TOGS: Gaussian Splatting with Temporal Opacity Offset for Real-Time 4D DSA Rendering

Shuai Zhang, Huangxuan Zhao, Zhenghong Zhou, Guanjun Wu, Chuansheng Zheng, Xinggang Wang, *Member, IEEE*, and Wenyu Liu, *Senior Member, IEEE*

***Abstract*— Four-dimensional Digital Subtraction Angiography (4D DSA) is a medical imaging technique that provides a series of 2D images captured at different stages and angles during the process of contrast agent filling blood vessels. It plays a significant role in the diagnosis of cerebrovascular diseases. Improving the rendering quality and speed under sparse sampling is important for observing the status and location of lesions. The current methods exhibit inadequate rendering quality in sparse views and suffer from slow rendering speed. To overcome these limitations, we propose TOGS, a Gaussian splatting method with opacity offset over time, which can effectively improve the rendering quality and speed of 4D DSA. We introduce an opacity offset table for each Gaussian to model the temporal variations in the radiance of the contrast agent. By interpolating the opacity offset table, the opacity variation of the Gaussian at different time points can be determined. This enables us to render the 2D DSA image at that specific moment. Additionally, we introduced a Smooth loss term in the loss function to mitigate overfitting issues that may arise in the model when dealing with sparse view scenarios. During the training phase, we randomly prune Gaussians, thereby reducing the storage overhead of the model. The experimental results demonstrate that compared to previous methods, this model achieves state-of-the-art reconstruction quality under the same number of training views. Additionally, it enables real-time rendering while maintaining low storage overhead. The code will be publicly available.**

***Index Terms*— Gaussian Splatting, 4D DSA reconstruction, NeRF, Medical imaging, Real-time rendering**

I. Introduction

THE 4D-DSA volume can not only be viewed at any desired angle but also at any desired time during the passage of the contrast bolus through the vasculature [1]. It has been used in the diagnosis of some vascular abnormalities in clinics, such as aneurysms and AVMs/AVFs [2]. Compared to 2D and 3D DSA, 4D DSA incorporates temporal information that enables visualization of various abnormal features, such as the flow sequence and velocity of blood entering and leaving abnormal regions. This is crucial for the diagnosis of certain cerebrovascular diseases.

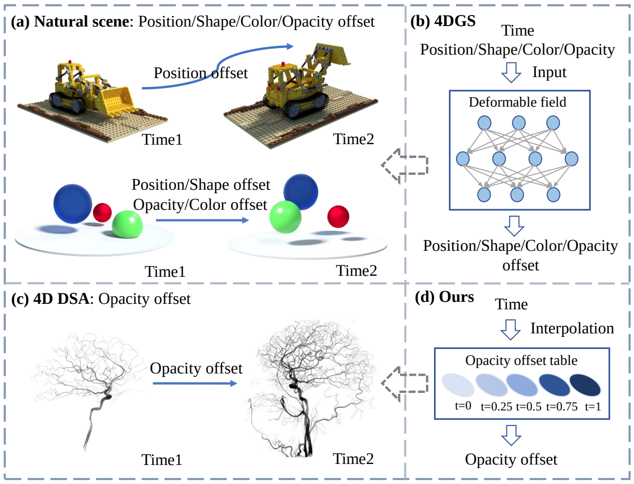

Fig. 1. Comparison of temporal variations between natural scenes and 4D DSA scenes

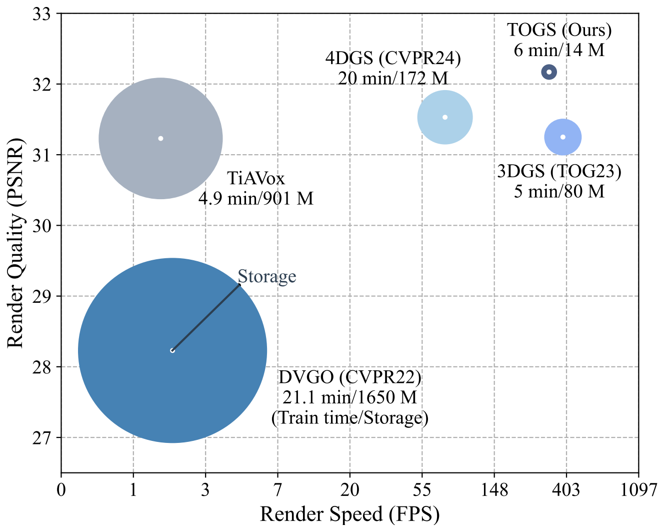

Fig. 2. Performance of different methods in 4D DSA reconstruction (30 training views).

