TriFlow: Triplane Latent Conditional Flow Matching for Efficient 3D Skull Shape Completion

Zhenhong Liu[1], Xingce Wang[1]✉

[1] School of Artificial Intelligence, Beijing Normal University, Beijing, China
wangxingce@bnu.edu.cn

Abstract. Automatic generation of patient-specific skull implants is essential for efficient and reliable cranioplasty. We propose TriFlow, a conditional flow matching model operating in a compact triplane latent space for high-fidelity 3D skull completion. By encoding skull geometry into structure-preserving triplane features and learning a conditional continuous velocity field, our method enables deterministic and efficient shape generation. A triplane-aware attention module further enforces intra-plane continuity and cross-plane geometric coherence. Experiments on the SkullFix and SkullBreak datasets demonstrate that TriFlow achieves state-of-the-art accuracy while significantly reducing training and inference cost, highlighting its potential for efficient automatic skull completion and future clinical translation.

Keywords: Flow matching model · Triplane representation · 3D reconstruction · 3D shape generation

1 Introduction

Skull repair is a common neurosurgical procedure for treating cranial defects caused by trauma, craniectomy, or other pathological conditions [1]. Patient-specific cranial implants must accurately match defect boundaries to ensure post-operative stability and reduce complications; however, traditional CAD-based design relies heavily on manual modeling and expert intervention, resulting in high costs, long turnaround times, and limited scalability [2].

Skull implant design is challenging due to complex anatomy, strict boundary accuracy requirements, and large inter-patient variability, making traditional CAD workflows inefficient for clinical use. Deep learning methods, include encoder-decoder and U-Net-based architectures [3,4], transformer-based models [5,6], point cloud completion [7,8,9], symmetry-aware reconstruction [10], data-augmentation-based frameworks [11], and implicit shape representations [12,13], have been applied to skull completion, yet many fail to jointly achieve anatomical consistency, boundary accuracy, and computational efficiency.

Recent advances in deep generative modeling, including normalizing flows (NFs) [14], variational autoencoders (VAEs) [15], and generative adversarial networks (GANs) [12], have shown strong capability in modeling complex 3D shape distributions. More recently, diffusion-based methods [16] have further

improved robustness for 3D shape completion under noisy and incomplete inputs [17,13], but their iterative stochastic sampling incurs considerable computational overhead. To address this limitation, flow-based generative models, offer a more efficient alternative by learning deterministic continuous transformations from a defective skull latent to the target distribution, making them well suited for high-quality 3D skull completion with clinical efficiency requirements. Recent triplane-based generative methods, such as DiffTF [18], DiffTF++ [19] and TPA3D [20], have explored plane-wise and cross-plane attention for efficient 3D generation. Different from these general 3D generation frameworks, our work focuses on conditional cranial implant completion, where the defective skull latent guides a continuous velocity field from incomplete to complete skull geometry.

To efficiently model high-quality 3D geometry within flow frameworks, triplane representations factorize volumetric features into three orthogonal 2D feature planes, significantly reducing memory and computational cost while preserving spatial structure [21]. In this work, we propose TriFlow, a conditional triplane latent flow matching model for fast and high-quality 3D skull shape completion. The proposed method introduces the following key contributions:

1. We propose TriFlow, which performs conditional skull completion in a compact triplane latent space, enabling efficient training and fast sampling while preserving geometric consistency.
2. A triplane-aware module is introduced to jointly model plane-wise surface continuity and cross-plane geometric alignment across the xy, xz, yz planes.
3. TriFlow produces anatomically plausible and robust skull completions on the SkullFix and SkullBreak datasets with high generation efficiency.

2 Methodology

Pipeline We propose a compact triplane-based conditional flow framework for skull completion (Fig. 1). The input skull is encoded from a TSDF volume into a structure-preserving triplane latent space using a 3D autoencoder. 3D RCU extracts axis-aligned triplane features, and 2D ResBlocks refine them for SDF prediction. A conditional velocity field then transforms the defective latent into a complete skull representation by solving a conditional ODE, with plane-wise and cross-plane attention modeling dependencies among the xy, xz, and yz planes. The implant is obtained by Boolean subtraction between the decoded skull and the defective input.

