Abstract

Deep Learning based segmentation models for medical imaging often fail under subtle distribution shifts calling into question the robustness of these models. Medical images however have the unique feature that there is limited structural variability between patients. We propose to exploit this notion and improve the robustness of deep learning based segmentation models by constraining the latent space to a learnt dictionary of base components. We incorporate a topological prior using persistent homology in the sampling of our dictionary to ensure topological accuracy after composition of the components. We further improve robustness by deep topological supervision applied in an hierarchical manner. We demonstrate the effectiveness of our method under various perturbations and in two single domain generalisation tasks.

1. Introduction

Robust image segmentation is essential for the safe translation of deep learning based segmentation models into critical applications such as clinical decision making. This is particularly relevant in the medical domain, where images have varying noise profiles and appearance shifts induced by different acquisition protocols across multiple source domains [10]. It is well documented that deep learning models can fail under subtle perturbations in the input space [6, 32] especially when distributional shifts in the test data are not accessible in the training phase.

There are various strategies used to improve model robustness by learning generalisable features. The two most common approaches in the literature either employ data augmentation strategies [14, 49, 50, 52] or adversarial training [9, 30, 45]. Self-supervised learning strategies have also recently gained in popularity and have shown to improve model generalisability [7]. A school of thought which is particularly applicable to segmentation is that shape features representative of the structural content in an image are invariant across shifts in the input space. Therefore, a group of methods which aims to disentangle shape from textural (style) features have been promising to increase model robustness [11, 34, 48].

The task of medical image segmentation is unique in that there is limited spatial variation among subjects due to anatomical consistency. With this in mind, we propose to constrain the lower dimensional shape representational features in a deep learning based segmentation model to a dictionary of components which is sampled in a topology preserving manner. The dictionary is learnt by discretising the features in the latent space using vector quantisation [43, 46]. We use an application of algebraic topology popular in topological data analysis called persistent homology [2, 3] to sample components from the dictionary such that the components are composed together correctly like a jigsaw to form the segmentation outputs. In the multi-label segmentation setting, we propose to enforce the topological constraints in a hierarchical manner so that high abstraction spatial components fit together to form the class segmentation outputs i.e. parts of the heart, which is then composed
to together to form the foreground segmentation i.e. whole heart. We apply our hierarchical topological constraint in a deeply supervised manner. The contributions of this paper are summarized as follows:

- We propose to improve the robustness of deep learning based segmentation models by constraining the latent space to a learnt dictionary of components.
- This is the first work to enforce a topological prior using persistent homology over the sampling of the learnt dictionary such that the components are not sampled independently (uniform prior) and hence are composed together to create topologically meaningful segmentations in a hierarchical manner.
- This is also the first work to successfully apply persistent homology for deep supervision of deep learning based segmentation models.

2. Background: Persistent Homology

Persistent homology is a popular tool in topological data analysis. We first introduce the simplicial complex, $\mathbb{K}$, a fundamental concept to understanding persistent homology. A simplicial complex, $\mathbb{K} \in \mathbb{R}^n$ is a high dimensional generalisation of a graph consisting of $k$ simplices up to $n$ dimensions [17]. A simplex is an arbitrary dimensional generalisation of a triangle i.e. vertex (0-simplex), edge (1-simplex), triangle (2-simplex) and tetrahedron (3-simplex). Simplicial homology uses matrix reduction algorithms to describe the connectivity of $\mathbb{K}$ as a sequence of mathematical groups denoted as the homology groups [18]. The $n$-dimensional homology group consists of $n$-dimensional topological features such as points ($n = 0$), tunnels ($n = 1$) and voids ($n = 2$). The number of topological features in each group is described in the $n^{th}$ Betti number [17]. This is used to describe the topology of objects. For example in Fig. 1, the Betti numbers of the label segmentation is $(1, 1)$ to represent 1 connected component and 1 tunnel.

