Spatial Correspondence between Graph Neural Network-Segmented Images

Author name(s) withheld EMAIL(S) WITHHELD
Address withheld

Editors: Under Review for MIDL 2023

Abstract

Graph neural networks (GNNs) have been proposed for medical image segmentation, by predicting anatomical structures represented by graphs of vertices and edges. One such type of graphs are predefined with fixed size and connectivity to represent a reference of anatomical regions of interest, thus known as templates. This work explores the potentials in these GNNs with common topology for establishing spatial correspondence, implicitly maintained during segmenting two or more images. With an example application of registering local vertebral sub-regions found in CT images, our experimental results showed that the GNN-based segmentation is capable of accurate and reliable localisation of the same interventionally interesting structures between images, not limited to the segmentation classes. The reported average target registration errors of 2.2±1.3 mm and 2.6±1.4 mm, for aligning holdout test images with a reference and for aligning two test images, respectively, were by a considerable margin lower than those from the tested non-learning and learning-based registration algorithms. Further ablation studies assess the contributions towards the registration performance, from individual components in the originally segmentation-purposed network and its training algorithm. The results highlight that the proposed segmentation-in-lieu-of-registration approach shares methodological similarity with existing registration methods, such as the use of displacement smoothness constraint and point distance minimisation albeit on non-grid graphs, which interestingly yielded benefits for both segmentation and registration. We therefore conclude that the template-based GNN segmentation can effectively establish spatial correspondence in our application, without any other dedicated registration algorithms.

Keywords: Registration, graph neural networks, orthopaedic surgery.

1. Introduction

Graph neural networks (GNNs) provides versatility in representing data sampled from non-grid spatial locations using connected vertices and edges. For medical imaging applications, GNNs have been proposed to represent the input images and extract features for tasks such as classification and recognition, as well as used in decoding for segmentation tasks (Han et al., 2022; Fu et al., 2021), whilst GNNs have also been employed to represent non-grid prediction output for segmentation (Worchel et al., 2022; Munkberg et al., 2022; Wickramasinghe et al., 2020; Kong and Shadden, 2021; Kong et al., 2021; Bongratz et al., 2022). In the latter, graph templates are designed for the regions of interest (ROIs) to segment and are refereed to as template-based GNNs in this paper.

This work focus on annotating spinal vertebra from CT images for planning orthopedic procedures, previously using convolutional neural networks (CNNs) such as variants of UNet (Li et al., 2021; Lessmann et al., 2019). Template-based GNNs have achieved promising

© 2023 Anonymous.

results in other segmentation applications (Worchel et al., 2022; Munkberg et al., 2022; Wickramasinghe et al., 2020; Kong and Shadden, 2021; Kong et al., 2021; Bongratz et al., 2022), compared with CNNs alone. These methods utilised a common surface-representing template, which can be fitted to individual subjects during test time, therefore obtaining segmentation of predefined ROI classes. While, localising finer vertebral sub-regions, including spinous/transverse/articular process and left and right pedicles, are also desirable in a number of surgical tasks, such as pedicle screw fixation. For instance, the screws need to be implanted along the pedicle axis without intruding its surface, to avoid damage to the spinal cord (Hu et al., 2013). Recent robot-assisted surgery may also benefit from precise planning of robotic trajectories, with respect to these local anatomies (Dillon et al., 2016).

One solution could be supervised segmentation which in general requires fully-labelled data sets with multiple local structure classes manually annotated. Alternatively, image registration can be used to establish dense correspondence between the planning image and a small number of pre-segmented images, as in atlas segmentation. Registering inter-subject vertebral images may also help population studies that require spatially normalised individual subjects and comparative analysis on surgical planning, but these potential clinical applications are considered out of the scope of this work.

In this work, we first validate both classical intensity-based and recent learning-based registration algorithms, in the registration-based atlas segmentation of local sub-regions. Moreover, we propose template-based GNNs to represent vertebra segmentation output and infer the spatial correspondence from the segmented vertebrae, a denser, more local correspondence between sub-regions without supervision other than the corresponding segmentation classes (the entire vertebra versus background in this case). This is enabled by the spatial connectivity from the GNNs, inherent within the common template. Interestingly, the experiments show that graph-segmentation-derived dense correspondence achieved significantly lower target registration errors (TREs), compared with the tested registration algorithms. We provide a set of ablation studies to further quantify the importance in the predefined template, adapted displacement regularisation and other training and network designs, in the context of replacing "traditional" registration algorithms in aligning the vertebral sub-regions.

In summary, we first present comparison experiments that show significantly improved registration accuracy, over the existing registration algorithms; second, we report ablation studies to demonstrate benefits from the proposed network design and training choices. We therefore conclude that the contribution of this work is the proposed new template-based GNN approach, adapted from those previously developed for segmentation, for inferring local correspondence on vertebral sub-regions.

2. Registration with a Reference Mesh

In this section, we provide the details of the proposed approach based on CNN and GNN, to register a reference mesh (i.e. the template in the context of segmentation used in previous studies), from a reference image, to the given voxel image.

The proposed registration method aims to register a set of predefined surface points in the reference image I^{ref} with those in the target image I^{tgt}. A smoothed surface mesh from the training data sets is used as reference mesh, which can be represented by sets of vertices, edges, and faces, i.e. $\mathcal{M}^{\text{ref}} = (\mathcal{V}^{\text{ref}}, \mathcal{E}^{\text{ref}}, \mathcal{F}^{\text{ref}})$. The registration task is to predict the