2.1 Triplane latent representation

We adopt a triplane latent representation that factorizes volumetric geometry into three orthogonal 2D planes, providing an efficient alternative to dense 3D grids for modeling 3D skull geometry [22]. Compared with volumetric signed distance function (SDF) representations, triplanes significantly reduce memory and computational cost while supporting high-resolution latent encoding.

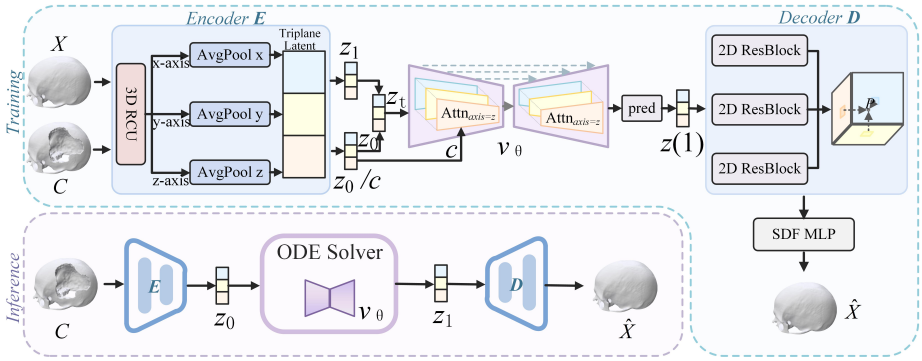

Fig. 1. The processes of our TriFlow approach.

Specifically, given a 3D skull shape, we first compute its truncated SDF (TSDF) and obtain a 3D grid $X \in \mathbb{R}^{H\times W\times D\times 3}$. An autoencoder E [23,22] is then used to encode X into a triplane latent space $Z = E(X)$, consisting of three aligned 2D feature planes $Z = (Z_{xy}, Z_{xz}, Z_{yz})$, where each component corresponds to an aligned 2D feature plane storing spatially consistent latent features. We denote a triplane latent sample by a lowercase symbol z, which concatenates the three planes in Z into a single latent tensor and serves as the state variable for subsequent flow-based modeling. A 3D residual encoder with axis-wise downsampling encodes the input into triplane latents, which are refined by lightweight 2D residual decoders and mapped to SDF values via an MLP. For each query point, features are bilinearly sampled from the three planes and converted to SDF values, from which the final surface is extracted using Marching Cubes [24].

The triplane fitting is supervised using an L2 loss between the predicted and ground-truth SDF values. To suppress spurious high-frequency artifacts and stabilize training, we further apply total variation (TV) regularization [25,26] and L2 regularization to the triplane features.

2.2 Conditional flow matching on triplane features

Let z_1 denote the triplane latent representation of a complete skull encoded by the autoencoder. In our conditional setting, the defective skull C is first encoded into a triplane latent representation $c = E(C)$, which is used to initialize the latent trajectory, i.e., $z_0 = c$. Meanwhile, the same latent code is employed as the conditional embedding c to guide the velocity field during flow matching. Flow matching constructs a continuous interpolation path between z_0 and z_1 as $z_t = (1-t)z_0 + tz_1$, where $t \in [0, 1]$.

The objective is to learn a conditional velocity field $v_\theta(z_t, t, c)$ that matches the ground-truth velocity $v^*(z_t, t) = z_1 - z_0$ along this path, under the guidance of the conditional latent code c extracted from the defective skull. The model is

trained by minimizing the following regression objective:

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{z_0,z_1,t,c}\left[\|v_\theta(z_t,t,c) - (z_1 - z_0)\|_2^2\right], \tag{1}$$

where t is uniformly sampled from $[0,1]$ during training.

During inference, generation is performed by solving the following ordinary differential equation:

$$\frac{dz_t}{dt} = v_\theta(z_t,t,c). \tag{2}$$

We employ a fixed-step Euler solver with K integration steps to obtain the final latent code $z_{t=1}$, which is then decoded into a complete skull representation.