In this work, the Betti numbers are useful topological descriptors for the binary labels but not a continuous segmentation output in $\mathbb{R}^{x \times y \times z}$ because Betti numbers are calculated on a single scale. Therefore, a continuous measure of the change in topological features in each homology group at different scales provide a rich and differentiable descriptor of the topology of data. This is described as persistent homology and requires a filtration function, $f$ to track the homology groups over multiple scales, $\epsilon$. A common filtration function is a distance function often used for point cloud data; one refers to this simplicial complex as a Vietoris-Rips complex [17]. Medical imaging is however structured as 2D and 3D grids and therefore the cubical complex, $C$ is naturally equipped to deal with such data. In the cubical complex, cubes and squares are equivalent to tetrahedra and triangles respectively in the simplicial complex The vertices in the cubical complex of a 3D or 2D image, $X$ would correspond to voxels and the edges or connectivity is determined by a grid which connects all voxels in an image.

In this work, we define a filtration based on the voxel values (pixel intensities) in $X$ which represents vertices in the cubical complex. Therefore, a cubical complex is
constructed at a threshold, $\epsilon$ over the output defined as: $\mathcal{C}^\epsilon = \{x \in X | x \geq \epsilon\}$. Given $m$ threshold values between $\epsilon_1$ and $\epsilon_m$, one can construct $m$ cubical complexes from each sub-level set. Thus, this will satisfy a nesting relationship between the cubical complexes of $X$ shown in Eq. (1) making it possible to track changes in the homology groups as $\epsilon_i$ decreases.

$$\emptyset = \mathcal{C}^{\epsilon_1} \subseteq \mathcal{C}^{\epsilon_2} \ldots \subseteq \mathcal{C}^{\epsilon_m} = X \ (1)$$

One can now derive a persistence diagram for $n$ dimensional topological features from tuples $(b, d)$ with $b$ denoting the threshold at which a topological feature is born and $d$ being the threshold at which it dies. In Fig. 1, we overlap the persistence diagrams for 0 and 1 dimensional topological features. In the persistence diagram for the label in Fig. 1, $\epsilon_i$ decreases from 1 to 0. Here, the red circle with coordinates $(0, 1)$ signifies the birth of a single connected component (the ring) at $\epsilon = 1$ which dies at $\epsilon = 0$. The blue square denotes a tunnel born at $\epsilon = 1$ which also dies at $\epsilon = 0$. The persistence diagram for the output denotes similar topological features but with additional topological features representing noise found close to $(0, 0)$ and $(1, 1)$ due to the continuous output.

3. Related Work
3.1. Domain Generalisation

The goal of domain generalisation is to learn domain invariant features for downstream tasks without access to the target domain. Previous works largely focused on features alignment between multiple source domains to learn more generalisable features [22, 28, 33]. Meta-learning schemes have also been adopted to adapt neural networks between source domains [16, 29, 42].

The task of single domain generalisation (SDG) where one has access to only a single source domain is a much more challenging task which is relatively less explored in the literature. The natural choice to tackle this problem is through aggressive augmentation strategies such as CutOut [14] and MixUp [51]. BigAug [52] showed extensive input augmentation improves medical image segmentation significantly in the SDG setting. JiGen [7] solves JigSaw puzzles as a self-supervised method to improve domain generalisability. M-ADA [37] proposes adversarial data augmentation by using a Wasserstein Autoencoder to synthesise new domains in a meta-learning scheme. AdvBias [9] applies adversarial data augmentation specifically to MRI data in the input space by learning to generate bias field deformations. RandConv [50] proposes an interesting approach of using randomised convolutions to improve the robustness of convolutional neural networks (CNNs).

Compositionality has been incorporated into neural networks for image classification [26] and generation [1]. Compositional neural networks have also shown to improve robustness of CNNs for image classification under partial occlusion [26] and more recently for medical image segmentation [43].