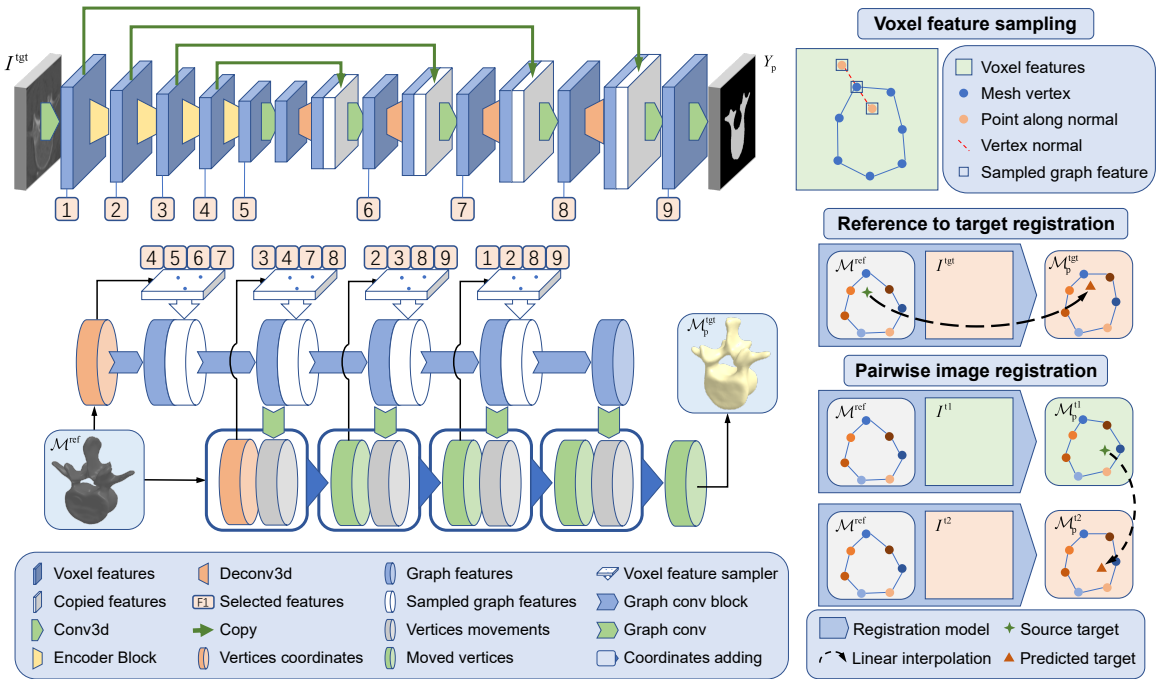


Figure 1: Overview of the proposed registration method, illustrating the GNN with a CNN-based image feature extractor (left) and the feature sampling (right-top) and reference-to-target (right-middle) and pairwise registration (right-bottom). Symbols are defined in the text.

displacements $\mathcal{D}_\mathrm{p} = f_\theta(I^\mathrm{tgt}, \mathcal{M}^\mathrm{ref})$ for reference mesh vertices from the image and reference mesh, where f_θ is a neural network with parameters θ. Applying the displacement to $\mathcal{M}^\mathrm{ref}$ results in a predicted mesh $\mathcal{M}_\mathrm{p}^\mathrm{tgt} = (\mathcal{V}_\mathrm{p}^\mathrm{tgt}, \mathcal{E}_\mathrm{p}^\mathrm{tgt}, \mathcal{F}_\mathrm{p}^\mathrm{tgt})$ for the I^tgt. That is, for any vertex $\boldsymbol{v}^\mathrm{ref} \in \mathcal{V}^\mathrm{ref}$, let the displacement be $\boldsymbol{d}_\mathrm{p}$, the predicted vertex $\boldsymbol{v}_\mathrm{p}^\mathrm{tgt}$ is calculated by $\boldsymbol{v}^\mathrm{ref} + \boldsymbol{d}_\mathrm{p}$. Therefore, a series of corresponding points $\mathcal{R}_\mathrm{p}^\mathrm{ref,tgt} = \{(\boldsymbol{v}^\mathrm{ref}, \boldsymbol{v}_\mathrm{p}^\mathrm{tgt}) \mid \boldsymbol{v}^\mathrm{ref} \in \mathcal{V}^\mathrm{ref}, \boldsymbol{v}_\mathrm{p}^\mathrm{tgt} \in \mathcal{V}_\mathrm{p}^\mathrm{tgt}\}$ from the reference image to the target image are generated through the proposed registration method. For a new target point $\boldsymbol{p}^\mathrm{ref}$ in I^ref, the predicted corresponding point in I^tgt for it can be obtained by using the piecewise linear interpolator $\boldsymbol{p}_\mathrm{p}^\mathrm{tgt} = \Phi(\mathcal{R}_\mathrm{p}^\mathrm{ref,tgt}, \boldsymbol{p}^\mathrm{ref})$, where $\Phi(\mathcal{R}^\mathrm{a,b}, \boldsymbol{p}^a)$ denotes the interpolated coordinate at $\boldsymbol{p}^a$ using a series of paired points in a and b. Figure 1 illustrates this reference to target registration process.

More generally, pairwise registration method registers the set of predefined surface points from one image I^t1 to a second image I^t2, illustrated in Figure 1. Denote the corresponding predicted meshes as $\mathcal{M}_\mathrm{p}^\mathrm{t1} = (\mathcal{V}_\mathrm{p}^\mathrm{t1}, \mathcal{E}_\mathrm{p}^\mathrm{t1}, \mathcal{F}_\mathrm{p}^\mathrm{t1})$ and $\mathcal{M}_\mathrm{p}^\mathrm{t2} = (\mathcal{V}_\mathrm{p}^\mathrm{t2}, \mathcal{E}_\mathrm{p}^\mathrm{t2}, \mathcal{F}_\mathrm{p}^\mathrm{t2})$ respectively. With the vertex displacement predictions from the input to the target, the proposed registration method can be applied by registering the reference mesh $\mathcal{M}^\mathrm{ref}$ to I^t1 and I^t2 separately, with displacements $\mathcal{D}^\mathrm{t1} = f_\theta(I^\mathrm{t1}, \mathcal{M}^\mathrm{ref})$ and $\mathcal{D}^\mathrm{t2} = f_\theta(I^\mathrm{t2}, \mathcal{M}^\mathrm{ref})$ respectively. For any vertex $\boldsymbol{v}^\mathrm{ref} \in \mathcal{V}^\mathrm{ref}$, let the predicted vertices in the two images be $\boldsymbol{v}^\mathrm{ref} + \boldsymbol{d}_\mathrm{p}^\mathrm{t1}$ and $\boldsymbol{v}^\mathrm{ref} + \boldsymbol{d}_\mathrm{p}^\mathrm{t2}$. The relative displacement between the two images for $\boldsymbol{v}^\mathrm{ref}$ is $\boldsymbol{d}_\mathrm{p}^\mathrm{t2} - \boldsymbol{d}_\mathrm{p}^\mathrm{t1}$ and the correspondence between the two predicted vertices is established as $\boldsymbol{v}_\mathrm{p}^\mathrm{t2} = \boldsymbol{v}_\mathrm{p}^\mathrm{t1} - \boldsymbol{d}_\mathrm{p}^\mathrm{t1} + \boldsymbol{d}_\mathrm{p}^\mathrm{t2}$. Therefore, in the pairwise registration task, for a point $\boldsymbol{p}^\mathrm{t1}$ from I^t1, the corresponding point in I^t2 can be predicted as $\boldsymbol{p}_\mathrm{p}^\mathrm{t2} = \Phi(\mathcal{R}_\mathrm{p}^\mathrm{t1,t2}, \boldsymbol{p}^\mathrm{t1})$, where $\mathcal{R}_\mathrm{p}^\mathrm{t1,t2} = \{(\boldsymbol{v}_\mathrm{p}^\mathrm{t1}, \boldsymbol{v}_\mathrm{p}^\mathrm{t2}) | \boldsymbol{v}_\mathrm{p}^\mathrm{t1} \in \mathcal{V}_\mathrm{p}^\mathrm{t1}, \boldsymbol{v}_\mathrm{p}^\mathrm{t2} \in \mathcal{V}_\mathrm{p}^\mathrm{t2}\}$.