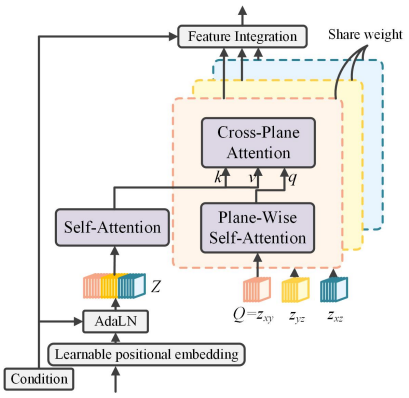

Fig. 2. Detailed structure of our triplane-aware module.

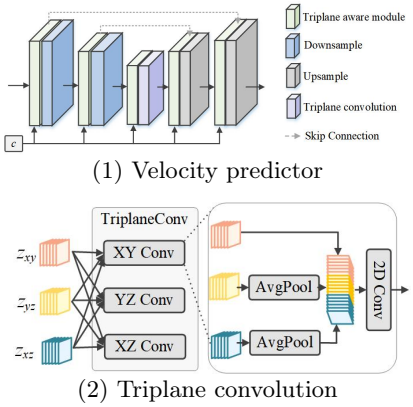

Fig. 3. Network structure of the velocity predictor.

2.3 Triplane-aware module

To effectively model global 3D geometry and enforce consistency across different triplane views, we design a triplane-aware module (Fig. 2) that integrates plane-wise self-attention, cross-plane attention, and AdaLN-based conditional modulation.

Given three feature planes Z_{xy}, Z_{xz}, and Z_{yz}, we first flatten each plane into tokens and add learnable positional embeddings to preserve plane-specific spatial locations. Plane-wise self-attention is applied independently within each plane to capture long-range intraplane dependencies. Cross-plane attention is then used to exchange information among the xy, xz, and yz planes, thereby improving interplane geometric consistency.

To incorporate patient-specific shape information, the defective skull C is encoded into a latent embedding c, which is injected into the attention blocks through adaptive layer normalization (AdaLN) [27]. Specifically, c modulates the

scale and shift parameters of normalized triplane features, allowing the attention responses to adapt to the anatomical context of the defective skull.

Finally, the attended tokens are reshaped to their original plane resolutions and fused with the input features through residual addition and a 2D projection layer. By modeling intraplane and interplane dependencies under conditional guidance, the triplane-aware module improves spatial consistency, boundary smoothness, and anatomically plausible 3D skull reconstruction.

2.4 Conditional velocity field network

The velocity field network v_θ adopts a U-Net architecture on triplane latent features treated as 2D maps. It includes an encoder, a middle block, and a decoder with skip connections to preserve multi-scale spatial information.

To model dependencies across the three orthogonal planes, we integrate the triplane-aware module into the velocity predictor and introduce a triplane convolution in the middle block. Specifically, average pooling is performed along shared axes, such as the shared x axis between the xy and xz planes. The pooled features are broadcast back to the corresponding plane resolution, concatenated with the original plane features, and projected by a final 2D convolution. This improves cross-plane communication while keeping computation in the 2D feature space.

The defective skull C is encoded as a conditional embedding and injected via adaptive normalization to guide the learned flow dynamics. Fig. 3 illustrates the overall network structure.

3 Experiments

Datasets We evaluated our method on SkullFix [28], SkullBreak [28], and MUG500 [29]. For SkullFix and SkullBreak, we use the official training/testing splits of 100/110 and 570/100 cases, respectively. Both datasets contain real skull volumes with synthetic defects generated following their official protocols. MUG500 includes 500 healthy skulls and 29 craniotomy cases with corresponding implants. We use only the 29 craniotomy cases for generalization evaluation, without including them in training.

Implementation details All tested samples were evaluated using the same hyperparameter settings. The resolution of the input 3D mesh was 256, meaning that $\max(H, W, D) = 256$. The threshold of signed distance defined as $3/256$. The spatial resolution of the encoded triplane latent signals was set to 128, meaning that $\max(H', W', D') = 128$, with 12 channels. The autoencoder was trained for $50,000$ epochs with an initial learning rate of $1e-4$ and a batch size of 4. During training of the flow matching model, the initial latent state z_0 was set to the latent code of the defective skull, and target latents z_1 were obtained from the encoder. An interpolation time $t \in [0, 1]$ was uniformly sampled to construct the intermediate latent state z_t. The velocity field was optimized using the flow matching objective with a batch size of 8 and a learning rate of 1×10^{-4}. Inference is performed via Euler-based ODE integration with 200 steps.