3.2. Persistent Homology in Deep Learning

Persistent homology for deep representation learning is limited. Topological autoencoders [31] is the first work however to incorporate persistent homology to preserve the topological structure of the data manifold in the latent representation. Attempts have also been made to incorporate topological signatures into deep neural networks [21]. For example, [21] takes a topological signature in the form of a persistence diagram as input into a deep neural network using a novel input layer. They showed SOTA results for graph and shape classification. Persistent homology has also been used as a complexity measure to analyse deep neural network architectures [38].

The application of persistent homology to deep learning based segmentation is limited to the output space to produce topologically meaningful segmentations [23] or as a post-processing method [13]. Cubical persistent homology [15, 47] has recently been used to analyse fMRI data [39].

4. Methods
4.1. Compositionality for Segmentation

We firstly assume a segmentation network can be decomposed into an encoder ($\Phi_e$) to map the input space to a lower dimensional embedding space ($\Phi_e: X \rightarrow \mathcal{E}$) and a decoder ($\Phi_d$) which maps the embedding space to the segmentation output ($\Phi_d: \mathcal{E} \rightarrow \mathcal{Y}$).

We claim the output space of a segmentation model for medical imaging is highly correlated across subjects. Hence, one can hypothesise the low dimensional embedding features to have low variance across the sampled distribution. We therefore propose to constrain the embedding space to a set of $N$ discrete points required to capture the entire training and test distribution. We assume, the area encompassed by an arbitrary radius around a discrete point only represents shifts of the point due to perturbations in the input space. We also assume each discrete point, $\epsilon_i$ is a component representing a spatial structure which make up the set of $N$ components in a dictionary $\mathcal{D}$ shown in Fig. 2.

$\mathcal{D}$ is learnt through vector quantisation as proposed by [46] to discretise the latent space. Prior to the embedding features passing through the quantisation block, we apply a $1 \times 1$ convolutional layer to reduce the number of features for quantisation. The quantisation process aims to collapse the continuous embedding space $\mathcal{E}$ to a set of discrete vectors. Given $m$ embedding feature vectors, this is achieved by minimising the euclidean distance between $z_i \in \mathcal{E}$ and its nearest component $e_k \in \mathcal{D}$ where
$k = \text{argmin}_j \| z_i - e_j \|_2$, formally defined in Eq. (2). However, in order to backpropagate through the sampling process and update $\hat{z}$ and $\mathbb{D}$, we apply straight-through gradient approximation. Therefore, stop gradients (sg) are applied to the appropriate operand during optimisation. We use a $\beta$ value of 0.2 [46].

$$L_{Quant} = \frac{1}{m} \sum_{i=0}^{i=m-1} (\|sg(z_i) - e_k\|_2 + \beta \| z_i - sg(e_k) \|_2)$$

The sampling of the nearest component in $\mathbb{D}$ leads to a quantised embedding space denoted $\hat{z}$ where $\hat{z} = e_k$. $\hat{z}$ then undergoes scaling before passing through a softmax function to form $\tilde{z}$. There are $c$ compositions corresponding to the number of classes in $O^1$ as shown in Fig. 1 where $c = 3$ (background omitted). Therefore, $\tilde{z}$ is subdivided into $m/c$ components which are then summed to form $O^1_k$ (class segmentation outputs from the bottleneck of the model: level 1). A post quantisation $1 \times 1$ convolutional layer is applied to $\tilde{z}$ to increase the number of channels before being passed into the decoder.

### 4.2. Topology Preserving Compositionality

The quantisation process assumes a uniform prior such that components in $\mathbb{D}$ are sampled independently. Therefore, we propose to incorporate a topological prior into the sampling process such that components are sampled and composed together to form topologically accurate shapes i.e. segmentation maps. In this work, the composition of components is simply the summation over the components. We impose three restrictions in order to preserve topology. We flattened each feature in $\tilde{z}$ into $d$ dimensional vectors.

