



# High-Resolution Cranial Defect Reconstruction by Iterative, Low-Resolution, Point Cloud Completion Transformers

Marek Wodzinski<sup>1,2</sup>(✉) , Mateusz Daniol<sup>1</sup> , Daria Hemmerling<sup>1</sup> ,  
and Mirosław Socha<sup>1</sup>

<sup>1</sup> Department of Measurement and Electronics, AGH University of Science and Technology, Krakow, Poland  
wodzinski@agh.edu.pl

<sup>2</sup> Information Systems Institute, University of Applied Sciences Western Switzerland (HES-SO Valais), Sierre, Switzerland

**Abstract.** Each year thousands of people suffer from various types of cranial injuries and require personalized implants whose manual design is expensive and time-consuming. Therefore, an automatic, dedicated system to increase the availability of personalized cranial reconstruction is highly desirable. The problem of the automatic cranial defect reconstruction can be formulated as the shape completion task and solved using dedicated deep networks. Currently, the most common approach is to use the volumetric representation and apply deep networks dedicated to image segmentation. However, this approach has several limitations and does not scale well into high-resolution volumes, nor takes into account the data sparsity. In our work, we reformulate the problem into a point cloud completion task. We propose an iterative, transformer-based method to reconstruct the cranial defect at any resolution while also being fast and resource-efficient during training and inference. We compare the proposed methods to the state-of-the-art volumetric approaches and show superior performance in terms of GPU memory consumption while maintaining high-quality of the reconstructed defects.

**Keywords:** Cranial Implant Design · Deep Learning · Shape Completion · Point Cloud Completion · SkullBreak · SkullFix · Transformers

## 1 Introduction

Cranial damage is a common outcome of traffic accidents, neurosurgery, and warfare. Each year, thousands of patients require personalized cranial implants [2]. Nevertheless, the design and production of personalized implants are expensive and time-consuming. Nowadays, it requires trained employees working with

computer-aided design (CAD) software [11]. However, one part of the design pipeline, namely defect reconstruction, can be directly improved by the use of deep learning algorithms [7, 8].

The problem can be formulated as a shape completion task and solved by dedicated neural networks. Its importance motivated researchers to organize two editions of the AutoImplant challenge, during which researchers proposed several unique contributions [7, 8]. The winning contributions from the first [3] and second editions [18] proposed heavily-augmented U-Net-based networks and treated the problem as segmentation of missing skull fragment. They have shown that data augmentation is crucial to obtain reasonable results [19]. Other researchers proposed similar encoder-decoder approaches, however, without significant augmentation and thus limited performance [10, 14]. Another group of contributions attempted to address not only the raw performance but also the computational efficiency and hardware requirements. One contribution proposed an RNN-based approach using 2-D slices taking into account adjacent slices to enforce the continuity of the segmentation mask [21]. The contribution by Li *et al.* has taken into account the data sparsity and proposed a method for voxel rearrangement in coarse representation using the high-resolution templates [6]. The method was able to substantially reduce memory usage while maintaining reasonable results. Another contribution by Kroviakov *et al.* proposed an approach based on sparse convolutional neural networks [5] using Minkowski engine [1]. The method excluded the empty voxels from the input volume and decreased the number of the required convolutions. The work by Yu *et al.* proposed an approach based on principal component analysis with great generalizability, yet limited raw performance [23]. Interestingly, methods addressing the computational efficiency could not compete, in terms of the reconstruction quality, with the resource-inefficient methods using dense volumetric representation [7].

The current state-of-the-art solutions, even though they reconstruct the defects accurately, share some common disadvantages. First, they operate in the volumetric domain and require significant computational resources. The GPU memory consumption scales cubically with the volume size. Second, the most successful solutions do not take into account data sparsity. The segmented skulls are binary and occupy only a limited part of the input volume. Thus, using methods dedicated to 3-D multi-channel volumes is resource-inefficient. Third, the final goal of the defect reconstruction is to propose models ready for printing/manufacturing. Working with volumetric representation requires further postprocessing to transfer the reconstructed defect into a manufacturable model.

