

Boosting 3D Liver Shape Datasets with Diffusion Models and Implicit Neural Representations

Khoa Tuan Nguyen^{1,2}, Francesca Tozzi^{4,5}, Wouter Willaert^{4,6},
Joris Vankerschaver^{2,3}, Niki Rashidian^{4,7}, and Wesley De Neve^{1,2}

¹ IDLab, ELIS, Ghent University, Ghent, Belgium

² Center for Biosystems and Biotech Data Science, Ghent University Global Campus,
Incheon, Korea

³ Department of Mathematics, Computer Science and Statistics, Ghent University,
Ghent, Belgium

`{khoatuan.nguyen,joris.vankerschaver,wesley.deneve}@ghent.ac.kr`

⁴ Department of Human Structure and Repair, Ghent University, Ghent, Belgium

⁵ Department of Hepatobiliary Surgery, Centre Hospitalier de l'Université de
Montréal, Canada

⁶ Department of GI Surgery, Ghent University Hospital, Ghent, Belgium

⁷ Department of HPB Surgery & Liver Transplantation, Ghent University Hospital,
Ghent, Belgium

`{francesca.tozzi,wouter.willaert,nikdokht.rashidian}@ugent.be`

Abstract. While the availability of open 3D medical shape datasets is increasing, offering substantial benefits to the research community, we have found that many of these datasets are, unfortunately, disorganized and contain artifacts. These issues limit the development and training of robust models, particularly for accurate 3D reconstruction tasks. In this paper, we examine the current state of available 3D liver shape datasets and propose a solution using diffusion models combined with implicit neural representations (INRs) to augment and expand existing datasets. Our approach utilizes the generative capabilities of diffusion models to create realistic, diverse 3D liver shapes, capturing a wide range of anatomical variations and addressing the problem of data scarcity. Experimental results indicate that our method enhances dataset diversity, providing a scalable solution to improve the accuracy and reliability of 3D liver reconstruction and generation in medical applications. Finally, we suggest that diffusion models can also be applied to other downstream tasks in 3D medical imaging. Our code is available at https://github.com/Khoa-NT/hyperdiffusion_liver.

Keywords: 3D liver generation · 3D liver reconstruction · Diffusion · Implicit Neural Representations · Laparoscopy.

1 Introduction

Motivation. We are focused on 3D generation of human abdominal organs, particularly the liver, and during the preparation of a 3D liver dataset, we discovered that pretrained diffusion models [10] cannot be utilized because their

training datasets [3,23] lack organ-specific information. Thus, we need to train from scratch using 3D medical shape datasets [11,12,18]. However, after analysis, we found that these datasets cannot be fully utilized due to the presence of artifacts. To address the problem of insufficient 3D objects¹, we propose a new unconditional 3D liver HyperDiffusion model to generate synthetic datasets that supplement the limited amount of available real liver data.

Implicit Neural Representations (INRs). INRs have recently emerged as a robust method for encoding various continuous functions, including images, video sequences, audio sequences, and 3D objects, utilizing a Multi-Layer Perceptron (MLP) for representation learning [16,25,28]. When applied to complex 3D objects, INRs overcome the limitations of conventional discrete representations, such as point clouds or meshes [14]. However, INRs lack generalizability because MLPs tend to suffer from overfitting to individual objects, making it impractical to train a new MLP for each new object [19].

HyperNetworks. The use of a HyperNetwork [6] refers to the use of an approach that leverages one network (called a HyperNetwork) to generate the weights for another network (called the main network), which eliminates the need to train the main network from scratch. Recent studies have demonstrated the application of this method in optimizing weights [20], generating audio samples [27], and constructing 3D [4,8] generative models. Recognizing the potential of this approach, we utilize HyperDiffusion [4], a denoising diffusion model [7] designed to generate the weights of a MLP that represents a 3D object using the INR paradigm. We apply this model to address the generalization problem in INRs and obtain accurate synthetic 3D liver objects from the generated MLP.