3. Network Construction and the Training Loss

In this work, the neural network with both CNN and GNN modules from Bongratz et al. (2022) is adopted, illustrated in Figure 1. The U-Net-like CNN module ingests the input image and predicts a segmentation mask for vertebra. The GNN module takes the reference mesh as input and performs graph convolution with vertex features extracted from CNN module to adjust vertex coordinates progressively. Formally, at each graph convolution layer, denote the vertex $\boldsymbol{v}_i$'s features in GNN as $\boldsymbol{f}_{i,\mathrm{GNN}}$, it is updated by aggregating vertex features of neighbours and itself from both GNN and CNN modules:

$$\boldsymbol{f}_{i,\mathrm{GNN}} = h(W_0 \boldsymbol{f}_i + \boldsymbol{b}_0 + \sum_{j \in \mathcal{N}(i)} (W_1 \boldsymbol{f}_j + \boldsymbol{b}_1)) \tag{1}$$

$$\boldsymbol{f}_i = \mathrm{concat}[\boldsymbol{f}_{i,\mathrm{CNN}}, \hat{\boldsymbol{f}}_{i,\mathrm{GNN}}] \tag{2}$$

where h is Relu activation layer; $W_0, \boldsymbol{b}_0, W_1, \boldsymbol{b}_1$ are learnable weights; $\mathcal{N}(i)$ is the neighbour vertices of $\boldsymbol{v}_i$; and $\boldsymbol{f}_{i,\mathrm{CNN}}$ is the features extracted from CNN embeddings and $\hat{\boldsymbol{f}}_{i,\mathrm{GNN}}$ is previous graph features. Particularly, for vertex $\boldsymbol{v}_i$, denote its normal vector $\boldsymbol{n}_i$, which is the averaged normal vectors of the faces that $\boldsymbol{v}_i$ belongs to. $\boldsymbol{f}_{i,\mathrm{CNN}}$ is calculated by concatenating the sampled CNN embeddings H at multiple points along $\boldsymbol{n}_i$:

$$\boldsymbol{f}_{i,\mathrm{CNN}} = \mathrm{concat}_{\alpha_k \in \alpha}[\phi(H, \boldsymbol{v}_i + \alpha_k \boldsymbol{n}_i)], \tag{3}$$

with $\phi(H, \boldsymbol{v})$ representing the sampled embedding from CNN embedding H at point $\boldsymbol{v}$ and α is the predefined distances list. Such sampling is expected to provide more context beyond the surface and facilitate the model training, described as follows.

To train the network f_θ, a composed loss functions Bongratz et al. (2022) is adapted:

$$\begin{aligned} \mathcal{L} = {} & \lambda_{\mathrm{seg}}(t)\mathcal{L}_{\mathrm{seg}}(Y_{\mathrm{p}}, Y_{\mathrm{gt}}) + \lambda_{\mathrm{delay}}(t)(\lambda_{\mathrm{chamfer}}\mathcal{L}_{\mathrm{chamfer}}(\mathcal{M}_{\mathrm{p}}^{\mathrm{tgt}}, \mathcal{M}_{\mathrm{gt}}^{\mathrm{tgt}}) \\ & + \lambda_{\mathrm{norm,inter}}\mathcal{L}_{\mathrm{norm,inter}}(\mathcal{M}_{\mathrm{p}}^{\mathrm{tgt}}, \mathcal{M}_{\mathrm{gt}}^{\mathrm{tgt}}) + \lambda_{\mathrm{norm,intra}}\mathcal{L}_{\mathrm{norm,intra}}(\mathcal{M}_{\mathrm{p}}^{\mathrm{tgt}}) \\ & + \lambda_{\mathrm{edge}}(t)\mathcal{L}_{\mathrm{edge}}(\mathcal{M}_{\mathrm{p}}^{\mathrm{tgt}}) + \lambda_{\mathrm{disp}}\mathcal{L}_{\mathrm{disp}}(\mathcal{D}_{\mathrm{p}}, \mathcal{M}_{\mathrm{p}}^{\mathrm{tgt}})) \end{aligned} \tag{4}$$

where, $\mathcal{L}_{\mathrm{seg}}(Y_{\mathrm{p}}, Y_{\mathrm{gt}})$ is the binary cross entropy between the predicted and ground truth vertebral segmentation masks; $\mathcal{L}_{\mathrm{chamfer}}(\mathcal{M}_{\mathrm{p}}^{\mathrm{tgt}}, \mathcal{M}_{\mathrm{gt}}^{\mathrm{tgt}})$ is the curvature-weighted Chamfer loss that penalises mismatched vertex positions between the predicted and ground truth meshes; $\mathcal{L}_{\mathrm{norm,inter}}(\mathcal{M}_{\mathrm{p}}^{\mathrm{tgt}}, \mathcal{M}_{\mathrm{gt}}^{\mathrm{tgt}})$ is the normal distance loss that penalises mismatched vertex normal vectors between the predicted and ground truth meshes; $\mathcal{L}_{\mathrm{norm,intra}}(\mathcal{M}_{\mathrm{p}}^{\mathrm{tgt}})$ is the normal distance loss that promotes the consistency of adjacent face normal vectors in the predicted mesh; $\mathcal{L}_{\mathrm{edge}}(\mathcal{M}_{\mathrm{p}}^{\mathrm{tgt}})$ is the edge length loss that penalises long edges of the predicted mesh; and $\mathcal{L}_{\mathrm{disp}}(\mathcal{D}_{\mathrm{p}}, \mathcal{M}_{\mathrm{p}}^{\mathrm{tgt}})$ is the displacement regularisation loss which calculates the L2 norm of the predicted vertices displacements. Different from the Laplacian smoothing in Bongratz et al. (2022), we weighted the vertex displacement by the inverse of the edge length to account for differences in neighbors at different distance. The definitions of these loss functions are detailed in Appendix 6.1.

