NeRF-based CBCT Reconstruction needs Normalization and Initialization

Zhuowei Xu[‡1,2], Han Li[‡2,3,7], Dai Sun[1,2], Zhicheng Li[1,2], Yujia Li[1,2,4], Qingpeng Kong[1,2], Zhiwei Cheng[1,2], Nassir Navab[3], and S. Kevin Zhou[1,2,4,5,6⋆]

[1] School of Biomedical Engineering, Division of Life Sciences and Medicine, University of Science and Technology of China (USTC), Hefei, 230026, China
[2] Center for Medical Imaging, Robotics, Analytic Computing & Learning (MIRACLE), Suzhou Institute for Advance Research, USTC, Suzhou, 215123, China
[3] Computer Aided Medical Procedures (CAMP), TU Munich, 80333, Germany
[4] Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, 100190, China
[5] Jiangsu Provincial Key Laboratory of Multimodal Digital Twin Technology, Suzhou, 215123, China
[6] State Key Laboratory of Precision & Intelligent Chemistry, USTC, Hefei, China
[7] Munich Center for Machine Learning (MCML), Munich, Germany

Abstract. Cone Beam Computed Tomography (CBCT) is widely used in medical imaging. However, the limited number and intensity of X-ray projections make reconstruction an ill-posed problem with severe artifacts. NeRF-based methods have achieved great success in this task. However, they suffer from a **local-global training mismatch** between their two key components: the hash encoder and the neural network. Specifically, in each training step, only a subset of the hash encoder's parameters is used (**local sparse**), whereas all parameters in the neural network participate (**global dense**). Consequently, hash features generated in each step are highly misaligned, as they come from different subsets of the hash encoder. These misalignments from different training steps are then fed into the neural network, causing repeated inconsistent updates, which leads to unstable training, slower convergence, and degraded reconstruction quality. Aiming to alleviate the impact of this local-global optimization mismatch, we introduce a **Normalized Hash Encoder**, which enhances feature consistency and mitigates the mismatch. Additionally, we propose a **Mapping Consistency Initialization(MCI)** strategy that initializes the neural network before training by leveraging the global mapping property from a well-trained model. The initialized neural network exhibits improved early training stability, faster convergence and enhanced reconstruction performance. Our method is simple yet effective, requiring **only a few lines of code** while substantially improving training efficiency on 128 CT cases from 4 different datasets, covering 7 distinct anatomical regions. https://github.com/iddifficult/NI_NeRF.

Keywords: CBCT · NeRF · Hash Encoder

⋆ Corresponding author: skevinzhou@ustc.edu.cn
[1] ‡These authors contributed equally to this work.

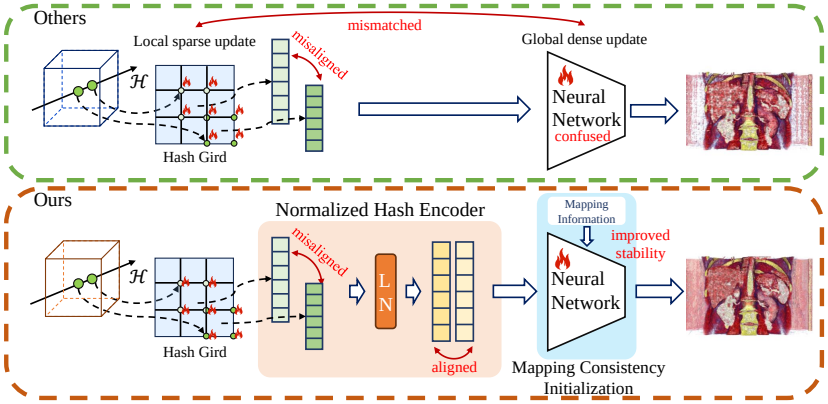

Fig. 1: This figure illustrates the update mismatch and feature misalignment problem. The small flame icons indicate the updated sections.