Another approach, yet still unexplored, to cranial defect reconstruction is the use of deep networks dedicated to point clouds (PCs) processing. Since the introduction of PointNet [15] and PointNet++ [16], the number of contributions in the area of deep learning for PC processing exploded. Several notable contributions, like PCNet [26], PoinTr [24], AdaPoinTr [25], 3DSGrasp [12], MaS [9], have been proposed directly to the PC completion task. The goal of the PC completion is to predict a missing part of an incomplete PC.

The problem of cranial defect reconstruction can be reformulated into PC completion which has several advantages. First, the representation is sparse, and thus requires significantly less memory than the volumetric one. Second, PCs are unordered collections and can be easily splitted and combined, enabling further optimizations. Nevertheless, the current PCs completion methods focus mostly on data representing object surfaces and do not explore large-scale PCs representing solid objects.

In this work, we reformulate the problem from volumetric segmentation into PC completion. We propose a dedicated method to complete large-scale PCs representing solid objects. We extend the geometric aware transformers [24] and propose an iterative pipeline to maintain low memory consumption. We compare the proposed approach to the state-of-the-art networks for volumetric segmentation and PC completion. Our approach provides high-quality reconstructions while maintaining computational efficiency and good generalizability into previously unseen cases.

2 Methods

2.1 Overview

The input is a 3-D binary volume representing the defective skull. The output is a PC representing the missing skull fragment and (optionally) its meshed and voxelized representation. The processing pipeline consists of: (i) creating the PC from the binary volume, (ii) splitting the PC into a group of coarse PCs, (iii) calculating the missing PC by the geometric aware transformer for each group, (iv) merging the reconstructed coarse PCs, (v) optional voxelization and postprocessing for evaluation. The pipeline is shown in Fig. 1.

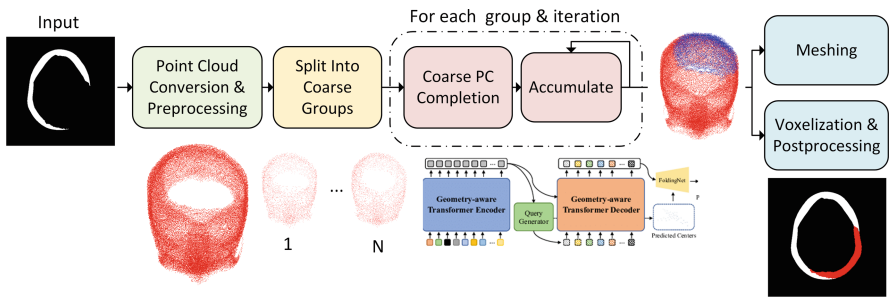


Fig. 1. Visualization of the processing pipeline.

## 2.2 Preprocessing

The preprocessing starts with converting the binary volume to the PC. The coordinates of the positive voxels are created only from the voxels representing the skull. The PC is normalized to  $[0-1]$  range, randomly permuted, and split into  $N$  equal groups, where  $N$  is calculated based on the number of points in the input PC in a manner that each group contains 32768 randomly sampled input points and outputs 16384 points. Thus, the  $N$  depends on the number of positive voxels. The higher the resolution, the more groups are being processed. The number of outputs points is lower than the input points because we assumed that the defect is smaller than the skull.

## 2.3 Network Architecture - Point Cloud Completion Transformer

We adapt and modify the geometry-aware transformers (PoinTr) [24]. The PoinTr method was proposed and evaluated on coarse PCs representing object surfaces. The full description of the PoinTr architecture is available in [24].

We modify the network by replacing the FoldingNet [22] decoder working on 2-D grids with a folding decoder operating on 3-D representation. The original formulation deforms the 2-D grid into the surface of a 3-D object, while the proposed method focuses on solid 3-D models. Moreover, we modify the original k-NN implementation (with quadratic growth of memory consumption with respect to the input size) to an iterative one, to further decrease and stabilize the GPU memory consumption.