Our Contributions. In this paper, we analyze various overlooked issues in 3D medical shape datasets and propose an approach to supplement the limited amount of available 3D liver data by creating a hybrid 3D liver dataset that combines high-quality real 3D liver objects with synthetic ones. As part of our validation, we conducted a survey where experts classified most of our synthesized 3D liver objects as real, demonstrating the quality and fidelity of our synthetic dataset.

Currently, few, if any, detailed investigations into the underlying causes of 3D liver generative model phenomena have been published. The most closely related study sharing a similar use case with ours is the recent work on TrIND [24], which focuses on anatomical trees, whereas our work concentrates on abdominal organs, particularly the liver.

2 Dataset Analysis

Currently, the Laion-5b [23] and Objaverse-XL [3] datasets are widely used to train text-image or text-3D pair diffusion models at scale [22]. Objaverse-XL, which contains over 10 million 3D objects, is the largest dataset in terms of both scale and diversity, driving recent advancements in 3D generation [10,22] and 3D

¹ We use ‘**object**’ to distinguish the 3D model from the term ‘**model**’ in deep learning.

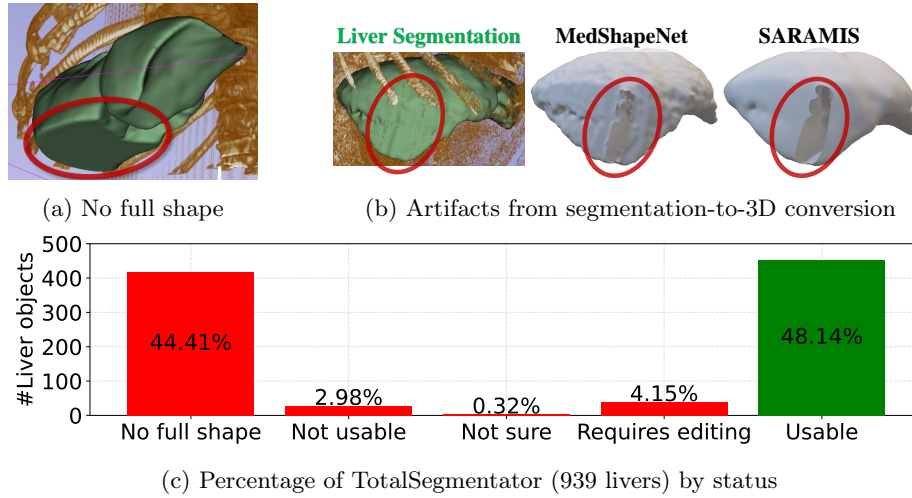


Fig. 1: Our analysis of TotalSegmentator, MedShapeNet, and SARAMIS shows that current 3D medical shape datasets are largely built on 3D segmentation results, such as those from TotalSegmentator. This approach results in disorganized datasets, with usable liver objects (labeled ‘Usable’; represented as a **green bar**) accounting for only 48.14%, while the others (labeled ‘No full shape’, ‘Not usable’, ‘Not sure’, and ‘Requires editing’; represented as **red bars**) are prone to artifacts and largely unusable in the case of TotalSegmentator.

reconstruction [30]. However, it lacks real 3D medical shapes for applications in the medical field. Fortunately, several open 3D medical shape datasets have been released, including SARAMIS [18], MedShapeNet [11], and AbdomenAtlas [12]. Upon analysis, we found that these datasets generate 3D objects based on 3D organ segmentations from CT scans, with the most commonly used dataset being the TotalSegmentator dataset [29]. Therefore, in this study, we analyzed the TotalSegmentator dataset in more detail and identified the following problems, as shown in Fig. 1.

The first problem is that the 3D objects are incomplete. Since 3D organ segmentation does not require complete shapes, the resulting objects are often missing parts, which we refer to as ‘No full shape’. For instance, the liver *s0046* in Fig. 1a is incomplete, missing the lower part due to the CT scan being taken in the thoracic region.