1 Introduction

Cone Beam Computed Tomography (CBCT) is widely used in dental, orthopedic, and interventional imaging due to its lower radiation dose, and faster scanning speed [22]. However, the potentially harmful effects of X-ray radiation limit the intensity and number of projections in CBCT scans, leading to sparse-view data acquisition. This sparsity in projections makes CBCT reconstruction an ill-posed problem, causing image degradation and severe artifacts.

To relieve this issue, numerous sparse-view CBCT reconstruction algorithms have been proposed, broadly categorized into three types: traditional methods such as FDK [7] and SART [1], supervised learning methods like iBP-Net and DIF-Net [11,13,28,14], and self-supervised methods like NeRF-based [25,21,5,27] and 3DGS-based approaches [26,4,8,12]. Among them, NeRF has emerged as a powerful self-supervised method, effectively reducing artifacts in traditional methods, eliminating the need for paired data in supervised learning, and requiring no distribution assumptions as in 3DGS-based methods that may inherently introduce false artifacts. However, the vanilla NeRF [15] architecture needs an extremely long training convergence time (hours to days) because it relies entirely on a neural network (i.e., multilayer perceptron or MLP) to learn the mapping of the entire 3D space, resulting in a massive neural network and slow convergence speed.

To address this, recent NeRF-based methods have replaced frequency encoding with hash encoding as the positional encoding, significantly reducing computational complexity and training difficulty, thereby leading to much faster convergence [27,5]. While hash encoding effectively accelerates training, it introduces a new challenge: a fundamental **local-global optimization mismatch** between the hash encoder and the neural network. Existing methods primarily

focus on designing efficient models or effective ray sampling strategies but have overlooked this critical issue. Specifically, as shown in Fig. 1, since the parameters on the hash grid are independently learnable [16], only a subset of the hash encoder's parameters is used (**local sparse**) in each training step, whereas all parameters in the neural network participate (**global dense**). Consequently, hash features generated in each step are highly misaligned, as they come from different subsets of the hash encoder. These misaligned features from different training steps are then fed into the neural network, causing repeated inconsistent global neural network updates in each training step, which leads to unstable training, slower convergence, and degraded reconstruction quality.

In this paper, we claim that Normalization and Initialization can alleviate the impact of the local-global optimization mismatch. Therefore, we introduce a **Normalized Hash Encoder**, which enhances feature consistency and mitigates the mismatch. Specifically, we add a Layer Normalization (LN) between the hash encoding model and the neural network. This ensures that the features of the hash encoding maintain a unified global mean and variance across the whole training process, thereby mitigating the misalignment problem. Additionally, we propose a **Mapping Consistency Initialization(MCI)** strategy that initializes the neural network before training by leveraging the global mapping property from a well-trained model. Sepcifically, we first train a complete NeRF-based CBCT reconstruction model on entire volume of one case and then reuse its neural network component as the initialization for other reconstruction tasks. By transferring learned knowledge across cases, the initialized neural network exhibits improved stability during early training, significantly accelerates convergence and enhances reconstruction performance.

To the best of our knowledge, we are the first to systematically investigate the **local-global optimization mismatch** and propose a simple-yet-effective and feasible method. We conduct extensive experiments on 128 CT cases collected from 4 different datasets, covering 7 distinct anatomical regions. The results show that our method not only outperforms NeRF-based methods in terms of reconstruction speed and quality but also achieves comparable reconstruction speeds to 3D Gaussian Splatting (3DGS) while surpassing 3DGS in reconstruction quality.

2 Method

2.1 Pipeline

As shown in Fig. 2, we display the complete pre-training method and the training process during reconstruction. During pre-training, since the use of ground truth information is permitted, We perform dense random sampling on the entire volume and directly supervise the entire NeRF model using the values corresponding to the sampled points on the GT. During reconstruction, we only load the Layer Normalization (LN) and the neural network weights into the new NeRF model, sample spatial points along the propagation path of the X-ray, and compute the