Currently, the classical Feldkamp-Davis-Kress (FDK) re-

This work was supported by National Natural Science Foundation of China (No. 62376102). (Corresponding author: Wenyu Liu.)

Shuai Zhang, Zhenghong Zhou, Xinggang Wang, and Wenyu Liu are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, 430074, China. (e-mail: shuaizhang@hust.edu.cn; zhouzhenghong1999@gmail.com; xgwang@hust.edu.cn; liuwy@hust.edu.cn).

Guanjun Wu is with the School of Computer Science &Technology, Huazhong University of Science and Technology, Wuhan, 430074, China. (e-mail: guajuwu@hust.edu.cn).

Huangxuan Zhao and Chuansheng Zheng are with the Department of Radiology, Union Hospital, Tongji Medical College, Huazhong University of Science and Technology, Wuhan, 430022, China. (e-mail: zhao_huangxuan@sina.com; hqzcsxh@sina.com).

construction algorithm [3] is considered the gold standard for reconstructing 4D DSA images in clinical practice. However, it requires acquiring a substantial number of 2D DSA images from different perspectives. This results in the generation of a significant amount of radiation, posing a threat to the health of the patient. Therefore, achieving better 4D DSA reconstruction results with fewer views is an important research question. Moreover, real-time rendering speed plays a crucial role in enabling doctors to promptly visualize lesions and interact with them. Therefore, our research is primarily focused on developing 4D DSA reconstruction methods that can simultaneously achieve high rendering speed and superior rendering quality.

In recent years, neural radiance fields(NeRF) [4]–[7] have achieved remarkable results in tasks such as novel view synthesis and scene reconstruction. It is also widely used in the field of medical imaging [8]. As an implicit scene representation method, NeRF directly employs Multilayer Perceptrons (MLPs) to map the input position and direction information to color and density information. The resulting image is then obtained through volume rendering techniques [9]. However, due to the requirement of multiple MLP calculations for each rendered image, the original NeRF model incurs significant training and rendering time overhead. In recent studies, there have been various attempts to improve NeRF by using explicit representations instead of implicit representations [10] [11] or by employing plane decomposition techniques [12]. These approaches aim to reduce memory usage and enhance training and rendering speed.

3D Gaussian Splatting [13] utilizes 3D Gaussians to represent the scene and employs an efficient differentiable splatting technique as a replacement for volume rendering. This approach enables real-time rendering while ensuring high-quality images. In addition to the advantage of fast rendering speed, 3D Gaussian splatting (3DGS) also avoids the computational overhead of rendering blank regions in space. Traditional NeRF requires sampling a large number of points in space, including blank regions. However, in 3DGS, the Gaussians only occur in non-blank regions, thus avoiding rendering blank areas. 4D DSA images often contain a significant amount of blank regions, making Gaussian splatting a suitable method for rendering such scenes. Since 4D DSA images are dynamic, it is necessary to find an appropriate way to extend 3D Gaussian splatting to handle the dynamics of the data.

Currently, several methods extend 3D Gaussian Splatting (3DGS) to dynamic scenes, One class of methods directly extends 3D Gaussians to 4D Gaussians [14]–[16], while another class of methods models the temporal variations of Gaussians using deformable fields [17]–[20]. However, these works primarily focus on dynamic scenes in natural environments. As shown in Fig. 1, in natural scenes, objects change position, shape, opacity, and color over time, with the most significant change being the displacement of objects. In the 4D DSA reconstruction task, the blood vessels in the brain remain stationary, and the signal intensity of the contrast agent in the blood vessels changes with the flow of blood, the primary variation occurs in the opacity of blood vessels. Therefore, the emphasis in reconstructing natural scenes and 4D DSA scenes differs. The current dynamic 3DGS methods are primarily designed for natural dynamic scenes. When directly applied to the 4D DSA reconstruction problem, issues such as redundant module design, low computational efficiency, and difficulties in achieving convergence may arise.