The second problem is the presence of artifacts during the segmentation-to-3D conversion process. Even when the 3D segmentation is correct, the exported 3D object can have artifacts, such as holes, as shown by liver *s0095* in Fig. 1b. After resolving the artifacts through a proper segmentation-to-3D conversion in 3D Slicer²[5], we requested our collaborating surgical team at Ghent University Hospital to manually review 939 liver objects exported from 1228 CT

² <https://www.slicer.org>

scan subjects in the TotalSegmentator dataset and categorize them by status. They determined that only 48.14% (452 objects) are usable ('Usable'), while the remaining are categorized as follows: 44.41% (417 objects) as 'No full shape', 2.98% (28 objects) as 'Not usable' due to not resembling a real liver, 0.32% (3 objects) as 'Not sure', and 4.15% (39 objects) as 'Requires editing', such as cases where segmentation errors caused two organs to merge, needing re-segmentation or manual separation. Fig. 1c shows the overall distribution.

With the goal of leveraging high-quality 3D objects from these 3D medical shape datasets, the problems we identified (which may not be limited to the TotalSegmentator dataset but could also apply to other CT datasets) can lead to errors during use. For example, in the 3D liver generation task, generated livers may not be in full shape due to the dataset containing 'No full shape' objects. Therefore, we follow recent works [2,31] to address current dataset insufficiencies and propose generating synthetic datasets, either as 3D object datasets or INR datasets [15], rather than investing time in analyzing additional datasets.

3 Synthetic Dataset Generation

We adopt the framework of HyperDiffusion [4] as a 3D liver generative model to generate additional synthetic 3D objects in the form of INRs. The framework consists of two stages: (1) training MLPs to represent each 3D liver object as an INR and (2) training a HyperDiffusion model in the space of MLP weights.

Once the HyperDiffusion model is trained, new MLP weights corresponding to valid INRs can be synthesized through the reverse diffusion process from a randomly sampled noise signal. The generated 3D liver objects are then reconstructed using the Marching Cubes (MC) algorithm³[13]. Fig. 2 illustrates our workflow.

3.1 Instance Liver MLP

In the first stage, for each surface \mathbf{S}_i of a 3D object in the training set $\{\mathbf{S}_i | i = 1, \dots, N\}$, we train a MLP to predict the occupancy values by minimizing the binary cross-entropy loss function:

$$L = \text{BCE}(f(\mathbf{x}, \theta_i), \mathbf{o}_i(\mathbf{x})) \text{ for } \mathbf{x} \in \mathbb{R}^3, \quad (1)$$

where f is a MLP parametrized by $\theta_i \in \mathbb{R}$, \mathbf{x} represents a 3D coordinate, and $\mathbf{o}_i(\mathbf{x}) \in \{0, 1\}$ is the ground-truth occupancy of \mathbf{x} (1 for inside and 0 for outside) with respect to \mathbf{S}_i .

MLP Architecture. The MLP architecture is a regular fully connected network, where the input is a 3D coordinate concatenated with a positional encoding [17]. It consists of three hidden layers, each with 128 neurons and ReLU activation functions. The output is a scalar occupancy value [16]. In addition, we use the same MLP architecture for different 3D objects in the training dataset.