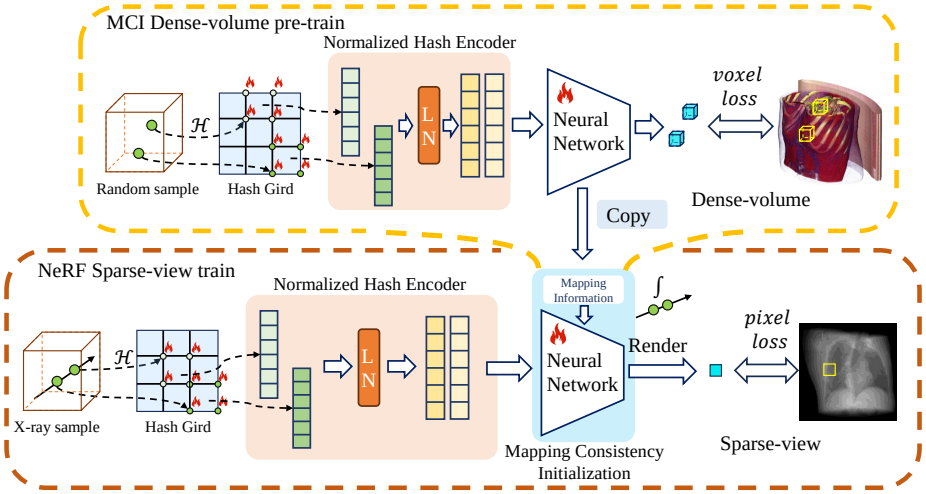

Fig. 2: Our pipeline of pre-training and normal reconstruction.

loss by integrating and comparing the values with those corresponding to the projection image.

2.2 X-ray rendering process

Due to the penetrative nature of X-rays and the known positions of the light source and detector, we adopt the Beer-Lambert Law (BLL) to replace the α-blender in the original NeRF. The BLL describes the exponential attenuation of light intensity as X-rays pass through an object.Where A is the projection value, I_0 is the initial X-ray intensity, μ_i is the attenuation coefficient at point i, and δ_i is the step size.

$$A \doteq -\ln\left(\frac{I}{I_0}\right) = \sum_{i=1}^{N} \mu_i \delta_i. \tag{1}$$

2.3 Normalized Hash Encoder

Layer Normalization (LN) scales all feature vectors within a batch to have a unified mean and variance. The formula for this transformation is as follows. To validate the impact of feature misalignment on the neural network, we record a batch of hash-encoded features every 100 epochs, along with the network's corresponding outputs. After 100 additional epochs, we reprocess the recorded features and compute the L1 error between the two inference results to assess neural network stability. As shown in Fig. 3, without Layer Normalization (LN), the neural network converges slowly and exhibits significant confusion; incorporating LN enables the network to converge quickly and stably. We place LN between the neural network and the hash encoder.

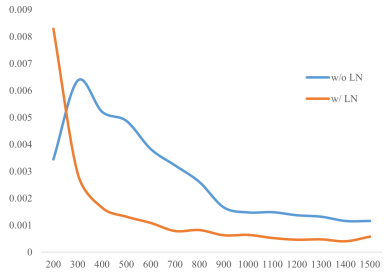

Fig. 3: Variation of MLP during training process, lower means stabler.

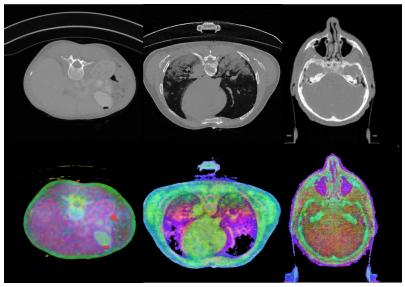

Fig. 4: Feature PCA results of abdomen, chest and head.

Furthermore, at the channel level of the hash features, we find some channels exhibit meaningless noise. Via Fast Fourier Transform (FFT), we distinguish and mask these channels during training to improve model's performance.

2.4 Mapping consistency initialization

To further investigate the function of neural network in NeRF, we employ Principal Component Analysis (PCA) to analyze the features of the hash encoding. Specifically, We use the first three principal components as the RGB channels of the image, where similar colors indicate similar features. As shown in Fig. 4 spatial points with similar attenuation coefficients(exhibit similar brightness in CT) have similar feature vectors. This finding indicates that the neural network inherently learns a consistent simple mapping in different cases, which inspired us to employ pre-training to facilitate faster and more stable model training.

