain is discretized by a triangulation mesh using the standard finite element division (see Figure [9]). In the 2D case, the image domain is partitioned into small squares and each small square is divided into two triangles. In the 3D case, each cube is divided into six tetrahedrons. The deformation mapping is assumed to be piecewise linear on each triangular face (or each tetrahedron).

A. Computations of Jacobian and Beltrami coefficients

We consider the computational of the Jacobian for the 2D case. The computation for the 3D case can be carried out similarly. For an image of size $H \times W$, we parametrize it into a standard domain $[-1, 1] \times [-1, 1]$. The standard domain is discretized by a regular grid by $G = \{(x_i, y_j)\}$, where $h_x = \frac{2}{H-1}$ and $h_y = \frac{2}{W-1}$. Each small square in the grid parameterization are divided into $2^2$ triangles. Since the deformation mapping is considered as a piecewise linear mapping, the Jacobian is a piecewise constant $2 \times 2$ matrix on each triangular face. We denote the vertices of a triangle face on the source grid by $v_0, v_1$ and $v_2$, where $v_i = (x_i, y_i)$ $(i = 0, 1, 2)$. The transformed vertices by the deformation mapping $f = (f_x, f_y)$ are given by $v_0^*, v_1^*$ and $v_2^*$, where $v_i^* = (x_i^*, y_i^*) = f(v_i)$ $(i = 0, 1, 2)$. Since $f$ is piecewise linear on each triangular face, we have

$$
\begin{pmatrix}
  x^*
  y^*
\end{pmatrix} = \begin{pmatrix}
  f_x(x, y)
  f_y(x, y)
\end{pmatrix} = \begin{pmatrix}
  a_0^x + a_1^x x + a_2^x y + b_x
  a_0^y + a_1^y x + a_2^y y + b_y
\end{pmatrix}
$$

(9)

Hence, the Jacobian of $f$ can be written as

$$
J(f) = \begin{pmatrix}
  \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} \\
  \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y}
\end{pmatrix} = \begin{pmatrix}
  a_1^x & a_2^x \\
  a_1^y & a_2^y
\end{pmatrix}
$$

(10)

Using $v_i^* = f(v_i)$ for $i = 0, 1, 2$, Equation [9] can be written in the matrix form:

$$
\begin{pmatrix}
  x_1^* & x_2^* & \cdots & x_n^*
  y_1^* & y_2^* & \cdots & y_n^*
\end{pmatrix} = \begin{pmatrix}
  a_1^x & a_2^x & b_x \\
  a_1^y & a_2^y & b_y
\end{pmatrix} \begin{pmatrix}
  x_1 & x_2 & \cdots & x_n
  y_1 & y_2 & \cdots & y_n
\end{pmatrix}.
$$

(11)

It can be further reduced into

$$
\begin{pmatrix}
  x_1^* - x_2^* & x_2^* - x_3^* & \cdots & x_{n-1}^* - x_n^*
  y_1^* - y_2^* & y_2^* - y_3^* & \cdots & y_{n-1}^* - y_n^*
\end{pmatrix} = \begin{pmatrix}
  a_1^x & a_2^x & b_x \\
  a_1^y & a_2^y & b_y
\end{pmatrix} \begin{pmatrix}
  x_1 - x_2 & x_2 - x_3 & \cdots & x_{n-1} - x_n
  y_1 - y_2 & y_2 - y_3 & \cdots & y_{n-1} - y_n
\end{pmatrix}.
$$

(12)

By writing Equation [12] as $X = J Y$, the Jacobian of $f$ can be obtained by $J = XY^{-1}$. The determinant of the Jacobian matrix can then be computed, which is a constant on each triangular face.

![Figure 5](image_url)

**Figure 5**: Illustration of Fill First, Dig Second strategy. A disk without holes is deformed to enclose the filled label mask. A hole will be dug out and further optimized to locate the inner boundary.
will compare our models with other pixel-wise segmentation networks, such as the baseline U-Net model [27] and the double with other state-of-the-art topology-preserving learning-based models such as TEDSNet[34] and TeTrIs[21]. Additionally, we will compare our methods with other state-of-the-art topology-preserving learning-based models such as TEDSNet[34] and TeTrIs[21]. Additionally, we will compare our methods with other state-of-the-art topology-preserving learning-based models such as TEDSNet[34] and TeTrIs[21]. Additionally, we will compare our methods with other state-of-the-art topology-preserving learning-based models such as TEDSNet[34] and TeTrIs[21]. Additionally, we will compare our methods with other state-of-the-art topology-preserving learning-based models such as TEDSNet[34] and TeTrIs[21]. Additionally, we will compare our methods with other state-of-the-art topology-preserving learning-based models such as TEDSNet[34] and TeTrIs[21].