³ `marching_cubes` from `scikit-image`

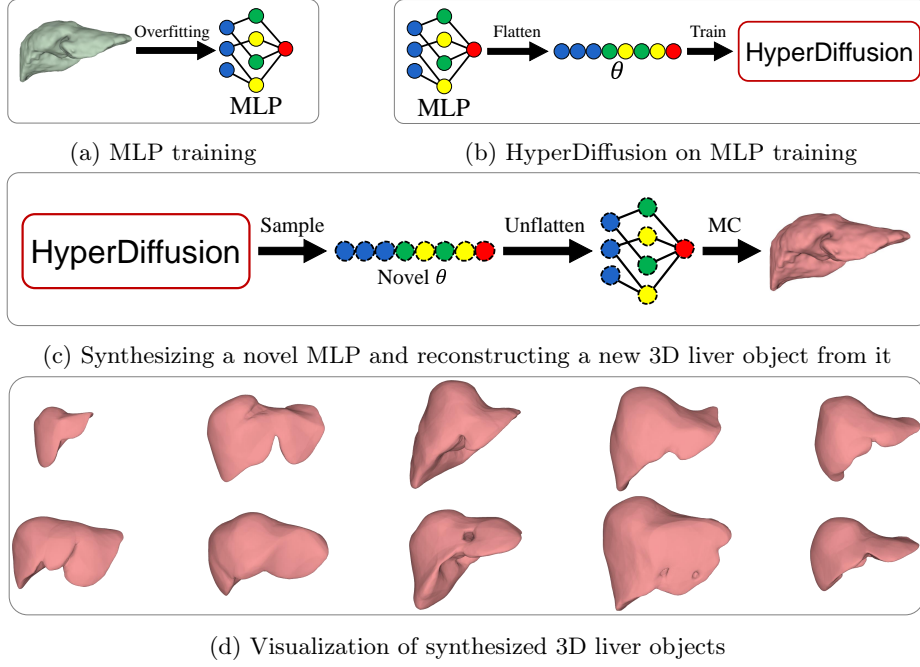


Fig. 2: Overview of the proposed 3D liver HyperDiffusion framework. Our framework consists of two training stages: (a) MLP training and (b) HyperDiffusion on MLP training. (c) To synthesize a novel 3D liver object, we use the optimized HyperDiffusion network to sample the novel θ from the diffusion process. The novel θ is reshaped back to an MLP and used to reconstruct the novel 3D liver object through the Marching Cubes (MC) algorithm. (d) Visualization of the novel 3D liver objects.

However, each MLP is optimized independently for a single 3D object, meaning there is no parameter sharing among the training set.

MLP Training. For MLP training, we normalize all 3D objects to fit within the volume range $[-0.5, 0.5]^3$ and randomly sample $20k$ points within this volume. To enhance surface precision at a finer scale, we additionally sample another $20k$ points near the surface, resulting in a total of $40k$ sampled points. To distinguish whether a point is inside or outside the surface, we compute the generalized winding numbers⁴[1] to obtain the ground-truth occupancy values. We train each MLP for 1000 epochs with a mini-batch size of 2048 points per epoch, which takes approximately 1 minute per 3D object. For inference, to reconstruct the 3D liver object, we apply the MC algorithm to extract the surface from the predicted occupancy values. The workflow is illustrated in Fig. 3. All of the optimized MLPs will serve as the training set for the next stage.

⁴ `fast_winding_number_for_meshes` from libigl

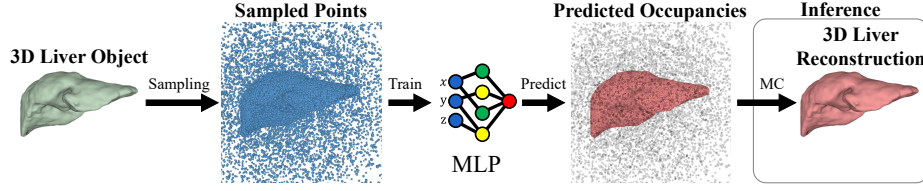


Fig. 3: Our workflow used for MLP training. From the 3D liver object, we sample a point cloud both inside and outside the object’s surface. Given the coordinates of each point as input, the MLP predicts the occupancy value: 1 for inside (points colored in red) and 0 for outside (points colored in gray). To reconstruct the 3D liver object for inference, we use the MC algorithm on the occupancy volume.