**Shape Disentanglement:** Firstly, the sampled components should only represent shape features which are distinct from one another so that there is no spatial overlap between the features i.e. disentangled. We can therefore apply a shape disentanglement loss term ($L_{DL}$) forcing the inner product between pairs of $d$-dimensional quantised latent features passed through the softmax function, $(\tilde{z}_i, \tilde{z}_j)$ to be close to 0 (orthogonal) [41] as shown in Eq. (3).

$$L_{DL} = \sum_{i=0}^{i=m-1} \sum_{j=i+1}^{j=m-1} \tilde{z}_i, \tilde{z}_j = 0$$

Due to the nature of the softmax function, the summation over all $m$ features in $\hat{z} \in \mathbb{R}^d$ yields a $d$ dimensional vector of 1s. This means if $\hat{z} \in [0, 1]$ and $L_{DL} = 0$, then this implies that the intersection of the space between $\tilde{z}_i$ and $\tilde{z}_j$ forms an empty set; $\bigcap_{i=0}^{i=m-1} \tilde{z}_i = \emptyset$. We can also infer from this definition that $\tilde{z}_i$ must then only consist of the integers 0 or 1; $\tilde{z}_i \in (\mathbb{Z}/2\mathbb{Z})^d$. For example, if $xy = 0$ and $x + y = 1$ then, $x, y \in \mathbb{Z}/2\mathbb{Z}$. Therefore, by design we additionally remove textual/style information from the semantics of the latent space, by approximately learning binarised feature maps, $\tilde{z}_i$. This means the $k^{th}$ class segmentation output in $O^1$ denoted $O^1_k$ where $k \in \mathbb{Z}/3\mathbb{Z}$ must also be binary as demonstrated in Eq. (4). An overlap between features will yield integers greater than 1 under the stated assumptions. For example, in the scaled (0-1) heat maps in Fig. 2 one can visualise brighter regions in $O^1$ representing overlapping components.

$$O^1_k = \sum_{i=m(k+1)/c}^{i=m(k+1)/c} \tilde{z}_i,$$

$$O^1_k \in (\mathbb{Z}/2\mathbb{Z})^d \ | \ \tilde{z} \in (\mathbb{Z}/2\mathbb{Z})^d, L_{DL} = 0$$

**Preserving Topology:** Next, we must sample components such that the topology of the composed output $O^1_k$ matches the down-sampled label map $L_k$ (see Fig. 2) i.e. same number of connected components and holes. In the toy example highlighted in Fig. 3, the label is a circle and in the top row we show an example where the two sampled components; a ring and semi-circle are composed together to form a shape with a semi-circle shaped hole. The Betti numbers of this shape would be $(1, 1); 1$ connected component and $1$ tunnel while the label has Betti numbers $(1, 0)$ and therefore the homology groups do not match. However, in the bottom row, a ring and smaller circle is composed to form a shape with Betti numbers $(1, 0)$ which matches the label map. Given, we sum over a non-overlapping set of sampled components when $DL = 0$ to form $O^1$, then we must enforce that the union over $\tilde{z}$ should form a single connected component like in Fig. 3 with our topological loss.

We next calculate the persistent homology of the cubical complex of each composition in $O^1$ and $L^1$ denoted, $PH(C(O^1_k))$ and $PH(C(L^1_k))$ respectively. We can then create persistence diagrams for each $k^{th}$ class segmentation in $O^1$, denoted $D^1_{L_k}$ which we aim to match with the persistence diagrams of the down-sampled label map $L^1$ (see Fig. 2). We minimise the $p^{th}$ Wasserstein distance [31] of two persistence diagrams shown in Eq. (5) where $\eta: \mathcal{D} \rightarrow \mathcal{D}'$ is a bijection between the persistence diagrams and $p = 2$. This loss function is proven to be stable to

![Figure 3. Toy example demonstrating the basic principle of TPC.](image-url)
noise [44] and differentiable [8].

\[
d_w(D, D') = \left( \inf_{\eta: D \rightarrow D'} \sum_{x \in D} \| x - \eta(x) \|_\infty \right)^{1/\beta} \tag{5}
\]

Therefore, by minimising the distance \(d_w(D^O_{i,k}, D^L_{i,k})\) in Fig. 2, we ensure the class segmentation outputs at level 1 of the decoder \((O_{1/2})\) all have one single connected component and only \(O^1_k\) has one tunnel.