With the above discretization, the Beltrami coefficient of $f$ can also be computed. Note that the Beltrami coefficient is defined by the first derivatives of $f$, which are constants on each triangular face. As such, the Beltrami coefficient is a constant complex number on each triangular face. More specifically, the Beltrami coefficient can be computed by

$$
\mu = \left( \frac{\partial f_x}{\partial x} - \frac{\partial f_y}{\partial y} \right) + i \left( \frac{\partial f_x}{\partial y} + \frac{\partial f_y}{\partial x} \right)
$$

$$
= \left( \frac{a_1^x - a_2^x}{a_1^x + a_2^x} + i \frac{a_1^y + a_2^y}{a_1^y + a_2^y} \right)
$$

The Jacobian determinant and the Beltrami coefficient provides information about the change of orientation on each triangular face under $f$. Figure 7 illustrates how the orientation of a map can be described by the Jacobian determinant. A triangle with the orientation "Green-Blue-Yellow" clockwise in (a) is transformed into the first three triangles in (b), which are all orientation-preserving. In these cases, the Jacobian determinants of the deformation maps are all positive and the magnitudes of the Beltrami coefficients are all less than 1. The triangle in (a) is further transformed into the last three triangles, which are orientation-reversing. In these cases, the Jacobian determinants of deformation maps are all negative and the magnitudes of the Beltrami coefficients are all greater than 1.

B. Reconstruction of quasiconformal mappings

Given a piecewise constant Beltrami coefficient $\mu$, the associated piecewise linear mapping $f$ can be reconstructed by solving a linear system, called the Linear Beltrami Solver [23]. Given a triangular face $T$, we denote $\mu(T)$ by $\mu(T) = \mu_T = \rho_T + \sqrt{-1} \tau_T$. On each $T$, the Beltrami’s equation can be written in a matrix form:

$$
\begin{pmatrix}
-a_2^y \\
a_1^y
\end{pmatrix} = \frac{1}{1 - \rho_T^2 - \tau_T^2} \begin{pmatrix}
\alpha_1(T) & \alpha_2(T) \\
\alpha_2(T) & \alpha_3(T)
\end{pmatrix} \begin{pmatrix}
\alpha_1^x \\
\alpha_2^x
\end{pmatrix},
$$

where $\alpha_1(T) = -(1 - \rho_T)^2 - \tau_T^2$, $\alpha_2(T) = -2\tau_T$ and $\alpha_3(T) = -(1 + \rho_T)^2 - \tau_T^2$. Applying the discrete divergence $Div$ on both sides, we obtain a linear system

$$
A_\mu f = b,
$$

where $f$ is a vector whose $i$-th entry is given by $f(v_i)$ of vertices $v_i$, $A_\mu$ is a matrix depending on $\mu$ and $b$ is a vector depends on $\mu$ and the boundary conditions. As such, $f$ can be reconstructed from $\mu$ by solving Equation 15. The representation of $f$ by $\mu$ is beneficial since $\mu$ measures the local geometric distortion under $f$. We can thus easily control the regularity of $f$ by controlling $\mu$.

To facilitate the optimization process in the Beltrami Adjustment Module, we apply the BS-Net [7], which is a neural network to output $f$ from an input $\mu$. Given a pre-trained and frozen BS-Net, the backward propagation can then be carried out effectively during the training process.

V. EXPERIMENT

In this section, we will evaluate the efficiency of our proposed models through various experiments. To systematically examine the impact of different parameters and the influence of the template mask shape, we will conduct some self-ablation studies. Our proposed segmentation model will be tested on multiple real image datasets. Moreover, we will compare our methods with other state-of-the-art topology-preserving learning-based models such as TEDSNet[34] and TeTrIs[21]. Additionally, we will compare our models with other pixel-wise segmentation networks, such as the baseline U-Net model [27] and the double U-Net model [17]. Finally, we will report quantitative measurements of the segmentation results.

