

Appendix

ImplicitAtlas: Learning Deformable Shape Templates in Medical Imaging

Jiancheng Yang^{1,2} Udaranga Wickramasinghe² Bingbing Ni^{1*} Pascal Fua²
¹Shanghai Jiao Tong University, Shanghai, China ²EPFL, Lausanne, Switzerland

A. Implementation Details

A.1. Details of Network Architectures

In the proposed *ImplicitAtlas*, the network architecture of the Implicit Template Network \mathcal{T} and of the Implicit Deformation Network \mathcal{D} are similar except for the output channel in the final layer. Given a latent vector $\mathbf{t} \in \mathbb{R}^c$ or $\mathbf{h} \in \mathbb{R}^c$ ($c = 256$), it is first transformed and reshaped into a feature map $32 \times 4 \times 4 \times 4$ by stacks of linear layer, group normalization [4] (group=8), leaky ReLU activation [5] and pixel shuffle [3]. This feature map is then transformed into $48 \times 8 \times 8 \times 8$, $32 \times 16 \times 16 \times 16$ and $16 \times 32 \times 32 \times 32$ by stacks of trilinear upsampling layer and convolution-normalization-activation blocks.

A coordinate \mathbf{p} obtains its multi-scale feature by interpolation from the feature maps. The interpolation can be efficiently implemented in modern deep learning frameworks, for example, using “grid_sample” in PyTorch. The multi-scale feature, together with the coordinate itself, is then transformed into the output by a light MLP.

A.2. Training Details

During training, we use a combination of task loss and regularization loss. For the “*ImplicitAtlas* + reg.” model, the total loss is

$$\mathcal{L} = \mathcal{L}_{Task} + \lambda_1 \mathcal{L}_2 + \lambda_2 \mathcal{L}_{LS} + \lambda_3 \mathcal{L}_{DP}, \quad (1)$$

where we simply set $\lambda_1 = 10^{-3}$, $\lambda_2 = 10^{-3}$ and $\lambda_3 = 10^{-2}$ for all cases in this study, but it could be improved by a grid search. For the “*ImplicitAtlas*” variant without regularization loss, we set $\lambda_2 = 0$ and $\lambda_3 = 0$.

The model is trained under a down-sampled meshgrid of lower resolution (32^3) instead of full resolution (128^3). We add a random Gaussian noise ($\mu = 0$, $\sigma = 0.01$) to the training meshgrid, and the label is sampled from full-resolution ground truth. The model is trained using Adam [1] optimizer for 1,500 epochs. The initial learning rate is 3×10^{-4} , and it is multiplied by 1/10 after 800 and 1,200 epochs.

A.3. Reconstruction Details

The reconstruction procedure for unknown shapes is similar to DeepSDF [2]. Particularly, we fix the model weight of the trained *ImplicitAtlas* model and randomly initialize the latent code \mathbf{h} , and optimize the reconstruction loss. We use the basic formulation in DeepSDF [2] for a fair comparison,

$$\mathcal{L} = \mathcal{L}_{Task} + \lambda_1 \mathcal{L}_2. \quad (2)$$

During the reconstruction, as we want to speed up the procedure, we use a 64^3 meshgrid for all cases. The model is trained using an Adam [1] optimizer for 1,000 epochs. The initial learning rate is 1×10^{-1} , and it is multiplied by 1/2 after 20, 100, 200, 400 and 800 epochs.

As for shape completion, we apply an exactly same setting as the reconstruction for unknown shapes. The only difference is the loss in Eq. 2 is calculated on the points rather than the meshgrid.

B. More Visualization

We provide more visualization for a better understanding of the datasets and the model. In Fig. A1, 20 samples per dataset of liver, pancreas and hippocampus are shown. As is common in medical shape datasets, many samples are of low quality. In Fig. A2, we show 20 generated samples per dataset of liver, pancreas and hippocampus. They are diverse, of high quality, and much smoother than the training samples, though there are still artefacts in a few samples.

Moreover, we provide animations and interactive 3D models in “supplementary_visualization.zip”. To illustrate the difference between the learned templates, we provide animations of the 5 templates from the same view in “liver_template.gif”, “hippocampus_template.gif” and “pancreas_template.gif”. Here, the hippocampus is shown in a different view from the standard one for better visualization. As demonstrated, although the templates look alike, they are indeed different in details, which is proven important for the shape representation capacity. Besides, as all the visualization is provided in 2D, we further provide interactive 3D models (empowered by plotly.js)

*Corresponding author.