Additionally, since the label maps are binary, all topological features are born at 1 and die at 0. Therefore, by minimising the Wasserstein distance, one is also forcing \(O^1_k\) to be binary and if \(DL = 0\), then \(\tilde{z}_i\) must also be 0 or 1 by deduction. This topological loss is hence an additional method to remove the textual information in \(\tilde{z}\) which in turn also removes textual information from the dictionary. One can also apply the topological loss to \(\tilde{z}\) directly with the assumption that the topological label for \(\tilde{z}_i\) is a single connected component which is born at 1 and dies at 0. We can then minimise the Wasserstein distance loss for the \(0^{th}\) dimensional homology with this fixed topological label to make sure \(\tilde{z}\) are binary components. We found this beneficial but too computationally expensive. We also didn’t binarise the embedding space via a threshold method such as a steep sigmoid function. This is because of the gradient vanishing either side of the threshold value which are the constant portions in the steep sigmoid function where most of the values will lie. A Dice score loss is added in order to increase the overlap between \(L^1\) and \(O^1\) as the topological loss is position and size invariant.

In order to preserve the topological constraints imposed in the quantised embedding space through the decoder, we apply deep topological and dice supervision at each level of the decoder. Therefore, as shown in Fig. 2, we sum the feature maps outputted at each level of the decoder to produce \(c\) class segmentation outputs before applying our deeply supervised class loss shown in Eq. (6).

**Hierarchical Topology:** We apply an additional topological and dice loss for the foreground segmentation output denoted, \(F^i\) formed from the addition of the foreground class segmentation maps in \(O^1\). This hierarchical loss expressed in Eq. (7) is further enforcing accurate class segmentation maps under the assumption they should be composed to form a topologically meaningful foreground segmentation. For example, in Fig. 2 the foreground segmentation is the whole heart which is forced to be a binary single connected component with no tunnels when the segmented parts of the heart (class segmentations) are combined via summation.

\[
\mathcal{L}_{\text{class,top}} = \sum_{i=1}^{i=p} \sum_{k=0}^{k=c-1} d_w(D^{O}_{i,k}, D^{L}_{i,k}) + \text{Dice}(O^1_k, L^1_k) \tag{6}
\]

\[
\mathcal{L}_{\text{Hier,top}} = \sum_{i=1}^{i=p} d_w(D^F_i, D^{FL}_i) + \text{Dice}(F^i, FL^i) \tag{7}
\]

The total loss for training the topological preserving compositionality (TPC) framework incorporated into a segmentation model is formally defined as:

\[
\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{Quant}} + \mathcal{L}_{\text{Class,top}} + \mathcal{L}_{\text{Hier,top}} + \mathcal{L}_{DL}.
\]

### 5. Experiments

**5.1. Datasets and Training**

We use 3 datasets in our experiments. **Abdomen:** This dataset is the Beyond the Cranial Vault (BTCV) dataset [27] consisting of 30 CT scans with 13 labels acquired from one domain. All images are normalised between 0 and 1 and randomly cropped 96×96×96 patches are used for training.

**Cardiac:** This dataset is the Multi-centre, multi-vendor multi-disease (M&Ms) cardiac imaging 3-class segmentation dataset [5] which is divided into 4 domains determined by the MRI scanner vendor. The end-systole and diastole annotations are available for each patient. There are 95 scans in domain A, 125 scans in domain B and 50 scans in domain C and D. All images were normalised between 0 and 1 and cropped to 288×288.