To address the unique problem of 4D DSA reconstruction, we propose a simple and efficient method to extend 3DGS to the scene with changing opacity, specifically for modeling the diffusion process of the contrast agent within blood vessels. In this process, the blood vessels remain stationary, and only the signal intensity of the contrast agent varies. Therefore, in our approach, we only model the changes in opacity within the Gaussians. For each Gaussian, we introduce an opacity offset table to store the offset values of opacity at different times. As shown in Fig. 1 (d), the opacity of each Gaussian is obtained by querying or interpolating the opacity offset table and adding it to the original opacity value, Then get the rendered image by splatting. Our method is specifically designed for 4D DSA reconstruction. As shown in Fig. 2. It offers advantages such as low memory consumption, fast rendering speed, and high rendering quality. The entire process of 4D DSA reconstruction is depicted in Fig. 3.

In summary, our contributions can be summarized as follows:

(1) We introduce an effective TOGS framework to handle the 4D DSA reconstruction problem, in which opacity offset tables are utilized to connect 3D Gaussians' status at different times.

(2) We introduce a Smooth loss to alleviate the overfitting problem that occurs in the sparse view case of the proposed method. We randomly prune Gaussians during the training process, which not only reduces memory overhead but also helps improve the quality of rendered images.

(3) Experiments prove that under the same number of views, the proposed method can achieve state-of-the-art reconstruction results. Additionally, it provides lower memory overhead and enables rendering speeds of over 300 FPS.

II. Related Works

A. Dynamic scene reconstruction

In recent years, NeRF [4] has been widely used in the 3D reconstruction of static scenes. It implicitly represents the scene using MLPs. MLP maps the position information and perspective information of points in the 3D space into their color and density information and then obtains the image through volume rendering. The original NeRF method has high memory and time overhead. Therefore, subsequent research has focused on improving training and rendering times, as well as reducing memory consumption [21]. These include explicit representations [22] [23] and tensor decomposition [12] methods.

Currently, many works have extended NeRF for dynamic scene reconstruction, and a mainstream approach is to introduce a deformation field to model the spatial movement of points in the scene, such as Nerfies [24], D-NeRF [25], HyperNeRF [26] and TiNeuVox [27] etc. In addition, there are some methods [28]–[30] that use feature planes to capture 4D

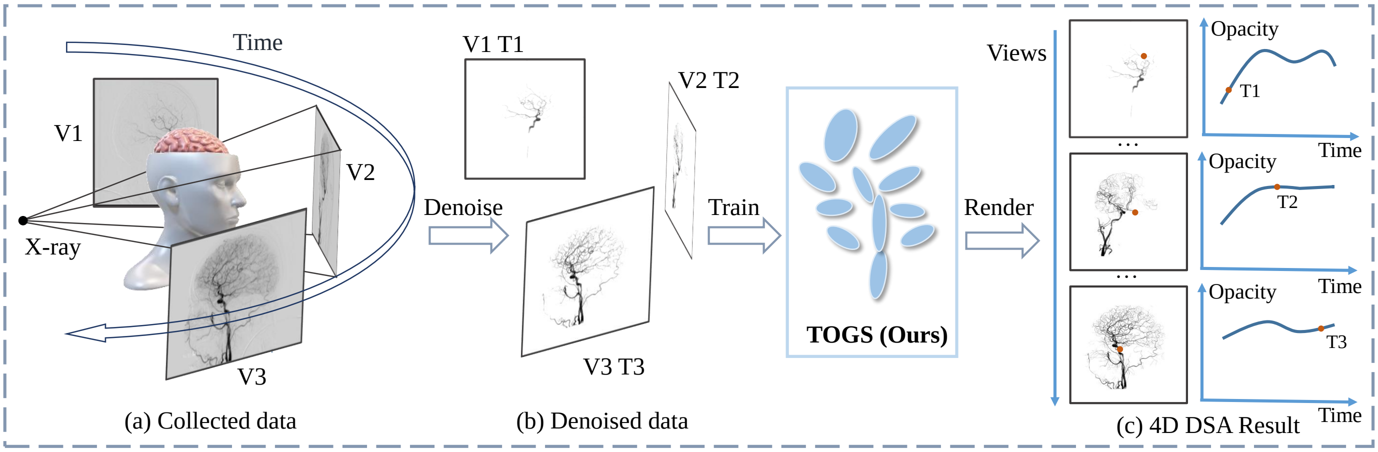

Fig. 3. An overview of the proposed TOGS for 4D DSA reconstruction. DSA data (a) is collected from different viewpoints and time points. The image is then denoised to retain only the blood vessel region. The denoised data (b) is subsequently used to train the TOGS model. The trained TOGS model is capable of rendering DSA images (c) from any desired time point and viewpoint.

features to achieve high-quality dynamic new views synthesis and faster convergence speed. Although these methods achieve fast training speed, they are still challenging in real-time rendering of dynamic scenes.