## 2.4 Objective Function

We train the network using a fully supervised approach where the ground-truth is represented by PCs created from the skull defects. In contrast to other PC completion methods, we employ the Density Aware Chamfer Distance (DACD) [20]. The objective function enforces the uniform density of the output and handles the unpredictable ratio between the input/output PCs size. We further extend the DACD by calculating the distance between the nearest neighbours for each point and enforcing the distance to be equal. The final objective function is:

$$O(P_r, P_{gt}) = DACD(P_r, P_{gt}) + \frac{\alpha}{S} \sum_{i=0}^S \sum_{j=0}^k \sum_{l=0}^k |P_r(i) - P_r(j)| - |P_r(i) - P_r(l)|, \quad (1)$$

where  $P_r, P_{gt}$  are the reconstructed and ground-truth PC respectively,  $S$  is the number of points in  $P_{rec}$ ,  $k$  is the number of nearest neighbours of point  $i$ ,  $\alpha$  is the weighting parameter. We apply the objective function to all PC ablation studies unless explicitly stated otherwise. The volumetric ablation studies use the soft Dice score.

The traditional objective functions like Chamfer Distance (CD) [20], Extended Chamfer Distance (ECD) [22], or Earth Mover's Distance (EMD) [9]

are not well suited for the discussed application. The CD/ECD provide sub-optimal performance for point clouds with uniform density or a substantially different number of samples, tends to collapse, and results in noisy training. The EMD is more stable, however, explicitly assumes bijective mapping (requiring knowledge about the desired number of points) and has high computational complexity.

## 2.5 Iterative Completion

The coarse PCs are processed by the network separately. Afterwards, the reconstructed PCs are combined into the final reconstruction. To improve the results, the process may be repeated  $M$  times with a different initial PC split and a small Gaussian noise added. The procedure improves the method’s performance and closes empty holes in the voxelized representation. The optional multi-step completion is performed only during the inference.

The iterative completion allows one to significantly reduce the GPU memory usage and the number of network parameters. The PCs are unordered collections and can be easily split and merged. There is no need to process large PCs in one shot, resulting in the linear growth of inference time and almost constant GPU memory consumption.

## 2.6 Postprocessing

The reconstructed PCs are converted to mesh and voxelized back to the volumetric representation, mainly for evaluation purposes. The mesh is created by a rolling ball pivoting algorithm using the Open3D library [27]. The voxelization is also performed using Open3D by the PC renormalization and assigning positive values to voxels containing points in their interior. The voxelized representation is further postprocessed by binary closing and connected component analysis to choose only the largest volume. Then, the overlap area between the reconstructed defect and the defective input is subtracted from the reconstructed defect by logical operations.

## 2.7 Dataset and Experimental Setup

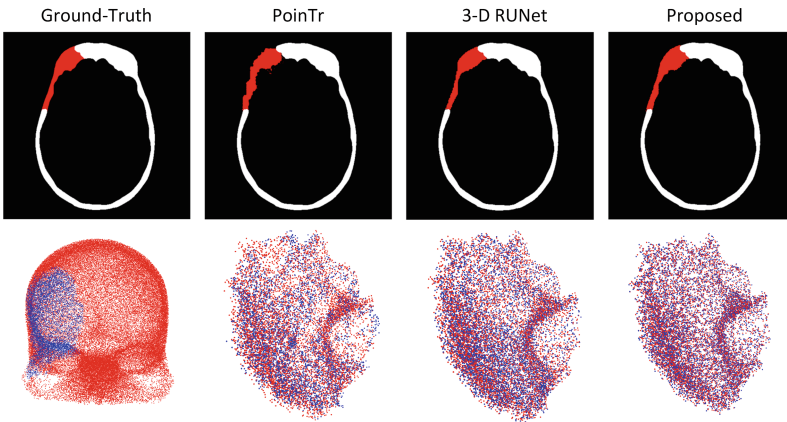
We use the SkullBreak and SkullFix datasets [4] for evaluation. The datasets were used during the AutoImplant I and II challenges and enable comparison to other reconstruction algorithms. The SkullBreak dataset contains 114 high-resolution skulls for training and 20 skulls for testing, each with 5 accompanying defects from various classes, resulting in 570 training and 100 testing cases. All volumes in the SkullBreak dataset are  $512 \times 512 \times 512$ . The SkullFix dataset is represented by 100 training cases mostly located in the back of the skull with a similar appearance, and additional 110 testing cases. The volumes in the SkullFix dataset are  $512 \times 512 \times Z$  where  $Z$  is the number of axial slices. The SkullBreak provides more heterogeneity while the SkullFix is better explored and enables direct comparison to other methods.