To ensure a fair comparison, all experiments are conducted on the same platform and under identical training settings, including batch number and running epochs. The backbone networks used for each method have the same number of layers to
We used the KiTS21\[ 13\] dataset for 3D segmentation of simply-connected structures, following the same process as in the

Dataset In this work, we utilized four datasets to evaluate the effectiveness of our proposed models. The 3D volumetric dataset of kidney images \[13\] was used for the self-ablation study. Out of the 300 images in the dataset, 210 were used for training, and the remaining 90 were used for testing. All images were central-cropped and resized to \(128 \times 128 \times 64\). To evaluate the unsupervised TPSN model for segmenting skin lesions\[33\], \[32\], we used the Ham10000 dataset, considering only the simply-connected lesions. We removed the images containing disconnected lesion regions in the original dataset, and the final dataset comprised 9981 images. Out of these, 8981 were used for training, and the remaining 1000 images were used for testing. All images were resized to \(256 \times 256\). To evaluate our model for segmenting genus-1 structures, we used the ACDC\[ 2\] dataset, where 5 slices containing circular vessel structures from 100 cases were extracted and used as the dataset. Out of these, 90 cases comprising 450 images were used for training, while the remaining 10 cases (50 images) were used for testing. We used the KiTS21\[13\] dataset for 3D segmentation of simply-connected structures, following the same process as in the self-ablation study. Finally, we evaluated the capability of our methods for multi-object segmentation on BTCV dataset\[20\]. We divided the whole 30 cases in the dataset into 25 and 5 for training and testing, respectively, and used only 8 organs (spleen, right kidney, left kidney, gallbladder, liver, stomach, right adrenal gland, left adrenal gland) in this work. All images were centrally cropped and resized to \(128 \times 128 \times 128\). The intensity of all images was normalized to \([0, 1]\).

Computing resources and Parameters All models were trained for 300 epochs using RMSprop optimizer with a learning rate of \(1 \times 10^{-5}\). For experiments on KiTS21 and BTCV, a batch size of 8 was used, while a batch size of 16 was used for experiments on ACDC. The models were trained on a CentOS 8.1 central cluster computing node with two Intel Xeon Gold 5220R 24-core CPUs and one NVIDIA A100 Tensor Core GPUs. In the supervised segmentation, the weighting parameters \(\lambda_{Jac}\) and \(\lambda_{Lap}\) were set to be 1.0 and 0.01, respectively, unless otherwise specified.

Metrics The following metrics will be used for quantitative measurements. Denote the domain in the predicted mask and in the label mask by \(X\) and \(Y\), respectively.

- *Dice score* is defined as \(DSC = \frac{2|X \cap Y|}{|X| + |Y|}\). The unit for Dice in the tables are \(\%\).
- *Hausdorff distance* is defined as \(d_H(X, Y) = \max\{\sup_{x \in X} d(x, Y), \sup_{y \in Y} d(X, y)\}\).
- *Betti error* used in this paper is defined as the \(0\)-dimensional Betti number \([14], [9]\). We use 8-connected component labeling algorithm to count for the regions.
- \(L_{Jac}\) reported in the tables is the value of ReLU-Jacobian loss computed as Equation \[1\]. It is zero for any bijective mappings.
- *BettiMatching Error* is a topologically and feature-wise accurate metric proposed in \([31]\).

A. Ablation Studies on DEN

We conducted ablation studies to explore the impact of various parameters and evaluate the influence of the template mask in the DEN module of the proposed model. All the networks use a single-level setting except the experiment evaluating the influence by the number of levels. The template mask for experiments on \(\lambda_{Jac}\), \(\lambda_{Lap}\) and No. of levels are the mean shape over all the label masks in the training set calculated by \([22]\). The weighting parameters \(\lambda_{Jac}\) and \(\lambda_{Lap}\) were set to be 1.0 and 0.01 unless otherwise specified.

Influence of \(\lambda_{Jac}\) on DEN: We begin by examining the significance of our proposed Jacobian regularization term, denoted as \(L_{Jac}\). As all mapping adjusted by BAM will be bijective, we conducted experiments with varying weighting parameters \(\lambda_{Jac}\) minimize the impact of other factors. In addition, all methods employ Dice loss as the fidelity term for supervised training, except for TopoLoss \([14]\) and BettiMatching \([31]\), which introduced their own fidelity loss functions. No supplementary data preprocessing or postprocessing is employed for any of the methods, facilitating a more transparent evaluation of the progress achieved by each approach. The objective of these experiments is not to attain a higher score compared to state-of-the-art methods through data manipulation but rather to assess our methods’ capacity to preserve topology in segmentation.