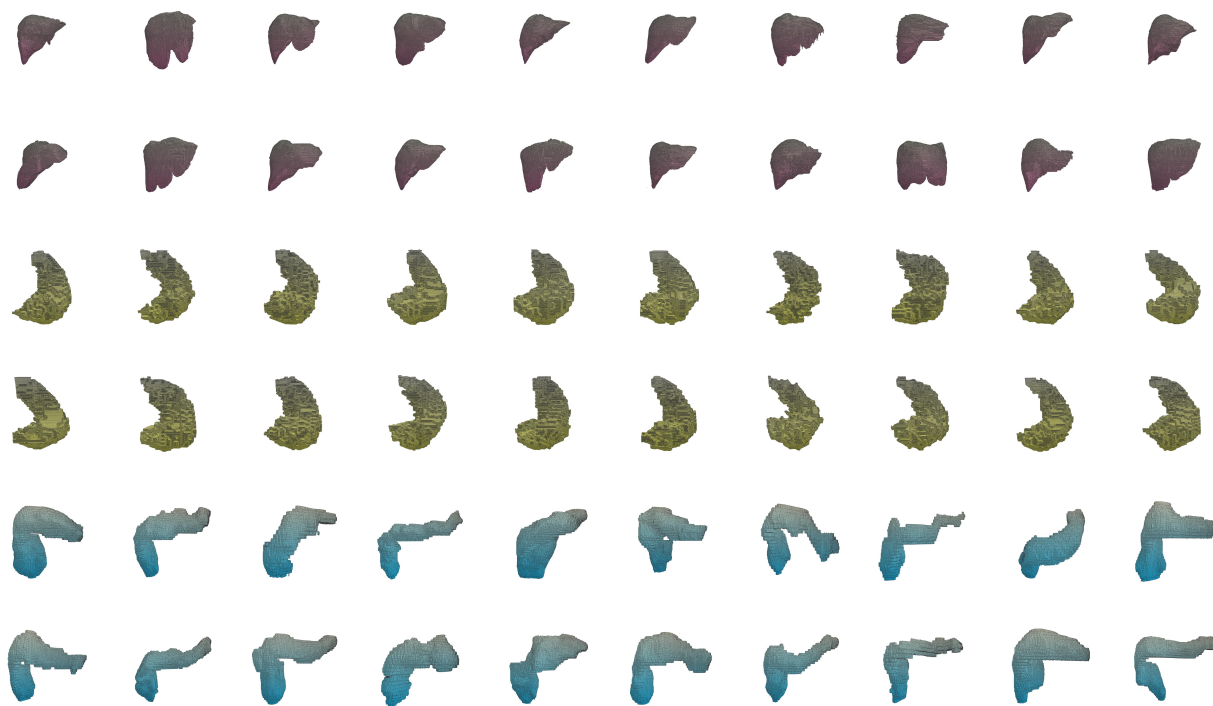


Figure A1. **Visualization of Dataset Samples.** Liver/hippocampus/pancreas are depicted in top/middle/bottom 2 rows.

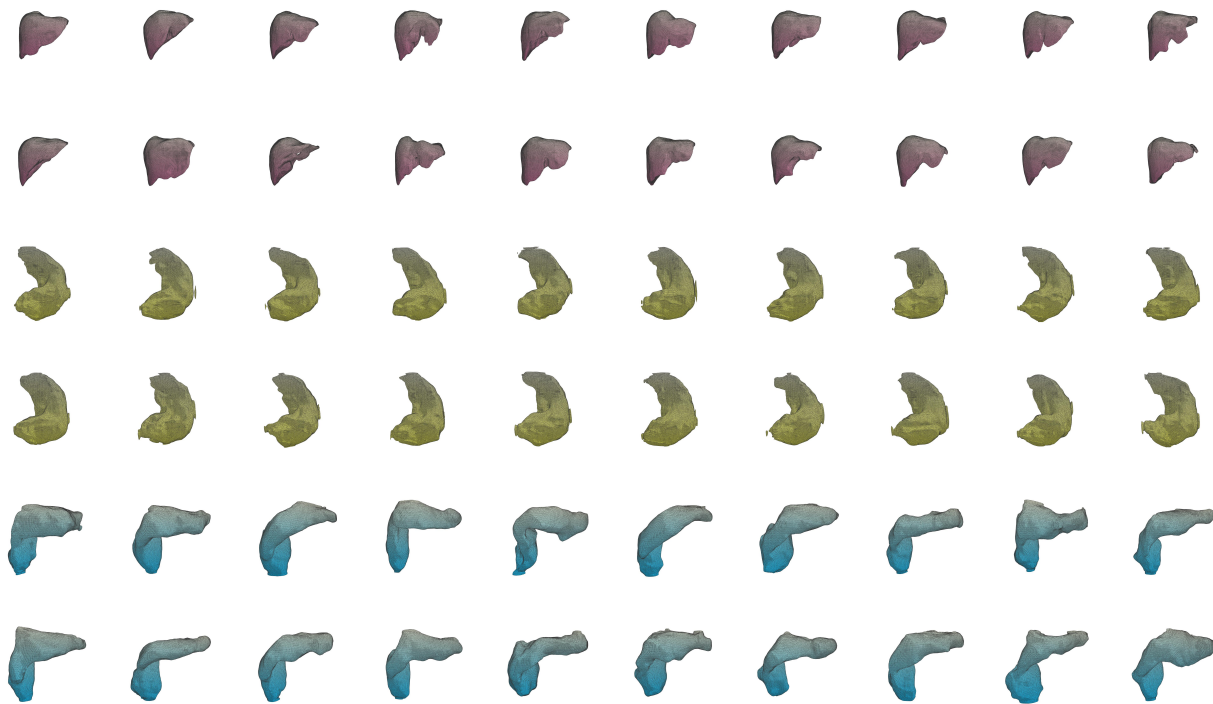


Figure A2. **Visualization of Unconditional Generation Samples.** Liver/hippocampus/pancreas are depicted in top/middle/bottom 2 rows. Note that there is no positional correspondence between this figure and Fig. [A2](#).

in “liver_generation.html”, “hippocampus_generation.html” and “pancreas_generation.html”. These 3 samples are generated as described in Sec. 4.4.

References

- [1] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv Preprint*, 2014. [1](#)
- [2] J. J. Park, P. Florence, J. Straub, R. A. Newcombe, and S. Lovegrove. DeepSdf: Learning Continuous Signed Distance Functions for Shape Representation. In *Conference on Computer Vision and Pattern Recognition*, 2019. [1](#)
- [3] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Conference on Computer Vision and Pattern Recognition*, pages 1874–1883, 2016. [1](#)
- [4] Yuxin Wu and Kaiming He. Group normalization. In *European Conference on Computer Vision*, pages 3–19, 2018. [1](#)
- [5] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv Preprint*, 2015. [1](#)