We perform several ablation studies. We check the influence of the input physical spacing on the reconstruction quality, training time, and GPU memory consumption. Moreover, we check the generalizability by measuring the gap between the results on the training and the external testing set for each method. We compare our method to the methods dedicated to PC completion: (i) PCNet [26], (ii) PoinTr [24], (iii) AdaPoinTr [25], as well as to methods dedicated to volumetric defect reconstruction: (i) 3-D VNet, and (ii) 3-D Residual U-Net. Moreover, we compare the reconstruction quality to results reported by other state-of-the-art methods.

We trained our network separately on the SkullBreak and SkullFix datasets. The results are reported for the external test set containing 100 cases for SkullBreak and 110 cases for the SkullFix datasets, the same as in the methods used for comparison. The models are implemented in PyTorch [13], trained using a single RTX GeForce 3090. We augment the input PCs by random permutation, cropping, rotation, and translation. The volumetric ablation studies use random rotation and translation with the same parameters as for the PCs. All the methods were trained until convergence. The hyperparameters are reported in the associated repository [17].

### 3 Results

The comparison in terms of the Dice coefficient (DSC), boundary Dice coefficient (BDSC), 95th percentile of Hausdorff distance (HD95), and Chamfer distance (CD), are shown in Table 1. Exemplary visualization, presenting both the PC and volumetric outcomes, is shown in Fig. 2.



**Fig. 2.** Exemplary visualization of the reconstructed point clouds/volumes for a case from the SkullBreak dataset. The PCs are shown for the defect only (reconstructed vs ground-truth) for the presentation clarity.

The results of the ablation studies showing the influence of the input size, generalizability, objective function, and the effect of repeating the iterative refinement are presented in Table 2.

The results present that the quantitative outcomes of the proposed method are comparable to the state-of-the-art methods, however, with significantly lower GPU memory consumption that makes it possible to perform the reconstruction at the highest available resolution. The results are slightly worse when compared to the volumetric methods, however, significantly better than other PC-based approaches.

**Table 1.** Quantitative results on the SkullBreak and SkullFix datasets. The final results are reported for original resolution using the DACD + kNN objective function and 3 iterative refinements (- denotes that results were not reported). The methods used for comparison are reported for the most successful setup (see Table 2).

Method	SkullBreak				SkullFix				GPU Mem ↓[GB]
	DSC ↑	BDSC ↑	HD95 ↓	CD ↓	DSC ↑	BDSC ↑	HD95 ↓	CD ↓	
Point Cloud Completion									
Proposed	0.87	0.85	1.91	0.31	0.90	0.89	1.71	0.29	~2.78
PCNet [26]	0.61	0.58	5.77	1.18	0.77	0.75	3.22	0.41	~2.37
PointTr [24]	0.67	0.66	5.17	0.82	0.82	0.81	3.02	0.36	~3.11
AdaPoinTr [25]	0.66	0.64	5.29	0.84	0.81	0.81	3.05	−0.36	~3.14
Volumetric Segmentation									
3-D VNet	0.87	0.90	1.87	0.21	0.91	0.93	1.66	0.11	21.89
3-D RUNet	0.89	0.91	1.79	0.18	0.91	0.92	1.67	0.09	22.47
State-of-the-art									
Kroviakov <i>et al.</i> [5]	−	−	−	−	0.85	0.94	2.65	−	<6.00
Li <i>et al.</i> [6]	−	−	−	−	0.81	−	−	−	−
Mahdi <i>et al.</i> [10]	0.78	0.81	3.42	−	0.88	0.92	3.59	−	<6.00
Pathak <i>et al.</i> [14]	−	−	−	−	0.90	0.95	2.02	−	−
Wodzinski <i>et al.</i> [18]	0.91	0.95	1.60	−	0.93	0.95	1.48	−	<40.00
Yu <i>et al.</i> [23]	−	−	−	−	0.77	0.77	3.68	−	CPU
Ellis <i>et al.</i> [3]	−	−	−	−	0.94	−	3.60	−	−
Yang <i>et al.</i> [21]	0.85	0.89	3.52	−	−	−	−	−	−