To avoid divergence at the initial training stage, frequently found in our preliminary experiments, a delayed weight strategy was adopted $\lambda_{\mathrm{delay}}(t) = 0.5 + \arctan[(t - 3000)/\pi]$, controlled by the number of steps t. A dynamic loss weighting mechanism is also empirically designed, $\lambda_{\mathrm{seg}}(t) = \lambda_{\mathrm{edge}}(t) = 0.5 - \arctan[(t - 10000)/1000\pi]$.

4. Experiments and Results

4.1. Data sets and preprocessing

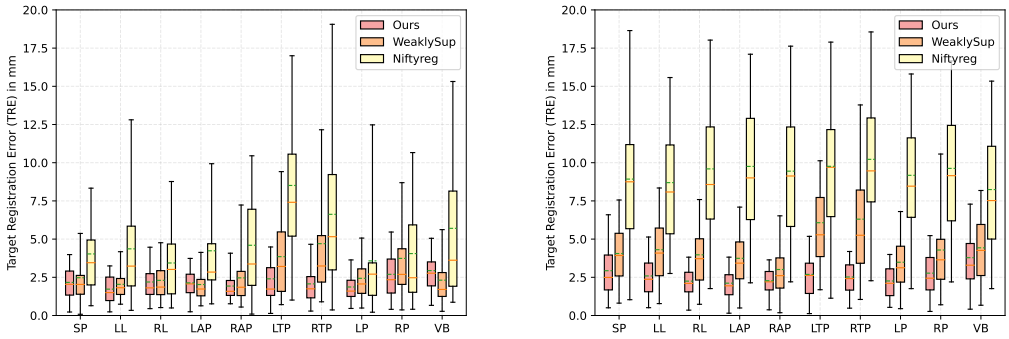
Three online published spine CT image segmentation data sets were used for training and testing, namely Lumbar vertebra segmentation CT image data sets (LumSeg), Spine and Vertebrae Segmentation Datasets (SpiSeg), and xVertSeg data sets (xVertSeg). The mixed data contain a total of 35 subjects with 175 lumbar vertebrae, and for each case, the original CT image and the voxel-labeled masks are given. The data were randomly divided into a training set of 24 subjects and a holdout test set of 11 subjects. All the results in the paper were based on the test set, without using a validation set for hyperparameter tuning, which may further improve the performance. All images and segmentation ground truth was resampled at a voxel dimension of $0.5mm \times 0.5mm \times 0.5mm$, which are randomly cropped, from the vertebral centre, to a size of $128 \times 192 \times 192$ in advance.

The ground truth surface meshes were obtained by using the marching cube algorithm (Lorensen and Cline, 1987) based on the segmentation labels, followed by a Laplacian smoothing filter configured at 20 iterations. For the purpose of validating sub-region registration, 9 planes manually were selected for each mesh based on which vertices were divided into 10 categories, representing spinous process (SP), left lamina (LL), right lamina (RL), left articular process (LAP), right articular process (RAP), left transverse process (LTP), right transverse process (RTP), left pedicle (LP), right pedicle (RP) and vertebral body (VB). The center of each category was marked as the target point for registration experiments.

An image from the training data sets was randomly selected as the predefined reference and the surface mesh was extracted with the marching cubes algorithm. The Laplacian smoothing algorithm was applied for 30 times on the reference mesh to remove the unsmooth and personalized details. The experimental results were compared with two image registration baselines, the iterative intensity-based method NiftyReg (Modat et al., 2010) and the learning-based method WeaklySup (Hu et al., 2018a,b; Fu et al., 2020). The former was based on SSD as similarity measure regualrised by bending energy, with otherwise default configurations; while the latter was trained in a weakly-supervised algorithm (Hu et al., 2018b), with the same supervising labels, training set, and reference case (for comparing with the reference-to-target registration). Both results are reported on the same test set. The official implementation of NiftyReg and a PyTorch adaptation of WeaklySup (Yang et al., 2022) were used. All experiments were performed on Nvidia GPU Quadro P5000.

4.2. Reference to target registration

The alignment between the fixed reference image and images in test set was first quantified and the TREs based on the geometric centres of individual sub-regions are summarised in Figure 2(a) and also in Table 3. Compared with the results from NiftyReg, the TREs are statistically lower (all p-values$<$0.001, paired t-tests at a significance level α=0.05), for all sub-regions. The improvement over the learning-based algorithms was less evident, with lower TREs observed in eight out of ten sub-regions and six of them yielded statistical significance, indicating a comparable registration performance in this task.



(a) Reference to target registration. (b) Arbitrary image pair registration.

Figure 2: The TREs results with median and mean in orange and green, respectively.

4.3. Arbitrary image pair registration via reference

In this experiment, 100 vertebral pairs were randomly sampled from the test set, and the sub-region TREs are illustrated in Figure 2(b). Our registration model achieves an average TRE on all sub-regions of 2.62±1.42 mm, significantly outperformed over both WeaklySup (4.37±2.67 mm) and NiftyReg (9.35±4.38 mm), with all p-values$<$0.023 and all p-values$<$0.001, respectively.