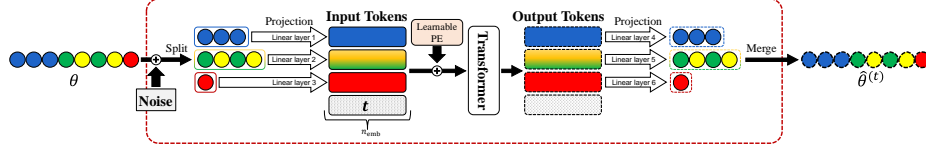


Fig. 4: The operation in HyperDiffusion during training. Random Gaussian noise is added to the flattened θ vector, which is then split into a sequence of tokens through projection. An additional time step token is appended to the sequence to indicate the current time step t , and a learnable Positional Encoding (PE) is then added to the sequence before being fed into the Transformer. The output tokens are projected back to the size of the weights and biases to merge and represent the denoised weights $\hat{\theta}^{(t)}$.

3.2 HyperDiffusion on Liver MLPs

In the second stage, we train a diffusion model on the optimized MLP weights and biases $\{\theta_i\}_{i=1}^N$ to model the space of weights through the diffusion process. We use the Transformer-based diffusion model from G.pt [20] and HyperDiffusion [4] as the baseline for our HyperDiffusion on liver MLPs, as shown in Fig. 4.

HyperDiffusion Training. Given an MLP as input into HyperDiffusion, we flatten it into a 1D vector θ and add Gaussian noise at time step $t \in [1, T = 1000]$ as part of the forward diffusion process. The noisy θ vector is then split into a sequence of tokens, where the sequence length corresponds to the number of weight and bias tensors in θ : [3456, 128, 16384, 128, 16384, 128, 128, 1] resulting in a sequence length of 8, according to our MLP design. Since each weight and bias tensor has a different size, a projection step is required to map them all to a uniform embedding size of $n_{\text{emb}} = 2880$. Each weight and bias tensor has a corresponding linear projection layer, and these layers are not shared among them. A sinusoidal embedding of the time step t is concatenated as an additional token, and all the input tokens are then combined with a learnable Positional Encoding (PE). The output tokens are then projected back to the original size and merged into a 1D vector, representing the denoised weights $\hat{\theta}^{(t)}$ at time step t .

Table 1: Quantitative liver MLP results obtained for 452 3D liver objects (‘Usable’). We use the following evaluation metrics: VIoU, NC, F-Score (higher is better), and Chamfer- L_1 distance (lower is better). The mean and standard deviation scores across all metrics indicate that our MLPs effectively represent the 3D liver objects, demonstrating the quality of the learned representations.

VIoU \uparrow	Chamfer- L_1 \downarrow	NC \uparrow	F-Score \uparrow
0.9747 ± 0.0043	0.0028 ± 0.0003	0.9780 ± 0.0053	1.0000 ± 0.0000

Table 2: We scale the MMD by 10^2 , while COV and 1-NNA are measured in percentages (%). For these metrics, higher values of COV are preferred, while lower values of MMD, 1-NNA, and FPD are desirable. Specifically, an optimal 1-NNA value is 50%. Overall, the Transformer-based model outperforms the 1D UNet-based model across all metrics.

Model Type	MMD \downarrow	COV (%) \uparrow	1-NNA (%) \downarrow	FPD \downarrow
Transformer	0.24	55.88	53.68	2.27
1D UNet	0.25	42.65	68.38	6.41

The Mean Squared Error (MSE) loss is calculated between the denoised weights $\hat{\theta}^{(t)}$ and the original weights θ . The MSE loss guides the model to minimize the difference between the denoised weights and the true weights during the reverse diffusion process.

Synthesizing New 3D Liver Objects. After training, we use Denoising Diffusion Implicit Models (DDIM) [26] to sample new MLPs. Starting with a pure noise θ (initialized as random Gaussian noise), the DDIM process gradually denoises it over several steps, progressively moving it towards a valid MLP that represents a new 3D liver object.