**Prostate:** We use the NCI-ISBI13 Challenge [4] dataset which consists of 60 scans divided into 2 domains with different MRI scanner types and acquisition protocols. There are two segmentations labels. Each domain consists of 30 scans. We normalised images between 0 and 1 and centre cropped to 256×256.

**Training:** All models are trained with Adam optimisation [25] with a base learning rate of 0.0001 and weight decay of 0.05 for a maximum of 500 epochs on three NVIDIA RTX 2080 GPUs. In order to prevent over-fitting, we apply a simple augmentation scheme for training our method consisting of random rotation and flipping (horizontal and vertical). We evaluate model performance with the Dice score and Betti error [23].

### 5.2. Perturbation Experiments

In the first set experiments we analyse how incorporating TPC into three popular segmentation 3D architectures (UNet [40], nn-UNet [24] and Swin-Unetr [19]) improves performance under various types of perturbations in the input space for the Abdomen dataset. We split the dataset into 18 and 12 for our train and test set respectively. We adjust noise levels between 1 and 30 % for Gaussian, Poisson and Salt and Pepper (S&P) noise. Gaussian Blur is incorporated with a Gaussian kernel which has a window size of 7×7 and variance ranging from 0.1 to 2.0. Random motion blur is applied by using the TorchIO deep learning library [36].
The gamma values ranged from 0.5 to 4.5 for the contrast variations. Finally, for the intensity perturbations, intensity values are scaled by a factor between 0.8 and 1.2. The dice scores and Betti errors are averaged across all parameter values used in the perturbations.

In Tab. 1, we note significant improvement in both evaluation metrics after incorporating TPC in all three models under various perturbations. Our method has a better effect with a convolutional backbone as opposed to the vision transformer (Swin-UNetr) for textural perturbations. This is likely because the coarse attention mechanism is more robust to noise where it was shown that the vision transformer is acting like a low-pass filter to remove noise whereas convolutions are behaving like high-pass filters [35].

We demonstrate the value of TPC with a visual example in Fig. 4 where we show the more visually accurate segmentation masks produced by incorporating TPC in the 3 models. We as expected note greater value of TPC in the CNN models when either 20% Gaussian noise or blur is applied to the image. TPC significantly reduced the number of unconnected components and holes produced by the baseline segmentation models. Additionally, the baseline UNet and nnUNet appear to miss some of the smaller segmentations and nnUNet produced more unconnected components and holes compared to the baseline UNet.

<table>
<thead>
<tr>
<th>Baseline</th>
<th>Gaussian</th>
<th>Poisson</th>
<th>S&amp;P</th>
<th>Blur</th>
<th>Motion</th>
<th>Contrast</th>
<th>Intensity</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNet</td>
<td>0.77±0.08</td>
<td>0.71±0.10</td>
<td>0.72±0.09</td>
<td>0.69±0.06</td>
<td>0.72±0.11</td>
<td>0.63±0.13</td>
<td>0.62±0.14</td>
</tr>
<tr>
<td>+ TPC</td>
<td>0.79±0.04</td>
<td>0.75±0.08</td>
<td>0.77±0.07</td>
<td>0.74±0.08</td>
<td>0.78±0.05</td>
<td>0.67±0.10</td>
<td>0.65±0.08</td>
</tr>
<tr>
<td>nnUNet</td>
<td>0.86±0.05</td>
<td>0.80±0.08</td>
<td>0.81±0.03</td>
<td>0.79±0.07</td>
<td>0.80±0.08</td>
<td>0.73±0.12</td>
<td>0.74±0.11</td>
</tr>
<tr>
<td>+ TPC</td>
<td>0.87±0.04</td>
<td>0.83±0.10</td>
<td>0.84±0.09</td>
<td>0.82±0.11</td>
<td>0.82±0.14</td>
<td>0.75±0.14</td>
<td>0.77±0.08</td>
</tr>
<tr>
<td>Swin-UNetr</td>
<td>0.88±0.05</td>
<td>0.83±0.09</td>
<td>0.80±0.08</td>
<td>0.81±0.07</td>
<td>0.85±0.04</td>
<td>0.75±0.16</td>
<td>0.78±0.19</td>
</tr>
<tr>
<td>+ TPC</td>
<td>0.87±0.07</td>
<td>0.86±0.06</td>
<td>0.84±0.06</td>
<td>0.85±0.08</td>
<td>0.88±0.08</td>
<td>0.87±0.10</td>
<td>0.80±0.06</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dice Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNet</td>
</tr>
<tr>
<td>+ TPC</td>
</tr>
<tr>
<td>nnUNet</td>
</tr>
<tr>
<td>+ TPC</td>
</tr>
<tr>
<td>Swin-UNetr</td>
</tr>
<tr>
<td>+ TPC</td>
</tr>
</tbody>
</table>