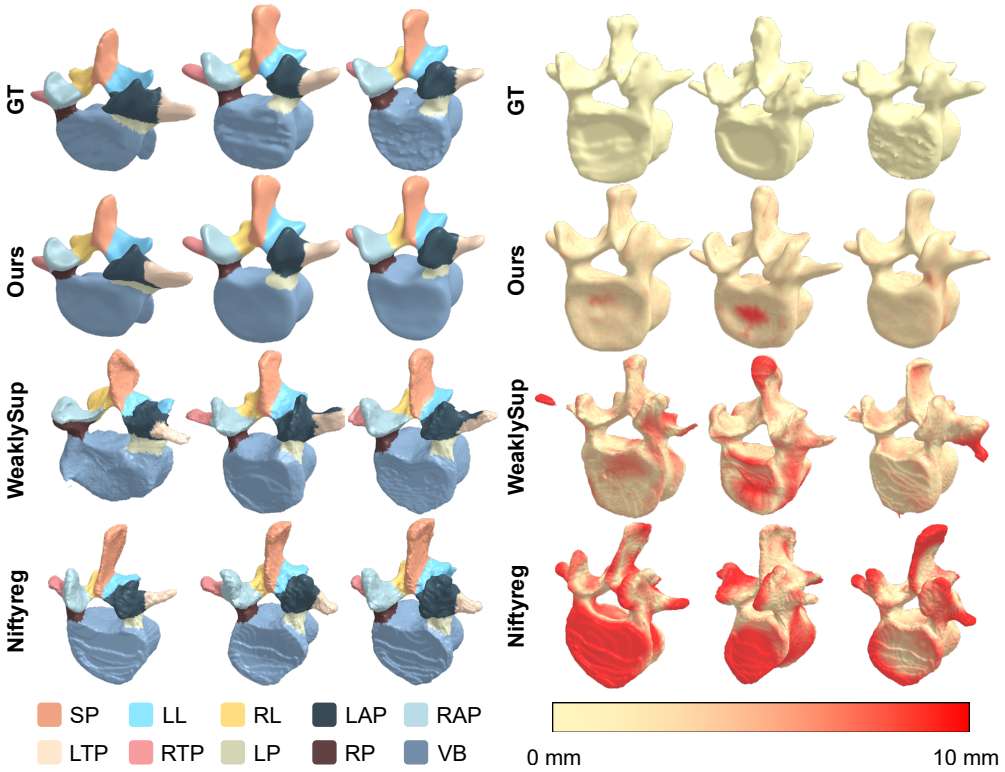
The choice of the predefined reference may be a source of bias, however, previous studies showed that segmentation tasks did not seem sensitive to such a smoothed mesh template (Bongratz et al., 2022). It was found to be a much stronger bias if the two test images are used as respective reference and target during test time - without using the intermediate predefined reference, as opposed to the two correspondence composing approach described above. This indeed led to much higher TREs, reported in an ablation study in Sec. 4.5.

4.4. Vertebra and sub-region segmentation

For reference purposes, we also report the results based on segmentation metrics on both the sub-regions and the entire vertebra. Hausdorff distance (HD), the average symmetric surface distance (ASSD), and Dice score are summarised in Table 1 and Figure 3(a). More examples are provided in Appendix 6.2 Figure 4. Interestingly, our model achieves significantly better results than the baselines (p-value$<$0.001). Some segmentation examples can be seen in Figure 3(b) and Table 2. An interesting observation, consistent with the conclusion from the ablation studies presented in Sec. 4.5, is that the performance advantage over the alternative registration shared across segmentation of vertebrae, segmentation of sub-regions and alignment of the two, which may suggest the designed optimisation, especially those that are not defined on the surface distance, could benefit both segmentation and registration tasks.

4.5. Ablation studies

To better understand 1) the importance in network architecture and loss function design and 2) their respective contributions to both segmentation and registration tasks, we provide a set



(*a*) Segmented sub-regions. (*b*) Vertebrae segmentation errors.

Figure 3: Examples of sub-regions (a) and vertebra segmentation errors with a colour error bar (b). Further examples are provided in Appendix 6.2 Figure 4.

Table 1: Sub-regions segmentation results in mm with average±standard deviation, details also described in text. Statistically significant results are in bold (paired t-tests at a significance level α=0.05).

	SP	LL	RL	LAP	RAP
Ours	**0.97/0.39**	**0.78/0.31**	**0.84/0.32**	**1.06/0.43**	**1.44/0.54**
WeaklySup	2.46/0.96	2.73/1.09	2.58/1.07	4.04/2.20	3.65/1.65
NiftyReg	4.98/1.98	5.59/2.02	5.67/2.27	6.78/4.05	8.46/5.02
	LTP	RTP	LP	RP	VB
Ours	**1.37/0.41**	**0.95/0.35**	**1.62/0.57**	**1.52/0.49**	**0.96/0.45**
WeaklySup	3.12/1.15	2.54/1.04	2.77/1.25	2.57/1.07	3.32/1.22
NiftyReg	5.11/2.09	4.68/1.97	3.69/1.63	3.81/1.68	8.88/3.59

Table 2: Vertebra segmentation results.

Models	**HD (mm)**	**ASSD (mm)**	**Dice (%)**
Ours	**1.14**±0.49	**0.54**± 0.11	**93.25**± 1.47
WeaklySup	3.22±1.76	1.19± 0.46	86.50± 5.21
NiftyReg	7.97± 4.65	2.52± 1.48	71.82±15.50

of ablation studies to compare the results when the following modification were independently made, summarised in Table 3. **Variable ref.**: This model was trained with a variable reference randomly from the training set, rather than a fixed reference and it was tested without using a fixed reference in pairwise registration experiments. **w/o ref.**: The trained model was tested in the arbitrary image pair registration task using the input pair images as respective reference and target directly instead of using the intermediate predefined reference. **w/o norm. feat.**: Graph features were only interpolated from the voxel features by the mesh vertices which means $\alpha = [0]$ in Equation 3. **Constant $\lambda_{\mathbf{vox}}$**: The weight for voxel segmentation loss was set to a constant value during the training. **Classical chamfer**: Classical chamfer loss was used which is equal to set $\kappa(\cdot \mid \kappa_{\max}) = 1$ in Equation (6). **Laplacian**: Uniform weights were used when calculating the displacement regularisation loss which means $w(\boldsymbol{v}, \boldsymbol{v}_{\text{nbr}}) = \frac{1}{\mathcal{N}(\boldsymbol{v})}$ in Equation (12) and it is equal to using the Laplacian smooth on the predicted displacements (Nealen et al., 2006). **w/o disp. reg.**: The model was trained without displacement regularisation loss. **Constant $\lambda_{\mathbf{edge}}$**: A constant weight for edge length loss was used when training.

Table 3: Ablation study results. TRE-reference and TRE-pair denote the performance from reference-to-target registration and pairwise registration experiments, described in Secs. 4.2 and 4.3, respectively. Other metrics are described in text.

	TRE-reference	TRE-pair	HD	ASSD	Dice
Ours	**2.15±1.33**	2.62±1.42	1.14±0.49	0.54±0.11	93.25±1.47
Variable ref.	5.48±2.81	5.48±5.39	3.16±0.84	0.98±0.20	87.22±3.75
w/o ref.	-	9.44±15.36	-	-	-
w/o norm. feat.	2.42±1.51	3.09±1.60	1.40±0.63	0.61±0.15	93.10±2.25
Constant λ_{vox}	2.80±1.62	3.04±1.52	1.84±1.05	0.74±0.27	92.71±2.94
Classical chamfer	3.06±1.64	3.45±1.71	1.96±0.85	0.75±0.20	91.82±2.16
Laplacian	2.28±1.39	2.78±1.51	1.10±0.47	0.53±0.11	92.71±2.68
w/o disp. reg.	3.53±1.68	4.06±2.02	1.52±0.62	0.65±0.15	93.19±2.22
Constant λ_{edge}	3.03±1.49	3.22±1.58	1.41±0.83	0.61±0.20	**93.81±2.19**

5. Disccusion

The proposed network uses a GNN-represented mesh to describe the bone surface and predicts the vertex displacements between a reference image and one or more target images, to establish inter-vertebral correspondence between sub-regions for planning orthopaedic surgery. Interestingly, the proposed method, adapted from earlier segmentation networks, outperformed all the direct registration algorithms without an intermediate reference, including those using the same CNN-GNN networks (with variable reference), previously proposed weak supervision and a non-learning registration baseline. This may suggest the potentials in well-developed GNN-based segmentation methods for establishing dense correspondence between segmented ROIs in wider applications and its competitiveness with existing registration algorithms.