The recent 3D Gaussian Splatting (3DGS) [13] adopts 3D Gaussian to represent the scene and utilizes efficient splatting techniques for image rendering. As a result, it significantly improves rendering speed and has gained widespread attention. There are multiple approaches to extend 3DGS to dynamic scenes [14]–[20]. Some methods directly allocate 3D Gaussians for each time point or introduce time attributes to Gaussians to represent dynamic scenes [14]–[16]. For example, Dynamic 3DGS [16] models dynamic scenes by tracking the position and variance of each Gaussian function at each timestep. This method has difficulties in accurately modeling monocular dynamic scenes, and it explicitly stores all the information of each Gaussian at each moment, leading to substantial storage overhead. Another popular extension method involves utilizing a deformation field to simulate the variations in 3D Gaussian attributes, such as position, shape, and opacity [17]–[20]. For example, In 4DGS [18], multi-resolution neural voxels and a decoder MLP are used as the deformation field. It utilizes HexPlane to connect different adjacent Gaussian fields, aiming to predict more precise motion and shape deformations. The methods designed for natural dynamic scenes that model the displacement and shape changes of Gaussians may contain redundancy when applied to the 4D DSA reconstruction problem. Additionally, using MLP or other deformation fields to model opacity transformations can introduce additional computational and memory overhead. Hence, there is a requirement to develop an efficient Gaussian splatting method that is specifically designed for the 4D DSA reconstruction problem.

B. 4D DSA Reconstruction

Compared to 3D DSA reconstruction, 4D DSA provides temporal information that aids in visualizing abnormal features of certain diseases. For instance, it enables the visualization of internal characteristics of AVM lesions or the flow sequence of blood entering and leaving abnormal regions. Recent research [31] [32] suggests that the temporal information contained in 4D-DSA enables the quantification of blood flow velocity. When combined with accurate vascular measurements, it allows for the quantification of blood flow. Specifically, in 4D-DSA, each point has a time-density curve (TDC). By utilizing the time-density curves between two points, along with the distance between them and the cross-sectional area of the vessel, blood flow can be quantified.

Among the current reconstruction methods for 4D DSA images, the gold standard used in clinics is the classical Feldkamp-Davis-Kress (FDK) reconstruction algorithm [3]. However, this algorithm requires a large number of 2D DSA images, resulting in significant exposure to radiation and potential harm to the patient. The emergence of deep learning methods has promoted research on CT/DSA image reconstruction [33]–[36]. These methods are mainly aimed at static CT reconstruction and do not consider the flow of contrast agents.

In recent years, NeRF has been increasingly applied to medical image reconstruction tasks, including CT reconstruction [37] [38] and deformable tissue reconstruction [39] [40]. In EndoNeRF [39], it follows the modeling in D-NeRF [25] and represents deformable surgical scenes as a canonical neural radiance field along with a time-dependent neural displacement field. In LerPlane [40], it treats surgical procedures as 4D volumes and factorizes them into explicit 2D planes of static and dynamic fields, leading to a compact memory footprint and significantly accelerated optimization. Specifically addressing the 4D DSA reconstruction problem, TiAVox [41] introduces a 4D attenuation voxel grid to reflect the attenuation characteristics of 4D DSA from both spatial and temporal dimensions. TiAVox has achieved shorter training times. However, there is still significant room for improvement in terms of the rendering speed of the model.

3DGS achieves real-time rendering while maintaining high rendering quality, which greatly enhances visualization. Paper [42] applied 3DGS to sparse CT reconstruction and achieved good performance. Therefore, it is necessary to introduce Gaussian splatting techniques into the 4D DSA reconstruction problem. Existing 4D Gaussian splatting methods mainly focus on dynamic scenes in the natural world, which differ from the dynamic processes in 4D DSA. Hence, this paper designs a Gaussian splatting algorithm specifically for the 4D DSA reconstruction problem. We directly employ an opacity offset table to model the opacity offset of Gaussians over time. Compared to current 4DGS methods, this approach offers advantages such as lower computational requirements, faster rendering speed, and smaller memory overhead.