Experiments were conducted in PyTorch on an NVIDIA V100 (40 GB). Following prior works [2,9], we report four standard evaluation metrics, including DSC and bDSC (both in the range of $[0, 1]$), HD95 (measured in mm), and MMD.

3.1 Experimental comparison

We compared TriFlow with seven recent SoTA automatic skull reconstruction and shape completion methods, including Get3D [21], 3DU-Net [5], Diffusion-SDF [13], DiffComplete [7], CDPNet [12], Pcdiff [9] and PCDreamer [6].

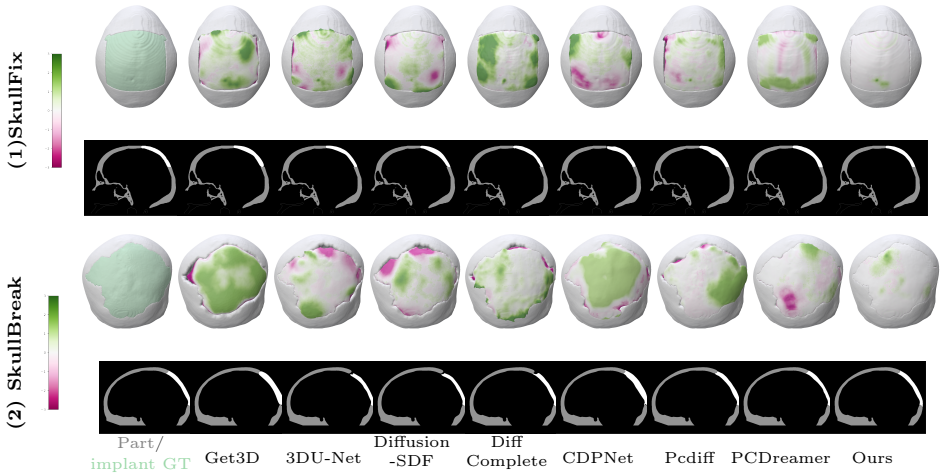

Fig. 4. Qualitative comparison of skull repair results on (1) SkullFix and (2) SkullBreak datasets.

Table 1. Results are reported using DSC, bDSC, HD95, and MMD. The MMD was multiplied by 10^2, with bold and underlined entries denoting the best and second-best results.

Method	SkullFix				SkullBreak			
	DSC↑	bDSC↑	HD95↓	MMD↓	DSC↑	bDSC↑	HD95↓	MMD↓
Get3D[21]	0.812	0.827	3.23	0.369	0.786	0.812	3.19	0.427
3DU-Net[5]	0.891	0.897	1.86	0.180	0.878	0.882	1.94	0.201
Diffusion-SDF[13]	0.873	0.885	2.04	0.236	0.868	0.873	2.16	0.243
DiffComplete[7]	0.889	0.902	1.78	0.187	0.893	0.903	1.71	0.176
CDPNet[12]	0.917	0.919	1.77	0.271	0.903	<u>0.914</u>	1.87	<u>0.144</u>
Pcdiff[9]	0.904	<u>0.922</u>	1.69	<u>0.211</u>	0.901	0.906	<u>1.78</u>	0.182
PCDreamer[6]	<u>0.928</u>	0.917	<u>1.49</u>	0.215	<u>0.913</u>	0.909	1.88	0.153
TriFlow	**0.951**	**0.955**	**1.13**	**0.108**	**0.941**	**0.948**	**1.21**	**0.117**

Comparison results Fig. 4 shows qualitative comparisons on SkullFix and SkullBreak using signed distance error colormaps and slice masks to assess geometric errors, boundary smoothness, and implant thickness. CDPNet [12], Get3D [21], and 3DU-Net [5] produce irregular surfaces and boundary artifacts, indicating limited fine-grained geometry modeling. PCDreamer [6] recovers coarse structures but often suffers from boundary misalignment and inconsistent slice thickness. DiffComplete [7] and Diffusion-SDF [13] generate smoother shapes but tend to oversmooth local anatomy. Pcdiff [9] shows noisy boundaries and jagged contours due to voxelization artifacts. In contrast, TriFlow produces anatomically coherent implants with smooth surface transitions and consistent slice-wise thickness.