4 Experiments

Setup. Recalling Fig. 1 from Section 2, we use 452 usable 3D objects (‘Usable’) as our liver dataset. We first train MLPs on the entire dataset to obtain a collection of 452 liver MLPs. The set of 452 liver MLPs is then split into non-overlapping partitions: training (80%), validation (5%), and testing (15%) subsets for HyperDiffusion. We train HyperDiffusion for 6000 epochs using the AdamW optimizer with a batch size of 32 and a learning rate of $2e^{-4}$ on a single A6000 GPU, which takes approximately 9 hours. We also trained a 1D UNet-based model⁵ to compare its performance with the transformer-based model by replacing the Transformer in HyperDiffusion with a 1D UNet. This takes approximately 22 hours to converge after 12000 epochs.

Evaluation Metrics. We follow [9,16,21] to evaluate the quality of 3D reconstructions from liver MLPs using the following metrics: Volumetric Intersection

⁵ from <https://github.com/lucidrains/denoising-diffusion-pytorch>

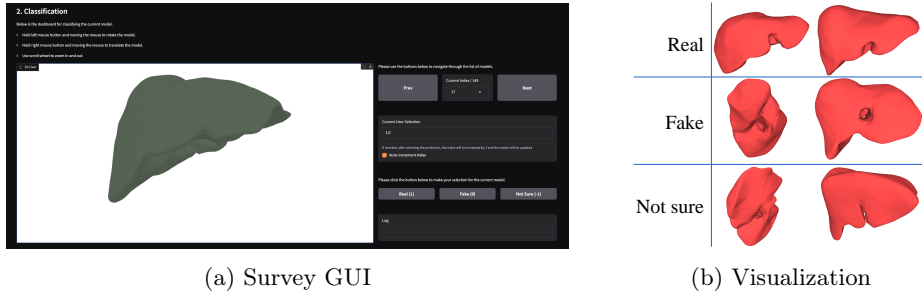


Fig. 5: (a) The GUI allows the user to view one 3D liver object at a time and navigate among 150 objects to review and select from the options – ‘Real’, ‘Fake’, and ‘Not sure’ – to classify the current 3D liver object. (b) Visualization of two synthesized 3D liver objects for each type of classification.

over Union (VIOU), Chamfer- L_1 distance, Normal Consistency (NC), and F-Score with a default threshold of 1%. The quantitative results in Table 1 show that our liver MLPs achieve high scores, nearly 1, in VIOU and NC. Additionally, we achieve the highest score in F-Score (with a maximum value of 1) and a Chamfer- L_1 distance close to zero. These results demonstrate that our liver MLPs effectively represent the 3D liver objects, providing a high-quality dataset for training HyperDiffusion.

To evaluate the quality of the reconstructed 3D liver object synthesis, we use the metrics from HyperDiffusion [4]: Minimum Matching Distance (MMD), Coverage (COV), 1-Nearest-Neighbor Accuracy (1-NNA) and Frechet Pointnet++ Distance (FPD). Since there is no specific ground truth for the synthesis, these metrics help assess how well the synthesized 3D objects match the existing real 3D liver objects. We measure the 3D liver object synthesis using the test set to evaluate how well the model generalizes to unseen data. The results in Table 2 show that the Transformer-based model is better than the 1D UNet-based model, with FPD being lower by a factor of 2.8 and 1-NNA closer to 50%. The synthesized 3D liver objects from the Transformer-based model will be used in our next assessment.

Expert Evaluation of 3D Liver Objects. We conducted an additional survey to evaluate our synthesized 3D liver objects at an expert level. We asked our surgical team to classify a collection of 150 3D liver objects, consisting of 50% real liver objects (68 from the test set and 7 from the validation set) and 50% fake liver objects (75 synthesized liver objects). As shown in Fig. 5, there are three options – Real, Fake, and Not sure – to classify all 150 3D liver objects.