To avoid the significant time consumed by traditional NeRF training, we bypass the NeRF rendering pipeline and directly supervise the attenuation coefficients of individual spatial points using a voxel-to-voxel loss, enabling an efficient dense-volume pre-training process. $\mathcal{M}$ means NeRF, x,y,z means coordinates.

$$Voxel\_loss = \|\mu_{gt} - \mathcal{M}(x, y, z)\|_1. \tag{2}$$

2.5 Sparse-view training

During training, we sample spatial points along X-ray trajectories and compute predicted projection values using the BLL. Then minimize the L1 loss between predicted and GT projections.

$$Pixel\_loss = \|A_{gt} - \sum_{i=1}^{N} \mathcal{M}(x_i, y_i, z_i)\delta_i\|_1. \tag{3}$$

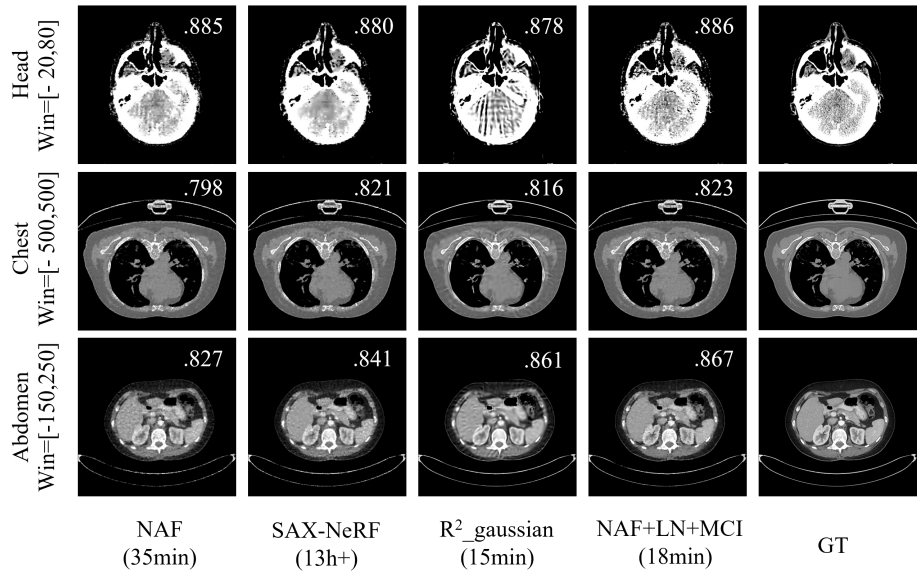

Fig. 5: Visualization of Reconstruction Results. The SSIM is displayed in the upper right corner of each image.

3 Experiments

3.1 Settings

Data We evaluate our method on four public datasets, covering common medical CT scenarios, including the head, chest, and abdomen. For the chest, we use the Covid-19 dataset [2], which includes data from 10 patients. For the abdomen, we utilize the Pancreas_CT dataset [20], comprising 82 patients. For the head, we extract 34 cases from the HAN_seg [19] dataset. All these datasets have a resolution of $512 \times 512 \times$ num_slice. Furthermore, we test four medical case in R^2_Gaussian dataset, which is in resolution of $256 \times 256 \times$ num_slice. To validate our method, we use 50 views for each dataset and conduct our projections with TIGRE toolbox [3] following NAF setting [27].

Baselines We compared our method with five baselines: FDK [7], a traditional analytical method; SART [1], a widely used iterative algorithm for sparse-view CT reconstruction; NAF [27], a NeRF variant with hash encoding; SAX-NeRF [5], a high-quality but time-consuming CT reconstruction method; and R^2_Gaussian [26], a 3DGS-based approach.

Implementation details Our experiments are implemented in PyTorch [18] and CUDA [17], trained using the Adam optimizer [6] with a learning rate of 1×10^{-3}. The batch size of view is 1, and the number of sample rays per projection is 1024, with 320 sample points per ray. We only use one abdomen case to pre-train neural network and load the same weights in all cases. For each method involved

Table 1: PSNR/SSIM score of methods on 4 datasets. Best and second-best.