III. Preliminary

A. 3D Gaussian Splatting

3DGS [13] represents the scene through a large number of 3D Gaussians, and each Gaussian is characterized by a covariance matrix $\Sigma \in \mathbb{R}^{3\times 3}$ and mean $\mu \in \mathbb{R}^3$:

$$G(X) = e^{-\frac{1}{2}(x-\mu^T)\Sigma^{-1}(x-\mu)}, \tag{1}$$

where the covariance matrix Σ is divided into a quaternion matrix R representing rotation and a matrix S representing scaling:

$$\Sigma = RSS^T R^T. \tag{2}$$

Paper [43] demonstrates how to project a 3D Gaussian into 2D. Given the viewing transformation W and the Jacobian of the affine approximation of the projective transformation J, the covariance matrix Σ' in camera coordinates can be obtained by:

$$\Sigma' = JW\Sigma W^T J^T. \tag{3}$$

In summary, each Gaussian $G(x, r, s, \alpha, c)$ is represented by the following attributes: position $x \in \mathbb{R}^3$, opacity $\alpha \in \mathbb{R}$, rotation factor $r \in \mathbb{R}^4$, and scaling factor $s \in \mathbb{R}^3$, and the color represented by the spherical harmonics c. To render a 2D image, 3DGS sorts all the Gaussians that make up a pixel and determines the color of each pixel by:

$$C = \sum_{i=1}^{n} c_i \sigma_i \prod_{j=1}^{i-1} (1 - \sigma_j), \tag{4}$$

where the c_i is the color calculated from the SH coefficient of the i-th Gaussian function. The σ_i is obtained by multiplying the opacity α with the covariance Σ' of the two-dimensional Gaussian function:

$$\sigma_i = \alpha_i \times e^{-\frac{1}{2}(x'-\mu_i')^T \Sigma^{-1}(x'-\mu_i')}, \tag{5}$$

where x and μ_i are coordinates in the projected space.

3DGS uses an adaptive density control strategy to control the number of Gaussians. For the under-reconstructed area, the Gaussian is copied and moved along the direction of the positional gradient. On the other hand, the large Gaussian distribution in the high variance area is divided into small Gaussian distributions. In addition, it deletes Gaussian whose opacity is less than the threshold every certain iteration.

IV. Methods

A. The overview of the proposed model

The pipeline of the proposed model is illustrated in Fig. 4. Generate a set of point clouds through random initialization, and use the point cloud to generate a set of 3D Gaussians $G(x, r, s, \alpha, c)$ defined by a center position x, opacity α, and 3D covariance matrix Σ obtained from quaternion r and scaling s. In order to model the signal intensity changes of contrast agents in 4D DSA, we introduce an opacity offset table T_α to each Gaussian, resulting in $G(x, r, s, \alpha, T_\alpha, c)$. After the current time t is input, the opacity offset value $\Delta\alpha_t$ is obtained by interpolating the opacity offset table. Then $G(x, r, s, \alpha + \Delta\alpha_t, c)$ is sent to the efficient differential Gaussian rasterization pipeline to render the image. By calculating the loss and performing backpropagation, we apply adaptive density control to the Gaussians.

B. Opacity offset table

In the 4D DSA reconstruction problem, the blood vessels remain stationary, while the signal intensity of the contrast agent inside the vessels varies due to blood flow. Therefore, the proposed method specifically focuses on modeling the temporal changes in Gaussian opacity. We directly introduce an opacity offset table T_α for each Gaussian to store the opacity offset values at different time points, so as to reduce additional calculation and space overhead as much as possible. During the rendering process, the opacity of the Gaussian at a given moment is calculated by adding the initial opacity value to the opacity offset value obtained through linear interpolation from the opacity offset table T_α.

$$\alpha_t = \alpha' + \Delta\alpha_t, where\ \Delta\alpha_t = interp(T_\alpha), \tag{6}$$

where α' is the initial opacity of the Gaussian, and $\Delta\alpha_t$ is the opacity offset value of the Gaussian at time t. In this paper, we use an opacity offset table of length 5 to store the opacity offset values for Gaussian at time t equal to 0, 0.25, 0.5, 0.75, and 1, respectively.

C. The initialization and density control of Gaussians

1) Random initialization: Generating a large number of effective initial point clouds using Structure from Motion (SfM) [44] can be challenging in the case of 4D DSA, as it is a dynamic imaging technique with only one image available for each moment and viewpoint. Therefore, this paper randomly generates 100,000 point clouds as initialization.