After completing the survey, the selections are: ‘Real’ (139), ‘Fake’ (4), and ‘Not sure’ (7). According to the members of the surgical team, they classified a liver as real if the overall structure is well represented, even with small holes on the posterior face, and selected ‘Not sure’ only when the holes are too prominent and deform the shape. Based on the survey results, our synthesized 3D liver objects appear realistic, as all were classified as real. However, some cases contain

arbitrary holes throughout the structure, as labeled ‘Fake’ in Fig. 5b, but the overall structure is well synthesized by our 3D Liver HyperDiffusion model.

5 Conclusions

In this paper, we presented our findings on the lack of high-quality data in 3D medical shape datasets and propose using the HyperDiffusion model to generate synthetic 3D liver objects to address this issue. Our results demonstrate that the quality of the synthesized 3D liver objects aligns well with real liver structures. In future work, we plan to incorporate conditional methods to control the synthesized shapes, such as text-to-3D or image-to-3D generation.

Acknowledgments. The authors received no specific funding for this work.

Disclosure of Interests. The authors declare no competing interests.

References

1. Barill, G., Dickson, N.G., Schmidt, R., Levin, D.I., Jacobson, A.: Fast winding numbers for soups and clouds. *ACM Transactions on Graphics (TOG)* **37**(4), 1–12 (2018)
2. Bluethgen, C., Chambon, P., Delbrouck, J.B., van der Sluijs, R., Polacin, M., Zambrano Chaves, J.M., Abraham, T.M., Purohit, S., Langlotz, C.P., Chaudhari, A.S.: A vision–language foundation model for the generation of realistic chest X-ray images. *Nature Biomedical Engineering* pp. 1–13 (2024)
3. Deitke, M., Liu, R., Wallingford, M., Ngo, H., Michel, O., Kusupati, A., Fan, A., Laforte, C., Voleti, V., Gadre, S.Y., et al.: Objaverse-XL: A Universe of 10M+ 3D Objects. *Advances in Neural Information Processing Systems* **36** (2024)
4. Erkoç, Z., Ma, F., Shan, Q., Nießner, M., Dai, A.: HyperDiffusion: Generating Implicit Neural Fields with Weight-Space Diffusion. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 14300–14310 (2023)
5. Fedorov, A., Beichel, R., Kalpathy-Cramer, J., Finet, J., Fillion-Robin, J.C., Pujol, S., Bauer, C., Jennings, D., Fennessy, F., Sonka, M., et al.: 3D Slicer as an image computing platform for the Quantitative Imaging Network. *Magnetic resonance imaging* **30**(9), 1323–1341 (2012)
6. Ha, D., Dai, A.M., Le, Q.V.: Hypernetworks. In: *International Conference on Learning Representations* (2017), <https://openreview.net/forum?id=rkpACellx>
7. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. *Advances in neural information processing systems* **33**, 6840–6851 (2020)
8. Jun, H., Nichol, A.: Shap-E: Generating Conditional 3D Implicit Functions. *arXiv preprint arXiv:2305.02463* (2023)
9. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction. *ACM Transactions on Graphics (ToG)* **36**(4), 1–13 (2017)
10. Lee, H., Savva, M., Chang, A.X.: Text-to-3D Shape Generation. In: *Computer Graphics Forum*. p. e15061. Wiley Online Library (2024)