Method	Chest [2]	Abdomen [20]	Head [19]
FDK [7]	19.56/.2867	21.78/.3865	24.62/.3067
SART [1]	23.75/.5855	28.05/.7645	30.42/.8821
R^2_Gaussian [26]	27.40/.7547	34.30/.9192	35.38/.9637
NAF [27]	26.12/.7149	33.53/.8961	34.09/.9533
NAF+LN	26.61/.7290	35.04/.9149	34.51/.9555
NAF+MCI	26.60/.7288	34.45/.9056	34.38/.9552
NAF+LN+MCI	27.51/.7618	35.54/.9234	34.53/.9579

Method	GS_chest	GS_foot	GS_head	GS_jaw
FDK [7]	26.28/.4967	26.22/.4479	29.35/.5753	29.73/.6524
SART [1]	31.87/.8652	30.29/.8669	35.18/.9252	33.13/.8388
R^2_Gaussian [26]	36.27/.9482	31.98/.8813	41.26/.9842	36.40/.8885
SAX_NeRF [5]	35.88/.9347	31.97/.8828	41.11/.9814	35.37/.8707
NAF [27]	34.77/.9050	31.3/.8726	40.65/.9736	34.15/.8366
NAF+LN	36.25/.9373	31.60/.8849	42.43/.9857	35.21/.8722
NAF+MCI	36.22/.9382	31.64/.8872	42.74/.9873	35.30/.8741
NAF+LN+MCI	37.18/.9484	31.64/.8861	42.85/.9885	35.36/.8745

in the testing, we train it for a sufficiently long period (3000 epochs for the NeRF-based method and 30000 epochs for R^2_Gaussian) to ensure convergence.

3.2 Reconstruction performance

Our evaluation includes two parts: (1) traditional image quality metrics (PSNR and SSIM [23]), and (2) Average Segment Dice Score. For the latter, we use TotalSegmentator [24,10], a widely adopted segmentation model, to assess the similarity between ground truth and reconstructed images [9]. Additionally, we test our method's performance under different numbers of views on three cases7.

Image quality performance Through our method, NAF achieves better image quality and the fastest reconstruction speed among NeRF-based approaches, as in Table 1. Compared to Gaussian Splatting (GS)-based methods, our method shows superior quality on most datasets and has a similar converge time (18 min). To further validate the clinical relevance of these methods, we visualize the images using different window settings, closely mimicking real clinical scenarios. The results show that GS-based methods exhibit severe artifacts and blurred organ boundaries as in Fig. 5. What's more, to demonstrate the versatility of our method, we also test its performance on SAX-NeRF, which imporves SSIM from 0.921 to 0.932.

Segment quality performance Our method makes NAF outperform much better than R^2-Gaussian on segment Dice, as in Fig. 6. These results demonstrate that our method helps NAF's reconstruction results have better anatomical structure fidelity.