As shown in Table 1, TriFlow achieves the best results on both SkullFix and SkullBreak across all metrics, with higher DSC and bDSC and lower HD95 and MMD than all compared methods. The gains are especially clear on SkullBreak, showing strong robustness to irregular defect boundaries. By learning a continuous conditional velocity field in the triplane latent space, TriFlow enables stable and geometrically faithful skull reconstruction, producing visually superior and anatomically plausible implants.

Efficiency analysis Table 2 compares TriFlow with seven representative SoTA methods in terms of training time, inference speed, parameter count, and FLOPs on the SkullBreak dataset. TriFlow achieves the lowest computational cost across all criteria, requiring only 2.8 hours for training and 1.1 seconds for generation, while using fewer parameters and significantly fewer FLOPs than all baselines. This efficiency advantage stems from the compact triplane latent representation and the deterministic conditional flow matching formulation. As a result, TriFlow enables fast and memory-efficient skull completion, making it well suited for real-world clinical deployment.

Table 2. Comparison of the computational burden.

Method	Training (h)	Generating (s)	Parameters (M)	FLOPs (G)
Get3D[21]	11.2	5.5	28.3	98
3DU-Net[5]	124	11	33.7	141
Diffusion-SDF[13]	113	12	41.2	147
DiffComplete[7]	161	15	48.3	233
CDPNet[12]	170	31	38.5	218
Pcdiff[9]	180	38	58.2	252
PCDreamer[6]	147	6	32.4	179
TriFlow	**2.8**	**1.1**	**22.4**	**83**

Generalization analysis To evaluate the generalization ability and cross-domain robustness of TriFlow, we train the model only on the synthetic SkullFix and SkullBreak datasets [28], and evaluate it on real-world MUG500 [29] dataset. As shown in Fig. 5, TriFlow maintains strong performance when transferred to real-world data. On the MUG500 dataset, TriFlow attains DSC 0.911, bDSC 0.913, HD95 1.77, and MMD 0.136. By integrating regularized triplane latents

with conditional flow matching and cross-plane attention, TriFlow achieves robust and geometrically coherent skull reconstruction across domains.

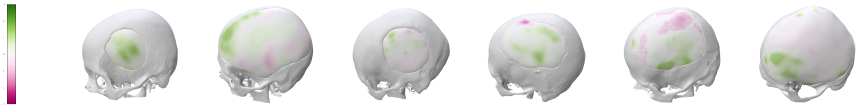

Fig. 5. Generalization performance of TriFlow on the MUG500 datasets.

3.2 Ablation study

We conduct ablation studies on the SkullFix [28] dataset to analyze the contributions of the triplane-aware module and the triplane fitting regularization (Table 3). Progressively introducing plane-wise self-attention, cross-plane attention, and triplane convolution consistently improves performance over the 2D convolution baseline, confirming the importance of explicitly modeling cross-plane geometric dependencies. Conditional guidance is essential for accurate generation, and replacing feature concatenation with AdaLN further improves all metrics, indicating more effective integration of defect-specific information. Removing TV and L2 regularization degrades performance, particularly in boundary-related metrics, while jointly applying both yields the best overall results by suppressing high-frequency artifacts and stabilizing training. Table 5 shows that the

Table 3. Ablation study of the triplane-aware module on SkullFix dataset.