Figure 7: Illustration of deformations with different values of Jacobian and the norm of Beltrami Coefficients. A positive Jacobian determinant (or a Beltrami coefficient whose norm is less than 1) represent an orientation-preserving deformation. A negative Jacobian determinant (or a Beltrami coefficient whose norm is greater than 1) represents an orientation change in the mapping.
Figure 8: Segmentation results comparing different methods for the ACDC dataset. Conventional pixel-wise approaches often produce isolated noisy regions (indicated by yellow arrow) and discontinuity errors (indicated by blue arrow). Other state-of-the-art deformable models may result in suboptimal outcomes due to over-constraint.

Figure 9: The convergence analysis was performed using different values of $\lambda_{Jac}$. The results demonstrate that larger values of $\lambda_{Jac}$ slow down the training process, but the convergence is still acceptable, typically within 100 epochs. However, when $\lambda_{Jac}$ exceeds 10, the segmentation results become less accurate due to over-constraint.

Influence of $\lambda_{Lap}$ on DEN: When $\lambda_{Lap}$ is less than $10^{-2}$, the mapping is not bijective as reflected by the non-zero value of $\mathcal{L}_{Jac}$ and the supremum of $\mu$ being greater than 1. The weighting parameter can vary widely according to Table I. However,
when $\lambda_{\text{Lap}} = 10^4$, the problem becomes over-constrained, resulting in significant degradation of the segmentation result as indicated by the Dice score.

**Influence of levels:** We evaluate the impact of the number of levels used in the multi-level TPSN by setting $\lambda_{\text{Jac}} = 1.0$ and $\lambda_{\text{Lap}} = 10^{-2}$. The segmentation results are reported in TABLE I and it is evident that the mlTPSN with varying numbers of levels can yield segmentation results with correct topology. The results show improvement from 1 to 3 levels, with only slight differences between 3 and 4 levels. Thus, in the after experiments, we use a 3-level configuration for mlTPSN.

**Influence of template mask** We also explore the robustness of our model under various template masks. To this end, we conducted experiments with four different template shapes: (i) a disk, (ii) an ellipse, (iii) a square, and (iv) the average shape from a dataset. We selected the Ham10000 dataset for this study and the quantitative results are reported in TABLE I. The results indicate that the segmentation outcomes obtained using different template synthetic shapes (i.e., shapes (i), (ii), and (iii)) are similar. However, the segmentation results achieved using the mean shape are the best, demonstrating that a template shape closer to the ground truth can enhance segmentation accuracy, as expected.

**Influence of Inputs** We conducted ablation studies on the input, specifically exploring scenarios where only the image or both the image and template were used as input. When using the same template mask as input, we excluded the direct input of the template information into the network. Even without the input of the template information, the network learned to warp the template into the region of interest during training. Therefore, whether we explicitly provided the template mask as an input to the network or not did not make a difference, as evidenced by the results shown in Table I.

### Table I: Quantitative measurements for the self-ablation studies.

| Parameter | Value | Dice | HD | Betti | $L_{\text{Jac}}$ | $\|\mu^i\|$ |
|-----------|-------|------|----|-------|------------------|----------------|
| $\lambda_{\text{Jac}}$ | 0 | 93.01 | 4.34 | 1.27 | 27.34 | 159.56 |
| | $10^{-2}$ | 92.96 | 4.12 | 1.08 | 0.25 | 12.47 |
| | $10^{-3}$ | 92.87 | 4.08 | 1.03 | 0.03 | 3.82 |
| | 1 | 92.65 | 3.86 | 1.00 | 0.00 | 0.48 |
| | $10^1$ | 92.88 | 3.84 | 1.00 | 0.00 | 0.43 |
| | $10^2$ | 89.11 | 4.41 | 1.00 | 0.00 | 0.55 |
| $\lambda_{\text{Lap}}$ | 0 | 92.72 | 3.97 | 1.03 | 0.07 | 4.65 |
| | $10^{-4}$ | 92.69 | 3.92 | 1.00 | 0.03 | 1.58 |
| | $10^{-2}$ | 92.65 | 3.86 | 1.00 | 0.00 | 0.48 |
| | 1 | 92.60 | 3.89 | 1.00 | 0.00 | 0.43 |
| | $10^2$ | 92.63 | 3.87 | 1.00 | 0.00 | 0.47 |
| | $10^4$ | 90.87 | 4.13 | 1.00 | 0.00 | 0.38 |
| No. of levels | 1 | 92.65 | 3.86 | 1.00 | 0.00 | 0.48 |
| | 2 | 93.21 | 3.82 | 1.00 | 0.00 | 0.61 |
| | 3 | 93.63 | 3.79 | 1.00 | 0.00 | 0.59 |
| | 4 | 93.71 | 3.79 | 1.00 | 0.00 | 0.43 |
| Shape | Disk | 92.49 | 3.91 | 1.00 | 0.00 | 0.37 |
| | Ellipse | 92.51 | 3.90 | 1.00 | 0.00 | 0.42 |
| | Square | 92.38 | 3.98 | 1.00 | 0.00 | 0.48 |
| | Mean | 92.65 | 3.86 | 1.00 | 0.00 | 0.48 |
| Input | Only $I$ | 92.63 | 3.88 | 1.00 | 0.00 | 0.43 |
| | $I$ and $M_{\text{temp}}$ | 92.65 | 3.86 | 1.00 | 0.00 | 0.48 |