## 4 Discussion

The reconstruction quality of the method is comparable to the volumetric networks, as shown in Table 1. Meanwhile, the proposed method takes into account the data sparsity, does not require significant computational resources, and scales well with the input size. The proposed method has good generalizability. The gap between the training and testing set is negligible, unlike the volumetric methods that easily overfit and require strong augmentation for practical use. The DACD, as well as the proposed extension, improve the reconstruction quality compared to the CD or ECD by taking into account the uniformity of the expected PC. The original PC completion methods do not scale well with the increase of PC

**Table 2.** The ablation studies related to the input size, the objective function, and the number of refinements. The results are reported for the SkullBreak dataset at the original scale (except the CD). The Gen. Gap denotes the difference between the training and testing set in terms of the DSC.

Method	DSC $\uparrow$	BDSC $\uparrow$	HD95 $\downarrow$ [mm]	CD $\downarrow$ [mm]	GPU Mem $\downarrow$ [GB]	Gen. Gap $\downarrow$ [% DSC]
Input Size (uniform voxel spacing)						
Proposed: original	0.87	0.85	1.91	0.31	$\sim 2.78$	4.18
Proposed: 1 mm	0.83	0.77	2.64	0.46	$\sim 2.69$	4.05
Proposed: 2 mm	0.74	0.71	3.89	0.67	$\sim 2.64$	4.57
Proposed: 4 mm	0.69	0.64	5.12	0.79	$\sim 2.63$	3.12
PCNet: original	–	–	–	–	$>24$	–
PCNet: 1 mm	0.37	0.33	10.57	1.76	$\sim 13.22$	1.13
PCNet: 2 mm	0.57	0.53	7.18	1.37	$\sim 5.37$	1.33
PCNet: 4 mm	0.61	0.58	5.77	1.18	$\sim 2.37$	3.07
PoinTr: original	–	–	–	–	$>24$	–
PoinTr: 1 mm	0.58	0.55	6.82	1.39	$\sim 21.41$	1.89
PoinTr: 2 mm	0.65	0.64	5.28	0.94	$\sim 6.48$	2.19
PoinTr: 4 mm	0.67	0.66	5.17	0.82	$\sim 3.11$	3.98
3-D RUNet: original	–	–	–	–	$>24$	–
3-D RUNet: 1 mm	0.89	0.91	1.79	0.18	22.47	10.11
3-D RUNet: 2 mm	0.85	0.85	2.09	0.25	7.84	14.51
3-D RUNet: 4 mm	0.76	0.77	2.89	0.63	3.78	17.48
Objective Function (proposed method, original size, 3 iters)						
DACD + kNN	0.87	0.85	1.91	0.31	$\sim 2.78$	4.18
DACD	0.85	0.81	2.78	0.42	$\sim 2.72$	4.22
ECD	0.75	0.71	4.11	0.68	$\sim 2.72$	3.99
CD	0.83	0.78	2.98	0.28	$\sim 2.72$	4.58
No. Refinements (proposed method, original size, DACD + kNN)						
1 iters	0.85	0.81	2.51	0.41	$\sim 2.78$	4.51
2 iters	0.87	0.83	1.98	0.32	$\sim 2.78$	4.18
3 iters	0.87	0.85	1.91	0.31	$\sim 2.78$	4.18
4 iters	0.87	0.85	1.90	0.30	$\sim 2.78$	4.18

size. The possible reason for this is connected with the noisy kNN graph construction when dealing with large PCs and increasing the number of neighbours is unacceptable from the computational point of view. The proposed method has almost constant memory usage, independent of the input shape, in contrast to both the volumetric methods and PC completion methods without the iterative approach. Interestingly, the proposed method outperforms other methods taking into account the data sparsity. The inference speed is slightly lower than for the volumetric methods (several seconds), however, this application does not require real-time processing and anything in the range of seconds is acceptable. However, the required memory consumption is crucial to ensure that the method can be eventually applied in clinical practice.