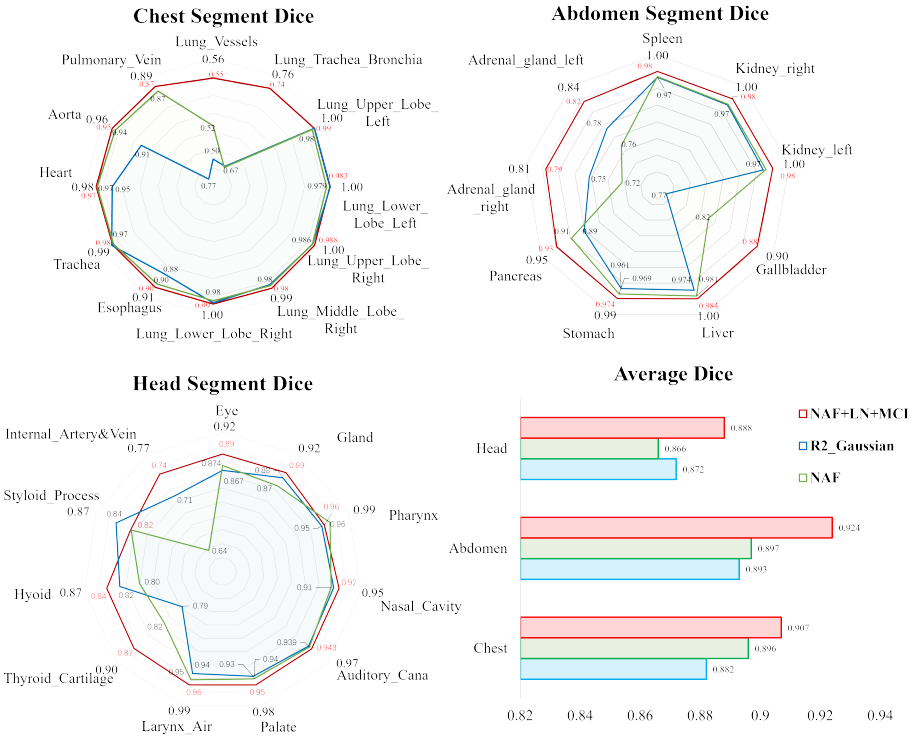

Fig. 6: The segment results of different methods on 3 datasets.

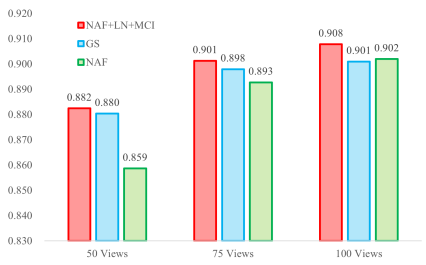

Fig. 7: SSIM score of different methods using different number of views.

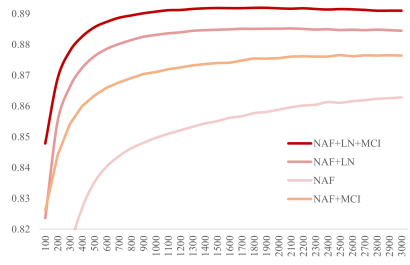

Fig. 8: Ablation study of LN and MCI on NAF.

3.3 Ablation study

To evaluate the impact of our proposed improvements, we select three images and compute the average SSIM across them. As shown in Fig. 8, each modification not only accelerates convergence but also enhances the final performance.

4 Conclusion

In conclusion, we address the local-global training mismatch in NeRF-based CBCT reconstruction, where the hash encoder's local sparse updates conflict with the neural network's global dense updates, causing misaligned features, unstable training, and slow convergence. To resolve this, we propose a Normalized Hash Encoder for feature consistency and a Mapping Consistency Initialization(MCI) strategy for stable neural network initialization. Our method, requiring minimal code changes, but significantly improves training efficiency and reconstruction quality. This work provides a robust solution for efficient and accurate CBCT reconstruction.

Acknowledgements This work was supported by the Natural Science Foundation of China (Grant 62271465), the Suzhou Basic Research Program (Grant SYG202338), and IMI BigPicture project (IMI945358)