References

Fabian Bongratz, Anne-Marie Rickmann, Sebastian Pölsterl, and Christian Wachinger. Vox2cortex: Fast explicit reconstruction of cortical surfaces from 3d mri scans with geometric deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20773–20783, 2022.

David Cohen-Steiner and Jean-Marie Morvan. Restricted delaunay triangulations and normal cycle. In *Proceedings of the nineteenth annual symposium on Computational geometry*, pages 312–321, 2003.

Neal P Dillon, Michael A Siebold, Jason E Mitchell, Gregoire S Blachon, Ramya Balachandran, J Michael Fitzpatrick, and Robert J Webster III. Increasing safety of a robotic system for inner ear surgery using probabilistic error modeling near vital anatomy. In *Medical Imaging 2016: Image-Guided Procedures, Robotic Interventions, and Modeling*, volume 9786, pages 438–452. SPIE, 2016.

Kexue Fu, Shaolei Liu, Xiaoyuan Luo, and Manning Wang. Robust point cloud registration framework based on deep graph matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8893–8902, June 2021.

Yunguan Fu, Nina Montaña Brown, Shaheer U. Saeed, Adrià Casamitjana, Zachary M. C. Baum, Rémi Delaunay, Qianye Yang, Alexander Grimwood, Zhe Min, Stefano B. Blumberg, Juan Eugenio Iglesias, Dean C. Barratt, Ester Bonmati, Daniel C. Alexander, Matthew J. Clarkson, Tom Vercauteren, and Yipeng Hu. Deepreg: a deep learning toolkit for medical image registration. *Journal of Open Source Software*, 5(55):2705, 2020. doi: 10.21105/joss.02705. URL https://doi.org/10.21105/joss.02705.

Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. Vision gnn: An image is worth graph of nodes. *arXiv preprint arXiv:2206.00272*, 2022.

Ying Hu, Haiyang Jin, Liwei Zhang, Peng Zhang, and Jianwei Zhang. State recognition of pedicle drilling with force sensing in a robotic spinal surgical system. *IEEE/ASME Transactions On Mechatronics*, 19(1):357–365, 2013.

Yipeng Hu, Marc Modat, Eli Gibson, Nooshin Ghavami, Ester Bonmati, Caroline M Moore, Mark Emberton, J Alison Noble, Dean C Barratt, and Tom Vercauteren. Label-driven weakly-supervised learning for multimodal deformable image registration. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 1070–1074. IEEE, 2018a.

Yipeng Hu, Marc Modat, Eli Gibson, Wenqi Li, Nooshin Ghavami, Ester Bonmati, Guotai Wang, Steven Bandula, Caroline M Moore, Mark Emberton, et al. Weakly-supervised convolutional neural networks for multimodal image registration. *Medical image analysis*, 49:1–13, 2018b.

Fanwei Kong and Shawn C. Shadden. Whole heart mesh generation for image-based computational simulations by learning free-from deformations, 2021. URL https://arxiv.org/abs/2107.10839.

Fanwei Kong, Nathan Wilson, and Shawn Shadden. A deep-learning approach for direct whole-heart mesh reconstruction. *Medical image analysis*, 74:102222, 2021.

Nikolas Lessmann, Bram Van Ginneken, Pim A De Jong, and Ivana Išgum. Iterative fully convolutional neural networks for automatic vertebra segmentation and identification. *Medical image analysis*, 53:142–155, 2019.

Qian Li, Zhijiang Du, and Hongjian Yu. Precise laminae segmentation based on neural network for robot-assisted decompressive laminectomy. *Computer Methods and Programs in Biomedicine*, 209:106333, 2021.

William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987.

Marc Modat, Gerard R Ridgway, Zeike A Taylor, Manja Lehmann, Josephine Barnes, David J Hawkes, Nick C Fox, and Sébastien Ourselin. Fast free-form deformation using graphics processing units. *Computer methods and programs in biomedicine*, 98(3):278–284, 2010.

Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8280–8290, 2022.

Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Laplacian mesh optimization. In *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 381–389, 2006.

Daniel Rueckert, Luke I Sonoda, Carmel Hayes, Derek LG Hill, Martin O Leach, and David J Hawkes. Nonrigid registration using free-form deformations: application to breast mr images. *IEEE transactions on medical imaging*, 18(8):712–721, 1999.

Udaranga Wickramasinghe, Edoardo Remelli, Graham Knott, and Pascal Fua. Voxel2mesh: 3d mesh model generation from volumetric data. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 299–308. Springer, 2020.

Markus Worchel, Rodrigo Diaz, Weiwen Hu, Oliver Schreer, Ingo Feldmann, and Peter Eisert. Multi-view mesh reconstruction with neural deferred shading. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6187–6197, 2022.

Qianye Yang, David Atkinson, Yunguan Fu, Tom Syer, Wen Yan, Shonit Punwani, Matthew J Clarkson, Dean C Barratt, Tom Vercauteren, and Yipeng Hu. Cross-modality image registration using a training-time privileged third modality. *IEEE Transactions on Medical Imaging*, 41(11):3421–3431, 2022.

6. Appendix

6.1. Loss Functions

6.1.1. Segmentation Loss

The binary cross entropy between the predicted and ground truth vertebral segmentation masks $\mathcal{L}_{\text{seg}}(Y_{\text{p}}, Y_{\text{tgt}})$ is defined as:

$$\mathcal{L}_{\text{seg}}(Y_{\text{p}}, Y_{\text{tgt}}) = -\frac{1}{N}\sum_{i=1}^{N}(Y_{\text{gt},i}\log Y_{\text{p},i} + (1 - Y_{\text{gt},i})\log(1 - Y_{\text{p},i})) \tag{5}$$

where Y_{gt} is the binary mask and Y_{p} is the predicted soft mask with values between $[0, 1]$. The subscript i represents the value at voxel i which iterates over all voxels in the image.