Table 1. The mean dice score and Betti error ± standard deviation before and after TPC is applied to 3 segmentation models under various perturbations in the input space. The results for the Abdominal dataset is shown. The baselines refers to no perturbations applied.

5.3. Single Domain Generalisation

In the single domain generalisation (SDG) study we evaluate segmentation performance when testing a trained
Table 2. The average dice score and Betti error ± standard deviations using several single domain generalisation methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Cardiac Dice</th>
<th>Cardiac Betti Error</th>
<th>Prostate Dice</th>
<th>Prostate Betti Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>.67±.17</td>
<td>2.03±.87</td>
<td>.51±.16</td>
<td>2.97±1.15</td>
</tr>
<tr>
<td>CutOut</td>
<td>.69±.14</td>
<td>1.78±.69</td>
<td>.52±.18</td>
<td>2.57±1.03</td>
</tr>
<tr>
<td>AdvBias</td>
<td>.70±.17</td>
<td>1.30±.54</td>
<td>.54±.15</td>
<td>2.03±.88</td>
</tr>
<tr>
<td>RandConv</td>
<td>.71±.14</td>
<td>1.00±.48</td>
<td>.58±.20</td>
<td>1.90±.63</td>
</tr>
<tr>
<td>BigAug</td>
<td>.73±.11</td>
<td>.77±.35</td>
<td>.64±.20</td>
<td>1.39±.41</td>
</tr>
<tr>
<td>Jigen</td>
<td>.69±.16</td>
<td>1.58±.71</td>
<td>.53±.14</td>
<td>1.51±.58</td>
</tr>
<tr>
<td>Ours(TPC)</td>
<td>.74±.10</td>
<td>.54±.21</td>
<td>.62±.11</td>
<td>1.02±.43</td>
</tr>
</tbody>
</table>

Table 3. The average dice score and Betti error before and after TPC is applied to 3 segmentation models after testing on the target domains. The results for prostate and cardiac datasets are shown segmentation model on an unseen target domain for the Prostate and Cardiac datasets. We adopt a cross-validation procedure by training a method on a single source domain and hold out the other domains for testing (3 for cardiac and 1 for prostate).

**SDG method comparison:** In the first set of experiments we use an adapted Residual-UNet architecture with a 2D ResNet-18 backbone as our baseline [20]. We compare our TPC approach with the following single domain generalisation methods: CutOut [14], AdvBias [9], RandConv [50], BigAug [52], Jigen [7] and the baseline model.

Tab. 2 demonstrates that our method achieves the best Betti error scores among the SDG methods for both the Cardiac and Prostate datasets. The significantly greater Betti error scores by our approach is indicative of our method’s superior capability of producing more topologically meaningful segmentations. However, we outperform all methods in the dice metric except for BigAug where we achieve similar scores. The comparable performance of TPC to BigAug shows how a thorough augmentation strategy can tackle SDG. However, we demonstrate without any aggressive augmentation strategies or adversarial training, simply imposing topological constraints into a segmentation model can still achieve similar or better SDG performance. We visually highlight the effectiveness of our method in Fig. 5 where TPC appears to better match the topology of the label for both the cardiac and prostate images. For example, in methods such as CutOut and AdvBias applied for prostate segmentation, there are unconnected components or tunnels (holes) and the segmentation maps appear less smooth.