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

References

1. Andersen, A.H., Kak, A.C.: Simultaneous algebraic reconstruction technique (sart): a superior implementation of the art algorithm. Ultrasonic imaging **6**(1), 81–94 (1984)
2. Antonelli, M., Reinke, A., Bakas, S., Farahani, K., Kopp-Schneider, A., Landman, B.A., Litjens, G., Menze, B., Ronneberger, O., Summers, R.M., et al.: The medical segmentation decathlon. Nature communications **13**(1), 4128 (2022)
3. Biguri, A., Dosanjh, M., Hancock, S., Soleimani, M.: Tigre: a matlab-gpu toolbox for cbct image reconstruction. Biomedical Physics & Engineering Express **2**(5), 055010 (2016)
4. Cai, Y., Liang, Y., Wang, J., Wang, A., Zhang, Y., Yang, X., Zhou, Z., Yuille, A.: Radiative gaussian splatting for efficient x-ray novel view synthesis. In: European Conference on Computer Vision. pp. 283–299. Springer (2024)
5. Cai, Y., Wang, J., Yuille, A., Zhou, Z., Wang, A.: Structure-aware sparse-view x-ray 3d reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11174–11183 (2024)
6. Diederik, K.: Adam: A method for stochastic optimization. (No Title) (2014)
7. Feldkamp, L.A., Davis, L.C., Kress, J.W.: Practical cone-beam algorithm. Josa a **1**(6), 612–619 (1984)
8. Gao, Z., Planche, B., Zheng, M., Chen, X., Chen, T., Wu, Z.: Ddgs-ct: Direction-disentangled gaussian splatting for realistic volume rendering. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024)
9. Guo, P., Zhao, C., Yang, D., Xu, Z., Nath, V., Tang, Y., Simon, B., Belue, M., Harmon, S., Turkbey, B., et al.: Maisi: Medical ai for synthetic imaging. arXiv preprint arXiv:2409.11169 (2024)
10. Isensee, F., Jaeger, P.F., Kohl, S.A., Petersen, J., Maier-Hein, K.H.: nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. Nature methods **18**(2), 203–211 (2021)
11. Jiao, F., Gui, Z., Li, K., Shangguang, H., Wang, Y., Liu, Y., Zhang, P.: A dual-domain cnn-based network for ct reconstruction. IEEE Access **9**, 71091–71103 (2021)

12. Li, Y., Fu, X., Li, H., Zhao, S., Jin, R., Zhou, S.K.: 3dgr-ct: Sparse-view ct reconstruction with a 3d gaussian representation. Medical Image Analysis p. 103585 (2025)
13. Lin, Y., Luo, Z., Zhao, W., Li, X.: Learning deep intensity field for extremely sparse-view cbct reconstruction. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 13–23. Springer (2023)
14. Liu, J., Bai, X.: Volumenerf: Ct volume reconstruction from a single projection view. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 743–753. Springer (2024)
15. 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: European conference on computer vision. pp. 405–421. Springer (2020)
16. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. arXiv:2201.05989 (2022)
17. NVIDIA, Vingelmann, P., Fitzek, F.H.: Cuda, release: 10.2.89 (2020), https://developer.nvidia.com/cuda-toolkit
18. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
19. Podobnik, G., Strojan, P., Peterlin, P., Ibragimov, B., Vrtovec, T.: Han-seg: The head and neck organ-at-risk ct and mr segmentation dataset. Medical physics **50**(3), 1917–1927 (2023)
20. Roth, H., Farag, A., Turkbey, E., Lu, L., Liu, J., Summers, R.: Data from pancreas-ct (version 2)[data set]. The Cancer Imaging Archive (2016)
21. Rückert, D., Wang, Y., Li, R., Idoughi, R., Heidrich, W.: Neat: Neural adaptive tomography. ACM Transactions on Graphics (TOG) **41**(4), 1–13 (2022)
22. Scarfe, W.C., Farman, A.G., Sukovic, P., et al.: Clinical applications of cone-beam computed tomography in dental practice. Journal-Canadian Dental Association **72**(1), 75 (2006)
23. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
24. 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), e230024 (2023)
25. Zang, G., Idoughi, R., Li, R., Wonka, P., Heidrich, W.: Intratomo: Self-supervised learning-based tomography via sinogram synthesis and prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1960–1970 (2021)
26. Zha, R., Lin, T.J., Cai, Y., Cao, J., Zhang, Y., Li, H.: R2-gaussian: Rectifying radiative gaussian splatting for tomographic reconstruction. arXiv preprint arXiv:2405.20693 (2024)
27. Zha, R., Zhang, Y., Li, H.: Naf: neural attenuation fields for sparse-view cbct reconstruction. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 442–452. Springer (2022)

28. Zheng, Y., Hatzell, K.B.: Ultrasparse view x-ray computed tomography for 4d imaging. ACS Applied Materials & Interfaces **15**(29), 35024–35033 (2023)