6.1.2. Curvature-weighted Chamfer Loss

The curvature-weighted Chamfer loss that penalises mismatched vertex positions between the predicted and ground truth meshes is defined as $\mathcal{L}_{\text{chamfer}}(\mathcal{M}_{\text{p}}^{\text{tgt}}, \mathcal{M}_{\text{gt}}^{\text{tgt}})$:

$$\begin{aligned}\mathcal{L}_{\text{chamfer}}(\mathcal{M}_{\text{p}}^{\text{tgt}}, \mathcal{M}_{\text{gt}}^{\text{tgt}}) = &\frac{1}{\left|\mathcal{V}_{\text{gt}}^{\text{tgt}}\right|}\sum_{\boldsymbol{u}\in\mathcal{V}_{\text{gt}}^{\text{tgt}}}\kappa(\boldsymbol{u} \mid \kappa_{\max})\min_{\mathbf{v}\in\mathcal{V}_{\text{p}}^{\text{tgt}}}\|\mathbf{u}-\mathbf{v}\|^2 \\ &+ \frac{1}{\left|\mathcal{V}_{\text{p}}^{\text{tgt}}\right|}\sum_{\boldsymbol{v}\in\mathcal{V}_{\text{p}}^{\text{tgt}}}\kappa(\tilde{\boldsymbol{u}} \mid \kappa_{\max})\min_{\mathbf{u}\in\mathcal{V}_{\text{gt}}^{\text{tgt}}}\|\mathbf{v}-\mathbf{u}\|^2,\end{aligned} \tag{6}$$

with $\tilde{\boldsymbol{u}}$ being the closest vertex in $\mathcal{V}_{\text{gt}}^{\text{tgt}}$ to $\boldsymbol{v}$, i.e. $\tilde{\boldsymbol{u}} = \arg\min_{\mathbf{u}\in\mathcal{V}_{\text{gt}}^{\text{tgt}}}\|\mathbf{v}-\mathbf{u}\|^2$. $\kappa(\cdot \mid \kappa_{\max})$ is the curvature function defined in Bongratz et al. (2022) with discrete mean curvature function $\bar{\kappa}(\cdot)$ defined in Cohen-Steiner and Morvan (2003):

$$\kappa(\boldsymbol{v} \mid \kappa_{\max}) = \min(1 + \bar{\kappa}(\boldsymbol{v}), \kappa_{\max}). \tag{7}$$

6.1.3. Inter-mesh Normal Distance Loss

The normal distance loss that penalises mismatched vertex normal vectors between the predicted and ground truth meshes is defined as $\mathcal{L}_{\text{norm,inter}}(\mathcal{M}_{\text{p}}^{\text{tgt}}, \mathcal{M}_{\text{gt}}^{\text{tgt}})$:

$$\begin{aligned}\mathcal{L}_{\text{norm,inter}}(\mathcal{M}_{\text{p}}^{\text{tgt}}, \mathcal{M}_{\text{gt}}^{\text{tgt}}) = &\frac{1}{\left|\mathcal{V}_{\text{gt}}^{\text{tgt}}\right|}\sum_{\boldsymbol{u}\in\mathcal{V}_{\text{gt}}^{\text{tgt}}}1 - \cos(\boldsymbol{n}(\boldsymbol{u}), \boldsymbol{n}(\tilde{\boldsymbol{v}})) \\ &+ \frac{1}{\left|\mathcal{V}_{\text{p}}^{\text{tgt}}\right|}\sum_{\boldsymbol{v}\in\mathcal{V}_{\text{p}}^{\text{tgt}}}1 - \cos(\boldsymbol{n}(\boldsymbol{v}), \boldsymbol{n}(\tilde{\boldsymbol{u}})),\end{aligned} \tag{8}$$

with $\tilde{\boldsymbol{v}}$ being the closest vertex in $\mathcal{V}_{\text{p}}^{\text{tgt}}$ to $\boldsymbol{u}$ and $\tilde{\boldsymbol{u}}$ being the closest vertex in $\mathcal{V}_{\text{gt}}^{\text{tgt}}$ to $\boldsymbol{v}$, i.e. $\tilde{\boldsymbol{v}} = \arg\min_{\mathbf{v}\in\mathcal{V}_{\text{p}}^{\text{tgt}}}\|\mathbf{u}-\mathbf{v}\|^2$ and $\tilde{\boldsymbol{u}} = \arg\min_{\mathbf{u}\in\mathcal{V}_{\text{gt}}^{\text{tgt}}}\|\mathbf{v}-\mathbf{u}\|^2$. $\boldsymbol{n}(\boldsymbol{v})$ is the normal vector for vertex $\boldsymbol{v}$, which is the averaged normal vectors of the faces that $\boldsymbol{v}$ belongs to.

6.1.4. Intra-mesh Normal Distance Loss

The normal distance loss that evaluates the consistency of adjacent face normal vectors in the predicted mesh is defined as $\mathcal{L}_{\text{norm,intra}}(\mathcal{M}_{\text{p}}^{\text{tgt}})$:

$$\mathcal{L}_{\text{norm,intra}}(\mathcal{M}_{\text{p}}^{\text{tgt}}) = \frac{1}{\left|\mathcal{E}_{\text{p}}^{\text{tgt}}\right|} \sum_{\substack{f_1 \cap f_2 = e \\ e \in \mathcal{E}_{\text{p}}^{\text{tgt}}}} 1 - \cos(\boldsymbol{n}(f_1), \boldsymbol{n}(f_2)), \tag{9}$$

where faces f_1, f_2 shares the edge e and have the face normal vectors $\boldsymbol{n}(f_1), \boldsymbol{n}(f_2)$ respectively.