B. Segmentation of real images

In this subsection, we evaluate the performance of our proposed models for segmenting real images, including both supervised and unsupervised image segmentation.

1) Supervised segmentation: We evaluated the efficiency of our proposed model for supervised segmentation by utilizing two image datasets, namely the ACDC2D and KiTS21, both of which have labeled segmentation masks. Additionally, we conduct a thorough comparison of our segmentation results with those of other state-of-the-art methods.

2D image segmentation We applied our proposed segmentation framework to the ACDC2D dataset to extract doubly-connected structures from 2D images. We compare our results with several state-of-the-art methods, including two non-deformation-based baseline models: UNet [27] and doubleUnet [17]; two methods utilizing their proposed topology losses: TopoLoss [14] and BettiMatching [31]; as well as two learning-based deformation-based models: TEDSNet [34] and TeTrIs [21]. The template mask is the mean shape over all the label masks in the training set calculated by [22].

Besides those methods, the FFDS strategy described in Section II-E is applied to all deformable approaches in the comparison. In particular, the first phase is executed during the initial 50 epochs to delineate the outer boundary, while the remaining epochs focus on approximating the inner boundary. Additionally, we report results obtained without employing the FFDS strategy to underscore its advantages.

Figure 8 illustrates the segmentation results obtained using different methods. It is evident that methods lacking topology regularization, such as UNet and doubleUnet, often produce incorrect isolated components (highlighted by the yellow arrow)
that are located far away from the object to be segmented. In contrast, the utilization of TopoLoss [14] and BettiMatching [31] demonstrates better topology control. However, their regularization, implemented as soft constraints, can still result in incorrect topology. TeTrIs [21], which performs segmentation by generating a mapping to warp a template mask, fails to preserve topology as the predicted mapping with a small Laplacian does not indicate bijectivity. Although TEDSNet successfully preserves topology, the generated mapping may be too restrictive to accurately deform the template to enclose the region of interest. Our methods, depicted in the last two columns, consistently yield topology-preserved segmentation results.

These results are further supported by the quantitative findings reported in Table II. Both TPSN and multi-level TPSN consistently produce results with correct topology, as indicated by the predicted mapping possessing a positive Jacobian (as reflected by the null value of the ReLU-Jacobian error). Without noisy outliers, our method outperforms others in terms of Hausdorff distance. In terms of the Dice score, our method achieves comparable results to pixel-wise classification methods. Furthermore, our methods outperform others in terms of BettiMatching error [31], highlighting the advancement of our approach.

Table II: Quantitative comparison with other methods for 2D double-connected component segmentation on the ACDC2D dataset.

| Method     | Dice | HD | Betti | BettiMatching | \(\mathcal{L}_{Jac}\) |
|------------|------|----|-------|---------------|-----------------|
| UNet       | 84.31| 2.46| 1.08  | 2.28          |                 |
| wUNet      | **86.10**| 2.26| 1.04  | 2.22          |                 |
| TopoLoss   | 82.85| 2.73| 1.02  | 2.14          |                 |
| BetMatCH   | 82.68| 2.66| 1.00  | 2.10          |                 |
| TeTrIs     | 79.86| 5.33| 1.14  | 2.54          | **7.49**        |
| TEDS       | 78.90| 4.80| **1.00**| 2.28          | 1.27            |
| TeTrIs(FFSD)| 80.26| 3.54| 1.04  | 2.15          | **2.49**        |
| TEDS(FFSD) | 81.54| 3.47| **1.00**| 2.11          | 1.46            |
| TPSN       | 82.75| 2.37| **1.00**| 2.06          | 0.00            |
| mTPSN      | 85.15| **2.04**| **1.00**| **2.02**      | 0.00            |

Table III: Quantitative comparison with other methods for 3D single-connected component segmentation on the KiTS21 dataset.