11. Li, J., Pepe, A., Gsaxner, C., Luijten, G., Jin, Y., Ambigapathy, N., Nasca, E., Solak, N., Melito, G.M., Memon, A.R., et al.: MedShapeNet—A Large-Scale Dataset of 3D Medical Shapes for Computer Vision. arXiv preprint arXiv:2308.16139 (2023)
12. Li, W., Qu, C., Chen, X., Bassi, P.R., Shi, Y., Lai, Y., Yu, Q., Xue, H., Chen, Y., Lin, X., et al.: AbdomenAtlas: A Large-Scale, Detailed-Annotated, & Multi-Center Dataset for Efficient Transfer Learning and Open Algorithmic Benchmarking. *Medical Image Analysis* p. 103285 (2024), <https://github.com/MrGiovanni/AbdomenAtlas>
13. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3D surface construction algorithm. In: *Seminal graphics: pioneering efforts that shaped the field*, pp. 347–353 (1998)
14. Luigi, L.D., Cardace, A., Spezialetti, R., Ramirez, P.Z., Salti, S., Stefano, L.D.: Deep Learning on Implicit Neural Representations of Shapes. In: *International Conference on Learning Representations (ICLR)* (2023)
15. Ma, Q., Paudel, D.P., Konukoglu, E., Van Gool, L.: Implicit Zoo: A Large-Scale Dataset of Neural Implicit Functions for 2D Images and 3D Scenes. In: *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*
16. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy Networks: Learning 3D Reconstruction in Function Space. In: *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2019)
17. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In: *ECCV* (2020)
18. Montaña-Brown, N., Saeed, S.U., Abdulaal, A., Dowrick, T., Kilic, Y., Wilkinson, S., Gao, J., Mashar, M., He, C., Stavropoulou, A., et al.: SARAMIS: Simulation Assets for Robotic Assisted and Minimally Invasive Surgery. *Advances in Neural Information Processing Systems* **36** (2024)
19. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019)
20. Peebles, W., Radosavovic, I., Brooks, T., Efros, A., Malik, J.: Learning to Learn with Generative Models of Neural Network Checkpoints. arXiv preprint arXiv:2209.12892 (2022)
21. Peng, S., Jiang, C.M., Liao, Y., Niemeyer, M., Pollefeys, M., Geiger, A.: Shape As Points: A Differentiable Poisson Solver. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2021)
22. Po, R., Yifan, W., Golyanik, V., Aberman, K., Barron, J.T., Bermano, A., Chan, E., Dekel, T., Holynski, A., Kanazawa, A., et al.: State of the Art on Diffusion Models for Visual Computing. In: *Computer Graphics Forum*. vol. 43, p. e15063. Wiley Online Library (2024)
23. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: LAION-5B: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems* **35**, 25278–25294 (2022)
24. Sinha, A., Hamarneh, G.: TrIND: Representing Anatomical Trees by Denoising Diffusion of Implicit Neural Fields. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. pp. 344–354. Springer (2024)
25. Sitzmann, V., Martel, J.N., Bergman, A.W., Lindell, D.B., Wetzstein, G.: Implicit Neural Representations with Periodic Activation Functions. In: *Proc. NeurIPS* (2020)

26. Song, J., Meng, C., Ermon, S.: Denoising Diffusion Implicit Models. In: International Conference on Learning Representations
27. Szatkowski, F., Piczak, K.J., Spurek, P., Tabor, J., Trzciński, T.: Hypernetworks build Implicit Neural Representations of Sounds. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 661–676. Springer (2023)
28. Tancik, M., Srinivasan, P.P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J.T., Ng, R.: Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. NeurIPS (2020)
29. Wasserthal, J., Breit, H.C., Meyer, M.T., Pradella, M., Hinck, D., Sauter, A.W., Heye, T., Boll, D.T., Cyriac, J., Yang, S., et al.: TotalSegmentator: Robust Segmentation of 104 Anatomic Structures in CT Images. *Radiology: Artificial Intelligence* **5**(5) (2023)
30. Yunus, R., Lenssen, J.E., Niemeyer, M., Liao, Y., Rupprecht, C., Theobalt, C., Pons-Moll, G., Huang, J.B., Golyanik, V., Ilg, E.: Recent Trends in 3D Reconstruction of General Non-Rigid Scenes. In: Computer Graphics Forum. p. e15062. Wiley Online Library (2024)
31. Zhang, Z., Yao, L., Wang, B., Jha, D., Durak, G., Keles, E., Medetalibeyoglu, A., Bagci, U.: DiffBoost: Enhancing Medical Image Segmentation via Text-Guided Diffusion Model. *IEEE Transactions on Medical Imaging* (2024)