Architecture Variant	DSC↑	bDSC↑	HD95↓	MMD↓
2D Conv Baseline	0.764	0.783	3.35	0.483
+ Plane-Wise Self Attention	0.861	0.863	2.03	0.259
+ Cross-Plane Attention	0.898	0.896	1.82	0.213
+ Triplane Convolution	0.907	0.906	1.68	0.165
W/ Conditional Feature (Concatenation + Std. LN)	0.934	0.936	1.40	0.125
W/ Conditional Feature (AdaLN)(Ours)	**0.951**	**0.955**	**1.13**	**0.108**

performance of TriFlow is sensitive to the choice of integration steps, batch size, and learning rate of flow matching model. A smaller batch size yields better results, with batch size of 8 consistently achieving the highest DSC and bDSC. A learning rate of 1e−4 provides the best balance between convergence speed and optimization stability, resulting in the lowest HD95 and MMD. Increasing the number of integration steps improves reconstruction quality but also increases inference cost; therefore, we adopt integration steps = 200, batch size = 8, and learning rate = 1e−4 as the default configuration.

Table 4. Impact of TV and L2 regularization on final model performance.

	DSC↑	bDSC↑	HD95↓	MMD↓
W/O TV/L2	0.834	0.869	2.32	0.219
W/ TV only	0.929	0.927	1.74	0.163
W/ L2 only	0.927	0.931	1.72	0.157
W/ TV & L2	**0.951**	**0.955**	**1.13**	**0.108**

Table 5. Ablation results with different hyperparameters.

steps	BS	LR	DSC↑	bDSC↑	HD95↓	MMD↓
100	8	1e-5	0.921	0.919	1.62	0.150
200	8	1e-5	0.938	0.943	1.27	0.126
200	**8**	**1e-4**	0.948	**0.955**	**1.13**	**0.108**
200	16	1e-4	0.941	0.948	1.23	0.115
200	24	1e-4	0.943	0.942	1.25	0.116
300	8	1e-4	**0.951**	0.954	**1.13**	0.109

Conclusion

We present a conditional flow matching framework for automatic defective skull completion. By combining triplane encoding with triplane-aware module and latent regularization, the proposed method achieves efficient and geometrically consistent reconstruction. TriFlow enables fast and anatomically reliable skull completion, showing potential for future clinical translation.

Acknowledgments. This research was partially supported by the Beijing Municipal Science and Technology Commission and Zhongguancun Science Park Management Committee (no. Z221100002722020), the National Nature Science Foundation of China (no. 62072045), and the National Nature Science Foundation of Beijing (no. 7242167).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. J. Li, G. Von Campe, A. Pepe, C. Gsaxner, E. Wang, X. Chen, et al., Automatic skull defect restoration and cranial implant generation for cranioplasty, Medical Image Analysis 73 (2021) 102171.
2. J. Li, D. G. Ellis, O. Kodym, L. Rauschenbach, C. Rieß, U. Sure, et al., Towards clinical applicability and computational efficiency in automatic cranial implant design: An overview of the autoimplant 2021 cranial implant design challenge, Medical Image Analysis 88 (2023) 102865.
3. H. Shi, X. Chen, Cranial implant design through multiaxial slice inpainting using deep learning, in: Towards the Automatization of Cranial Implant Design in Cranioplasty: First Challenge, AutoImplant 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 8, 2020, Proceedings 1, 2020, pp. 28–36.
4. O. Kodym, M. Španěl, A. Herout, Cranial defect reconstruction using cascaded CNN with alignment, in: Towards the Automatization of Cranial Implant Design in Cranioplasty, 2020, pp. 56–64.
5. S. Resmi, R. P. Singh, K. Palaniappan, Automatic skull shape completion of defective skulls using transformers for cranial implant design, Procedia Computer Science 235 (2024) 3305–3314.