3D image segmentation We evaluated our segmentation framework on the KiTS21 dataset to segment organs from 3D volumetric images. Specifically, we focused on segmenting the kidney, a simply-connected organ with a single component, using a 3D simply-connected cube as the template mask. In this experiment, only the left kidney with labels was utilized, and the quantitative results are detailed in Table III.

We observed that pixel-wise segmentation methods generally achieve higher Dice scores. This can be attributed to the inherent flexibility of pixel-wise classification approaches. However, these methods often suffer from topological errors and produce noisy isolated regions, resulting in higher Hausdorff distances. Among deformable approaches, TeTrIs performs relatively well in terms of Hausdorff distance but struggles to preserve the topology of the prior template mask. On the other hand, TPSN excels in topology preservation but exhibits significantly lower Dice scores and Hausdorff distances, possibly due to over-constrained conditions.

In contrast, our method capitalizes on the bijectivity of the predicted mapping for warping the template masks. This is evident from the zero value of \(\mathcal{L}_{Jac}\), indicating that our method is free from topological errors. Moreover, our method achieves...
Table IV: Quantitative comparison for unsupervised TPSN using the selected HAM10000 dataset.

| Method    | Time (s) | Dice   | HD     | Betti | $\mathcal{L}_{Jac}$ | $\|\mu\|$ |
|-----------|----------|--------|--------|-------|----------------------|---------|
| Zhang’s   | 2.87     | 87.99  | 9.69   | 1.00  | 0.00                 | 0.52    |
| unTPSN    | 0.02     | 88.15  | 9.94   | 1.00  | 0.00                 | 0.59    |

Figure 11: 3D segmentation results of the KiTS21 dataset with missing slices due to information loss. Pixel-wise segmentation methods like UNet produce two separate parts, as shown in the image. This issue cannot be resolved using connected component analysis, as demonstrated.

Figure 12: Qualitative results of unTPSN. The second row shows the ground truth, the third row shows the segmentation results obtained by our proposed model, and the last row shows their corresponding deformation maps. The Dice scores are reported at the bottom left, indicating that the segmentation results achieved by our proposed method are satisfactory and comparable to those obtained by the optimization approach.

The convergence of different models in terms of different metrics was also analyzed, as shown in Figure 10. The pixel-wise segmentation model shows faster convergence, while Dice loss (which is $1 - \text{Dice Score}$) converges the best due to the flexibility provided by pixel-wise classification. Among deformable methods, our model achieves the best convergence regarding Dice loss, while TEDSNet has the slowest convergence among the four methods. In terms of Hausdorff distance, we observe that all deformable segmentation models converge in a similar gradual pattern, while the pixel-wise segmentation model shows an unstable, non-convergent curve.

We also studied the tolerability of our model to information loss after sufficient training. We assumed that some slices of the scan were missing, resulting in missing information. The incomplete scans were then fitted into the trained network without processing. The segmentation results obtained by different methods are shown in Figure 11. Our method produces an accurate segmentation result with the correct topology even with missing information. In contrast, the segmentation result by UNet has two separate components, and using connected component analysis to correct the topology in such a case cannot solve the issue.

2) Unsupervised segmentation: We also evaluated the performance of our proposed unsupervised image segmentation model, which does not require labeled masks for training. We conducted experiments on a selected Ham10000 dataset and predicted the segmentation mask using the unsupervised loss (Equation (4)). Taking a disk whose radius is a quarter of the image sizes as the template mask, the segmentation results are reported in Table IV. Notably, our proposed unsupervised segmentation model takes only 0.02 seconds to obtain the segmentation result upon successful training, while the optimization approach [39] requires 2.87 seconds on average to segment a single image. The qualitative results of unTPSN are presented in Figure 11, where the second row shows the ground truth, the third row shows the segmentation results obtained by our proposed model and the last row shows their corresponding deformation maps. The Dice scores are reported at the bottom left, indicating that the segmentation results achieved by our proposed method are satisfactory and comparable to those obtained by the optimization approach.

C. Multi-object Segmentation

All of the previous experiments focus on the segmentation of a single structure in the image. In this subsection, we examine the performance of our proposed model for multi-object segmentation, which is a fundamental task in medical imaging. We used the BTCV dataset from the MICCAI 2015 Multi-Atlas Abdomen Labeling Challenge [20] for this experiment. The template masks for multiple objects used in this experiment were the bounding boxes for the organs, which can be extracted by detection networks [1]. To provide the same prior information for U-Net and double U-Net, we concatenated the template masks with the original image as the input.