6.1.5. Edge Length Loss

$\mathcal{L}_{\text{edge}}(\mathcal{M}_{\text{p}}^{\text{tgt}})$ calculates the average of the edge lengths in the predicted mesh:

$$\mathcal{L}_{\text{edge}}(\mathcal{M}_{\text{p}}^{\text{tgt}}) = \frac{1}{\left|\mathcal{E}_{\text{p}}^{\text{tgt}}\right|} \sum_{(\boldsymbol{v}_1, \boldsymbol{v}_2) = e \in \mathcal{E}_{\text{p}}^{\text{tgt}}} \|\boldsymbol{v}_1 - \boldsymbol{v}_2\|^2 \tag{10}$$

6.1.6. Displacement Regularisation Loss

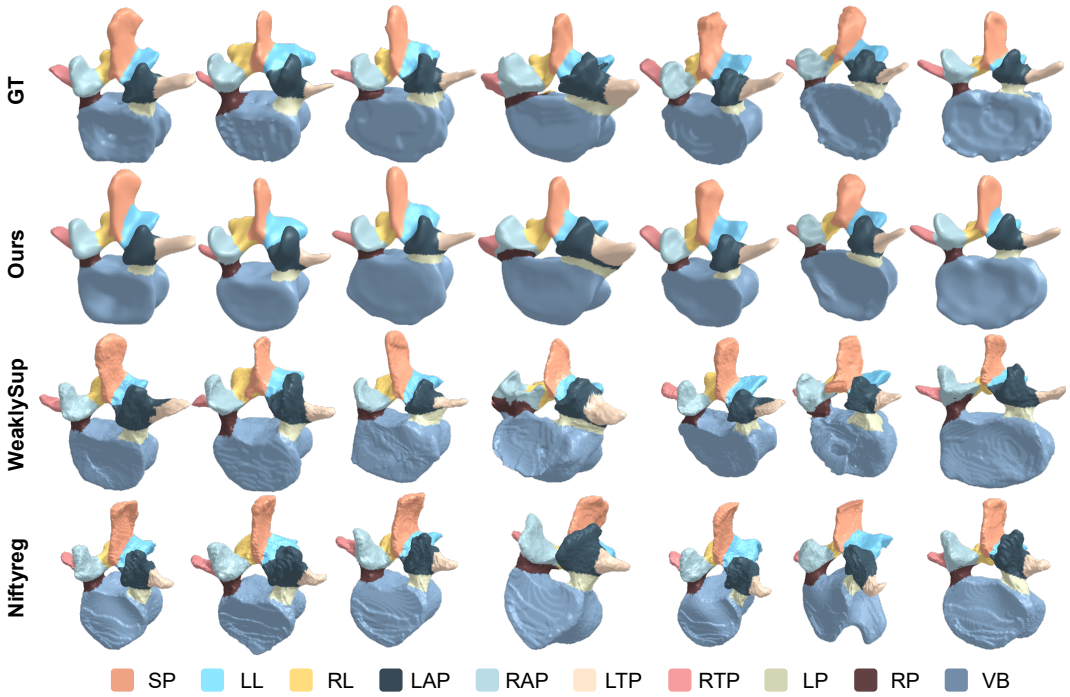
$\mathcal{L}_{\text{disp}}(\mathcal{D}_{\text{p}}, \mathcal{M}_{\text{p}}^{\text{tgt}})$ is defined as the sum of the derivative of displacements to constrain the predicted transformation to be smooth (Rueckert et al., 1999):

$$\mathcal{L}_{\text{disp}}(\mathcal{D}_{\text{p}}, \mathcal{M}_{\text{p}}^{\text{tgt}}) = \frac{1}{\left|\mathcal{V}_{\text{p}}^{\text{tgt}}\right|} \sum_{\boldsymbol{v} \in \mathcal{V}_{\text{p}}^{\text{tgt}}} \|\boldsymbol{d}_{\text{p}}(\boldsymbol{v}) - \sum_{\boldsymbol{v}_{\text{nbr}} \in \mathcal{N}(\boldsymbol{v})} w(\boldsymbol{v}, \boldsymbol{v}_{\text{nbr}}) \boldsymbol{d}_{\text{p}}(\boldsymbol{v}_{\text{nbr}})\|^2, \tag{11}$$

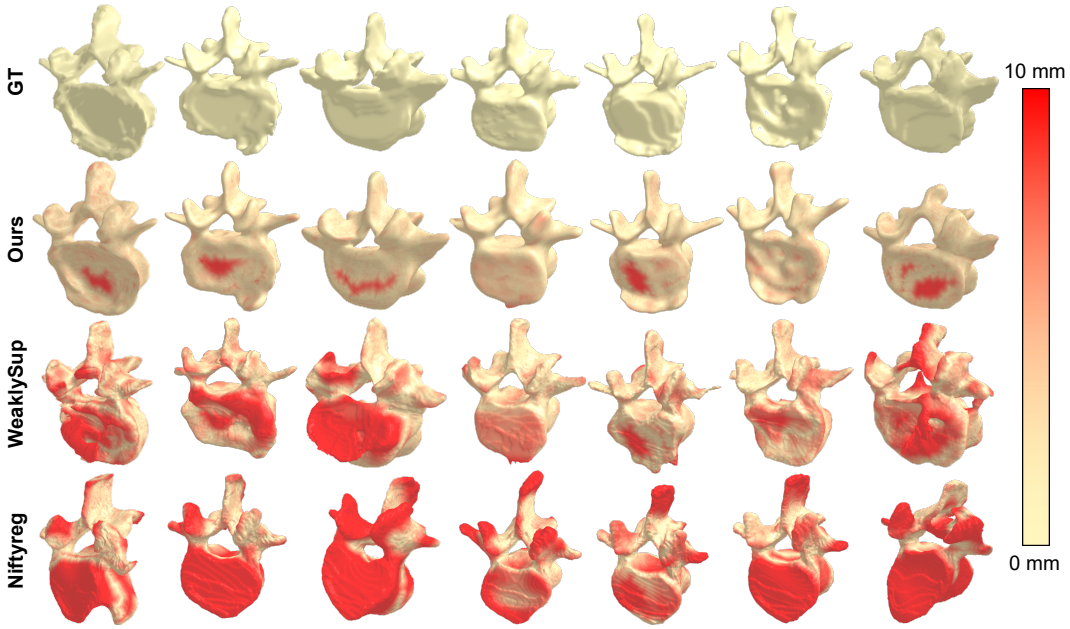
$$w(\boldsymbol{v}, \boldsymbol{v}_{\text{nbr}}) = \frac{1}{\|\boldsymbol{v} - \boldsymbol{v}_{\text{nbr}}\|} \Big(\sum_{\boldsymbol{v}' \in \mathcal{N}(\boldsymbol{v})} \frac{1}{\|\boldsymbol{v} - \boldsymbol{v}'\|} \Big)^{-1}. \tag{12}$$

For each vertex $\boldsymbol{v}$, the difference between the displacement $\boldsymbol{d}_{\text{p}}(\boldsymbol{v})$ and the averaged displacements of its neighbours which are weighted by the inverse of the edge length is computed as the displacement derivative.

6.2. Additional experimental results



(a) Examples of sub-regions segmentation results.



(b) Example of vertebrae segmentation. The surface distance error is illustrated with red colour.

Figure 4: Examples of vertebra and sub-region segmentation results.