**Model comparison:** In the second set of experiments, we evaluate the improvement in the segmentation performance in SDG by incorporating TPC into the 2D UNet, nnUNet and Swin-UNetr. In Tab. 3, we note significantly improved domain generalisability of all three models when incorporating TPC. We can demonstrate this visually in Fig. 6 where similar to previous experiments, TPC appears to produce smoother segmentations which are more topologically correct compared to the baseline models. This is especially true for prostate segmentation (bottom row) where there are fewer training examples and a larger domain shift. We further note the incomplete rings formed by the baseline UNet and nnUNet for the myocardium segmentations (top row) which is not the case after applying TPC. This experiment once again highlights the benefit of incorporating topological shape priors into structured segmentation tasks.

**5.4. Ablation Studies**

**Dictionary Experiments:** We want to learn a dictionary which is as sparse as possible without affecting segmenta-
Figure 6. Segmentation of a cardiac slice in the top row and prostate slice in the bottom row when testing on a target domain before and after incorporating TPC into 3 segmentation architectures.

<table>
<thead>
<tr>
<th>Size</th>
<th>Cardiac</th>
<th>Prostate</th>
<th>Abdomen</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensionality</td>
<td>128</td>
<td>64</td>
<td>512</td>
</tr>
<tr>
<td>Channels</td>
<td>324</td>
<td>256</td>
<td>1728</td>
</tr>
</tbody>
</table>

Table 4. Hyper-parameters used in our experiments. The 'size' is the number of components in the dictionary. The 'dimensionality' is the dimension of the codebook components. 'Channels' is the number of embedding features in the model bottleneck.

Figure 7. The mean dice score (left) and Betti Error (right) after each component of TPC is sequentially added (left to right) into the Baseline Model (Tab. 2) to assess SDG segmentation for the Cardiac and Prostate datasets. 'Cls Topology' refers to incorporating the topological loss over each class segmentation. 'Hier Topology' is adding the second topological loss for the foreground segmentation.

Figure 6. Segmentation of a cardiac slice in the top row and prostate slice in the bottom row when testing on a target domain before and after incorporating TPC into 3 segmentation architectures.

Table 4. Hyper-parameters used in our experiments. The 'size' is the number of components in the dictionary. The 'dimensionality' is the dimension of the codebook components. 'Channels' is the number of embedding features in the model bottleneck.

6. Discussion

TPC can also be used as a semi-supervised method when there is unlabelled data given the homology of the class segmentations outputs do not vary between subjects i.e. the two class segmentations of the prostate are always two single connected components with no tunnels. This will be explored in future work. We also aim to test our method in more challenging tasks such as instance segmentation where we believe it will be beneficial due to the ability of TPC to match the number of connected components. We noted this property to be useful in the abdominal segmentation tasks where there were multiple labels.

In conclusion, on the basis that there is limited anatomical variation among subjects in medical imaging, we assume medical image segmentation is structured. We therefore propose to improve the robustness of medical imaging segmentation models by constraining the embedding space to a dictionary of disentangled shape components. We use persistent homology to incorporate a hierarchical topological prior in sampling the dictionary to produce topologically accurate segmentations at multiple scales using deep supervision. We demonstrate the effectiveness of our method by incorporating TPC into 3 common segmentation architectures to improve performance under various perturbations and domain shifts. We finally show that our approach beats various SOTA methods in single domain generalisability.
References


[26] Adam Kortylewski, Ju He, Qing Liu, and Alan L Yuille. Improved regular-