The quantitative results, as shown in Table V, highlight the effectiveness of our method, which surpasses all other models. When using pixel-wise classification approaches, methods like UNet and doubleUNet face challenges in accurately preserving topology for all organs, even with prior input information. Notably, TeTrIs [21] fails to maintain topology, as evidenced by the Betti number errors. Additionally, the mapping generated by TeTrIs is not guaranteed to be bijective, as indicated by the non-zero ReLU-Jacobian errors. This emphasizes the inadequacy of relying solely on Laplacian regularization to ensure mapping bijectivity. In contrast, our approach consistently produces segmentation results with correct topology and achieves superior Hausdorff distances for most organs.
Figure 12: The segmentation results for the selected HAM10000 dataset using the unsupervised TPSN (unTPSN). First row: the input image; second row: the ground truth; third row: the predicted mask by unTPSN; fourth row: the deformation maps.

| Metric | Method      | Ave  | Spl  | Kid(R) | Kid(L) | Gall | Liv  | Sto  | AG(R) | AG(L) |
|--------|-------------|------|------|--------|--------|------|------|------|-------|-------|
| Dice   | UNet        | 62.90| 82.59| 86.89  | 82.39  | 36.06| 87.96| 52.86| 42.87 | 31.64 |
|        | wUNet       | 67.69| 83.33| 87.73  | 84.39  | 40.94| 89.41| 61.06| 49.12 | 45.57 |
|        | UNet+prior  | 82.07| 87.60| 88.79  | 87.26  | 78.87| 90.08| 87.62| 65.89 | 70.43 |
|        | wUNet+prior | 82.93| 89.05| 89.45  | 87.76  | 81.61| 91.27| 87.90| 66.42 | 70.00 |
|        | TeTrIs      | 81.79| 86.67| 87.85  | 86.71  | 82.81| 87.71| 86.18| 63.17 | 67.88 |
|        | TPSN        | 82.28| 88.45| 88.96  | 88.95  | 81.07| 89.99| 86.48| 64.98 | 69.34 |
|        | mlTPSN      | 83.49| 89.33| 89.10  | 89.13  | 82.11| 90.43| 87.07| 68.38 | 72.40 |
|        | Betti       | 3.27 | 1.60 | 5.20   | 2.00   | 1.60 | 5.60 | 6.20 | 1.60 | 2.40 |
|        | DoubleUNet  | 3.00 | 1.60 | 4.60   | 2.00   | 1.20 | 4.80 | 7.00 | 1.20 | 1.60 |
|        | UNet+prior  | 2.08 | 1.20 | 1.00   | 1.00   | 1.40 | 7.40 | 2.00 | 1.40 | 1.20 |
|        | TeTrIs      | 1.63 | 1.20 | 1.00   | 1.00   | 1.63 | 1.20 | 1.20 | 1.00 | 1.20 |
|        | TPSN        | 1.00 | 1.00 | 1.00   | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
|        | mlTPSN      | 1.00 | 1.00 | 1.00   | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |

Table V: Quantitative results of multi-object segmentation for the BCTV dataset. Ave: averaged over all organs, Spl: spleen, Kid(R): right kidney, Kid(L): left kidney, Gall: gallbladder, Liv: liver, Sto: stomach, AG(R): right adrenal gland, AG(L): left adrenal gland.
Figure 13: Visual comparison of Unet, TEDSnet, TeTrls, and our proposed DEN (TPSN without BAM) and TPSN. The images are degraded by: image blurs, metal artifacts, and noises. The predicted masks obtained methods are indicated by green contour while the ground truth is in red.
D. Segmentation for Corrupted Data

Our framework incorporates two crucial components: the Deformation Estimation Network (DEN) and the Beltrami Adjustment Module (BAM). DEN is trained on data to predict the mapping required to warp the template mask, while BAM serves as a post-processing module to ensure that the mapping generated by DEN is fully bijective. Under normal circumstances, DEN is effective in producing a bijective mapping that can accurately warp the template mask into the target region while preserving topology. However, it may encounter challenges in maintaining this essential property when presented with input data that falls outside the distribution of the training dataset. For example, if a network is trained exclusively on high-quality images, it may struggle to generate a bijective mapping for a low-quality image with significant noise. In such scenarios, BAM comes into play by adjusting the mapping $f$ produced by DEN to a bijective mapping $\tilde{f}$ while minimizing the discrepancy between the newly warped mask $M_{\text{temp}} \circ \tilde{f}$ and the original warped mask $M_{\text{temp}} \circ f$ generated by DEN. As a result, our framework consistently produces segmentation results that preserve topology, even under challenging circumstances.