6. G. Wei, Y. Feng, L. Ma, C. Wang, Y. Zhou, C. Li, Pcdreamer: Point cloud completion through multi-view diffusion priors, in: Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 27243–27253.
7. R. Chu, et al., DiffComplete: Diffusion-based generative 3D shape completion, Advances In Neural Information Processing Systems 36 (2024).
8. W. Zhang, H. Zhou, Z. Dong, J. Liu, Q. Yan, C. Xiao, Point cloud completion via skeleton-detail transformer, IEEE Transactions on Visualization and Computer Graphics 29 (10) (2022) 4229–4242.
9. P. Friedrich, J. Wolleb, F. Bieder, F. M. Thieringer, P. C. Cattin, Point cloud diffusion models for automatic implant generation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, 2023, pp. 112–122.
10. M. Wodzinski, M. Daniol, D. Hemmerling, Automatic skull reconstruction by deep learnable symmetry enforcement, Computer Methods and Programs in Biomedicine 263 (2025) 108670.
11. M. Wodzinski, K. Kwarciak, M. Daniol, D. Hemmerling, Improving deep learning-based automatic cranial defect reconstruction by heavy data augmentation: From image registration to latent diffusion models, Computers in Biology and Medicine 182 (2024) 109129.
12. Z. Du, J. Dou, Z. Liu, J. Wei, G. Wang, N. Xie, Y. Yang, Cdpnet: Cross-modal dual phases network for point cloud completion, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, 2024, pp. 1635–1643.
13. G. Chou, Y. Bahat, F. Heide, Diffusion-SDF: Conditional generative modeling of signed distance functions, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 2262–2272.
14. G. Yang, X. Huang, Z. Hao, M.-Y. Liu, S. Belongie, B. Hariharan, Pointflow: 3D point cloud generation with continuous normalizing flows, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 4541–4550.
15. S. Molnár, L. Tamás, Variational autoencoders for 3D data processing, Artificial Intelligence Review 57 (2) (2024) 42.
16. J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, Advances In Neural Information Processing Systems 33 (2020) 6840–6851.
17. T. Wang, B. Zhang, T. Zhang, S. Gu, J. Bao, T. Baltrusaitis, et al., Rodin: A generative model for sculpting 3D digital avatars using diffusion, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 4563–4573.
18. Z. Cao, F. Hong, T. Wu, L. Pan, Z. Liu, Large-vocabulary 3d diffusion model with transformer, in: International Conference on Learning Representations, Vol. 2024, 2024, pp. 14876–14887.
19. Z. Cao, F. Hong, T. Wu, L. Pan, Z. Liu, Difftf++: 3d-aware diffusion transformer for large-vocabulary 3d generation, IEEE Transactions on Pattern Analysis and Machine Intelligence 47 (4) (2025) 3018–3030.
20. B.-S. Wu, H.-E. Chen, S.-Y. Huang, Y.-C. F. Wang, Tpa3d: Triplane attention for fast text-to-3d generation, in: European Conference on Computer Vision, 2024, pp. 438–455.
21. J. Gao, T. Shen, Z. Wang, W. Chen, K. Yin, D. Li, O. Litany, Z. Gojcic, S. Fidler, Get3D: A generative model of high quality 3D textured shapes learned from images, Advances In Neural Information Processing Systems 35 (2022) 31841–31854.
22. E. R. Chan, C. Z. Lin, M. A. Chan, K. Nagano, B. Pan, S. De Mello, et al., Efficient geometry-aware 3D generative adversarial networks, in: Proceedings of

the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 16123–16133.
23. D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv:1312.6114 (2013).
24. W. E. Lorensen, H. E. Cline, Marching cubes: A high resolution 3D surface construction algorithm, Computer graphics 21 (1) (1987) 7–12.
25. N. Dey, L. Blanc-Feraud, C. Zimmer, P. Roux, Z. Kam, J.-C. Olivo-Marin, J. Zerubia, Richardson–lucy algorithm with total variation regularization for 3D confocal microscope deconvolution, Microscopy research and technique 69 (4) (2006) 260–266.
26. J. R. Shue, E. R. Chan, R. Po, Z. Ankner, J. Wu, G. Wetzstein, 3D neural field generation using triplane diffusion, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 20875–20886.
27. W. Peebles, S. Xie, Scalable diffusion models with transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 4195–4205.
28. O. Kodym, J. Li, A. Pepe, C. Gsaxner, S. Chilamkurthy, J. Egger, M. Španěl, SkullBreak/SkullFix–dataset for automatic cranial implant design and a benchmark for volumetric shape learning tasks, Data in Brief 35 (2021) 106902.
29. J. Li, M. Krall, F. Trummer, A. R. Memon, A. Pepe, C. Gsaxner, et al., MUG500+: Database of 500 high-resolution healthy human skulls and 29 craniotomy skulls and implants, Data in Brief 39 (2021) 107524.