The disadvantages of the proposed algorithm are connected to long training time, noise at the object boundaries, and holes in the voxelized output. The FoldingNet-based decoder requires a significant number of iterations to converge,



thus resulting in training time comparable or even longer than the volumetric methods. Moreover, the voxelization of PCs results in noisy edges and holes that require further morphological postprocessing. What is important, the method has to be extended by another algorithm responsible for converting the reconstruction into implantable implant before being ready to be used in clinical setup.

In future work, we plan to further reformulate the problem and, similarly to Kroviakov *et al.* [5], use only the skull contours. Since the ultimate goal is to propose models ready for 3-D printing, the interior of the skull defect is not required to create the mesh and STL/OBJ file. Another research direction is connected to the PC augmentation to further increase the network generalizability since it was shown that heavy augmentation is crucial to obtain competitive results [3, 18]. Moreover, it is challenging to perform the qualitative evaluation of the context of clinical need and we plan to perform evaluation including clinical experts in the future research.

To conclude, we proposed a method for cranial defect reconstruction by formulating the problem as the PC completion task. The proposed algorithm achieves comparable results to the best-performing volumetric methods while requiring significantly less computational resources. We plan to further optimize the model by working directly at the skull contour and heavily augmenting the PCs.

**Acknowledgements.** The project was funded by The National Centre for Research and Development, Poland under Lider Grant no: LIDER13/0038/2022 (DeepImplant). We gratefully acknowledge Polish HPC infrastructure PLGrid support within computational grant no. PLG/2023/016239.

## References

1. Choy, C., Gwak, J., Savarese, S.: 4D spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3070–3079 (2019)
2. Dewan, M.C., et al.: Global neurosurgery: the current capacity and deficit in the provision of essential neurosurgical care. Executive Summary of the Global Neurosurgery Initiative at the Program in Global Surgery and Social Change. *J. Neurosurg.* **130**(4), 1055–1064 (2019). <https://doi.org/10.3171/2017.11.JNS171500>
3. Ellis, D.G., Aizenberg, M.R.: Deep learning using augmentation via registration: 1st place solution to the AutoImplant 2020 challenge. In: Li, J., Egger, J. (eds.) *AutoImplant 2020*. LNCS, vol. 12439, pp. 47–55. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-64327-0\\_6](https://doi.org/10.1007/978-3-030-64327-0_6)
4. Kodym, O., et al.: SkullBreak/SkullFix - dataset for automatic cranial implant design and a benchmark for volumetric shape learning tasks. *Data Brief* **35** (2021)
5. Kroviakov, A., Li, J., Egger, J.: Sparse convolutional neural network for skull reconstruction. In: Li, J., Egger, J. (eds.) *AutoImplant 2021*. LNCS, vol. 13123, pp. 80–94. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-92652-6\\_7](https://doi.org/10.1007/978-3-030-92652-6_7)