To evaluate our method, we synthesized corrupted data that contains three types of image degradations commonly encountered in medical imaging. The first type is blurred images, which may result from unstable equipment settings or patient movement during imaging. The second type is images with Gaussian noise, which may be generated by signal noise in imaging machines. The third type is images with metal artifacts that may arise from in-vivo medical devices used by patients.

In Figure 13 we present the segmentation results obtained by our proposed framework and other state-of-the-art deformation-based segmentation models as well as the backbone UNet. The first two rows show the segmentation results of four blurry images. The red regions represent the ground truth, while the green contour shows the predicted mask obtained by different models. Our proposed method with the BAM yields the best results without topological errors. The segmentation results obtained by only DEN module without BAM may still contain topological errors. The third and fourth rows depict the segmentation results of four images with metal artifacts. Once again, our proposed method with BAM performs the best, while other methods produce mis-segmentation results and contain topological errors. The last two rows show the segmentation results of four noisy images. Our proposed method with BAM yields the best results without topological errors. These results demonstrate the capability of our proposed model to handle corrupted images.

VI. Conclusion

In this work, we propose a novel learning-based segmentation framework for topology-preserving image segmentation, namely the Topology-Preserving Segmentation Network (TPSN). The image to be segmented is first inputted into the Deformation Estimation Network. As a deformable model, the Deformation Estimation Network completes the segmentation task by generating a mapping using an encoder-decoder architecture that transforms the template mask into the region of interest. The mapping is regularized by the proposed ReLU Jacobian regularization, which encourages the orientation-preserving property of the mapping. The mapping is further fed into the Beltrami Adjustment Module to ensure bijectivity, thereby preserving the topology of the transformed mask. Our proposed learning model can be either a supervised model or extended to an unsupervised segmentation model using an intensity-based unsupervised loss function.

Extensive experiments were conducted to demonstrate the effectiveness of our proposed framework. We conducted self-ablation studies to analyze the effects of various parameters in the framework. To evaluate the capability of our model in different dimensions, we performed experiments on both 2D planar images and 3D volumetric images. We also conducted experiments to segment structures with different topologies. Compared with other state-of-the-art methods, our proposed method showed comparable results by Dice score and often outperformed others by geometric measures like Hausdorff distance. Furthermore, with the carefully designed ReLU Jacobian term and Beltrami Adjustment Module, our method consistently predicted segmentation masks with the correct topology.

In future work, we plan to investigate the use of more geometric constraints in our topology-preserving segmentation network. Specifically, we plan to explore the use of convexity as a constraint in the segmentation process. Additionally, we will consider incorporating some explicit geometry features such as the shape signature to further improve the accuracy of the segmentation. Moreover, while our current TPSN method provides accurate segmentation results, it is limited by the fixed topology of the input template mask. In future work, we plan to extend our method to incorporate interactive segmentation, allowing users to modify the topology of the segmentation mask interactively in real-time. This will enable more flexible and user-friendly segmentation.

ACKNOWLEDGMENT

This work was supported by HKRGC GRF (Project ID: 14307622), and Hong Kong Centre for Cerebro-Cardiovascular Health Engineering (COCHE).

REFERENCES

[1] Michael Baumgartner, Paul F Jäger, Fabian Isensee, and Klaus H Maier-Hein. nnDetection: A self-configuring method for medical object detection. In Proc. of Int. Conf. on Medical Image Computing and Computer-Assisted Intervention, pages 530–539. Springer, 2021.
[38] Daoping Zhang and Lok Ming Lui. Topology-preserving 3d image segmentation based on hyperelastic regularization. *Journal of Scientific Computing*, 87(3):1–33, 2021.

[39] Daoping Zhang, Xue-cheng Tai, and Lok Ming Lui. Topology-and convexity-preserving image segmentation based on image registration. *Applied Mathematical Modelling*, 100:218–239, 2021.

[40] Yuyin Zhou, Zhe Li, Song Bai, Chong Wang, Xinlei Chen, Mei Han, Elliot Fishman, and Alan L. Yuille. Prior-aware neural network for partially-supervised multi-organ segmentation. In *Proc. of Int. Conf. on Computer Vision*, pages 10672–10681, 2019.