6. Li, J., Pepe, A., Gsaxner, C., Jin, Y., Egger, J.: Learning to rearrange voxels in binary segmentation masks for smooth manifold triangulation. In: Li, J., Egger, J. (eds.) *AutoImplant 2021*. LNCS, vol. 13123, pp. 45–62. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-92652-6\\_5](https://doi.org/10.1007/978-3-030-92652-6_5)
7. Li, J., Egger, J. (eds.): *AutoImplant 2021*. LNCS, vol. 13123. Springer, Cham (2021). <https://doi.org/10.1007/978-3-030-92652-6>
8. Li, J., et al.: AutoImplant 2020-first MICCAI challenge on automatic cranial implant design. *IEEE Trans. Med. Imaging* **40**(9), 2329–2342 (2021)
9. Liu, M., et al.: Morphing and sampling network for dense point cloud completion. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, pp. 11596–11603 (2020). <https://doi.org/10.1609/aaai.v34i07.6827>
10. Mahdi, H., et al.: A U-Net based system for cranial implant design with pre-processing and learned implant filtering. In: Li, J., Egger, J. (eds.) *AutoImplant 2021*. LNCS, vol. 13123, pp. 63–79. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-92652-6\\_6](https://doi.org/10.1007/978-3-030-92652-6_6)
11. Marreiros, F., et al.: Custom implant design for large cranial defects. *Int. J. Comput. Assist. Radiol. Surg.* **11**(12), 2217–2230 (2016)
12. Mohammadi, S.S., et al.: 3DSGrasp: 3D shape-completion for robotic grasp (2023). <http://arxiv.org/abs/2301.00866>
13. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. In: *Advances in Neural Information Processing Systems*, vol. 32, pp. 8024–8035. Curran Associates, Inc. (2019)
14. Pathak, S., Sindhura, C., Gorthi, R.K.S.S., Kiran, D.V., Gorthi, S.: Cranial implant design using V-Net based region of interest reconstruction. In: Li, J., Egger, J. (eds.) *AutoImplant 2021*. LNCS, vol. 13123, pp. 116–128. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-92652-6\\_10](https://doi.org/10.1007/978-3-030-92652-6_10)
15. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: deep learning on point sets for 3D classification and segmentation (2017). <http://arxiv.org/abs/1612.00593>
16. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: deep hierarchical feature learning on point sets in a metric space (2017). <http://arxiv.org/abs/1706.02413>
17. Wodzinski, M.: The associated repository (2023). [https://github.com/MWod/DeepImplant\\_MICCAI\\_2023](https://github.com/MWod/DeepImplant_MICCAI_2023). Accessed 13 July 2023
18. Wodzinski, M., Daniol, M., Hemmerling, D.: Improving the automatic cranial implant design in cranioplasty by linking different datasets. In: Li, J., Egger, J. (eds.) *AutoImplant 2021*. LNCS, vol. 13123, pp. 29–44. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-92652-6\\_4](https://doi.org/10.1007/978-3-030-92652-6_4)
19. Wodzinski, M., et al.: Deep learning-based framework for automatic cranial defect reconstruction and implant modeling. *Comput. Methods Programs Biomed.* **226**, 1–13 (2022)
20. Wu, T., et al.: Density-aware chamfer distance as a comprehensive metric for point cloud completion (2021). <http://arxiv.org/abs/2111.12702>, [cs]
21. Yang, B., Fang, K., Li, X.: Cranial implant prediction by learning an ensemble of slice-based skull completion networks. In: Li, J., Egger, J. (eds.) *AutoImplant 2021*. LNCS, vol. 13123, pp. 95–104. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-92652-6\\_8](https://doi.org/10.1007/978-3-030-92652-6_8)
22. Yang, Y., Feng, C., Shen, Y., Tian, D.: FoldingNet: point cloud auto-encoder via deep grid deformation (2018). <http://arxiv.org/abs/1712.07262>, [cs]
23. Yu, L., Li, J., Egger, J.: PCA-skull: 3D skull shape modelling using principal component analysis. In: Li, J., Egger, J. (eds.) *AutoImplant 2021*. LNCS, vol. 13123, pp. 105–115. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-92652-6\\_9](https://doi.org/10.1007/978-3-030-92652-6_9)

24. Yu, X., et al.: PoinTr: diverse point cloud completion with geometry-aware transformers (2021). <http://arxiv.org/abs/2108.08839>
25. Yu, X., et al.: AdaPoinTr: diverse point cloud completion with adaptive geometry-aware transformers (2023). <http://arxiv.org/abs/2301.04545>
26. Yuan, W., et al.: PCN: point completion network (2019). <http://arxiv.org/abs/1808.00671>
27. Zhou, Q., Park, J., Koltun, V.: Open3D: a modern library for 3D data processing. CoRR abs/1801.09847 (2018). <http://arxiv.org/abs/1801.09847>