2) Pruning Gaussians based on maximum opacity: One of the key reasons why 3D GS can effectively represent the scene is the adoption of Gaussian adaptive density control. This strategy sets an opacity threshold and removes Gaussians with opacity below the threshold every certain number of iterations. During the 4D DSA imaging process, many points initially have low opacity, and their opacity gradually increases over time. In this paper, we prune the Gaussian when the maximum opacity of the Gaussian is below the threshold value U, as shown in the following formula:

$$\alpha' + max(\Delta\alpha_t) < U. \tag{7}$$

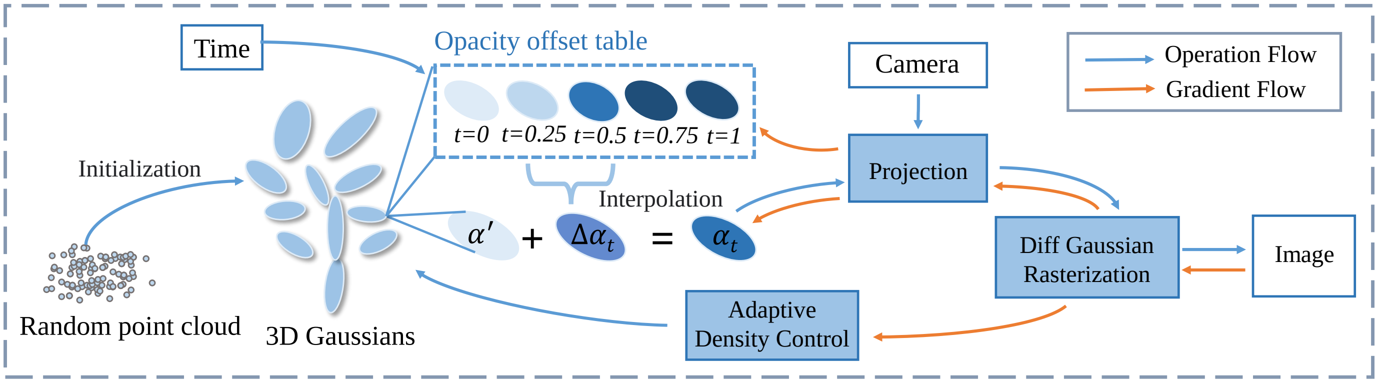

Fig. 4. Overview of our pipeline. First, we randomly generate point clouds and initialize them as 3D Gaussians. We introduce an opacity offset table for each Gaussian. When different time values $\boldsymbol{t}$ are input, interpolate the opacity offset table to obtain the opacity offset value $\boldsymbol{\Delta\alpha_t}$. Add it to the original opacity $\boldsymbol{\alpha'}$ to obtain the opacity $\boldsymbol{\alpha_t}$ at that moment. Then, Gaussians are input into the efficient differential Gaussian rasterization pipeline to render the image. Finally, by computing the loss and backpropagating gradients, we adaptively control the density of the Gaussians.

3) Randomly pruning Gaussians: Due to the utilization of random initialization and the inherent characteristics of 3DGS, the model is prone to generating an excessive number of Gaussians when fitting scenes under sparse viewing angles, which can result in overfitting issues. During the training phase of the model, Gaussian splitting, duplication, and pruning are executed between the 1000th and 20000th iterations. In this particular stage, for every 200 iterations, an extra 8% of the Gaussians are randomly pruned. This strategy allows us to eliminate redundant Gaussians, decrease the memory usage of the model, and enhance the quality of the rendered output simultaneously.

D. Optimization

The loss function of the proposed model consists of three components: Smooth loss, reconstruction loss, and Structural Similarity Index Measure (SSIM) [45] loss.

1) Smooth loss: We introduced an opacity offset table to simulate the signal intensity variation of contrast agent flow. However, during the experiments, we observed the presence of overfitting issues. To address this, we proposed a Smooth loss to constrain the optimization of the opacity offset table:

$$L_{\text{Smooth}} = \frac{1}{N}\sum_{k=0}^{N}\sum_{t=1}^{L}\left|\Delta\alpha_t^k - \Delta\alpha_{t-1}^k\right|, \tag{8}$$

where $\Delta\alpha_t^k$ represents the value indexed as t in the opacity offset table of the k-th Gaussians, N represents the total number of points, and L represents the length of opacity offset table.

By introducing the Smooth loss, we reduce the differences between consecutive opacity offsets in the opacity offset table. This promotes a smoother temporal variation of opacity for the Gaussians, which helps alleviate overfitting issues.

2) Reconstruction loss: Similar to 3DGS [13], we introduce the reconstruction loss in the model:

$$L_{\text{recon}} = \|\hat{I} - I\|_1, \tag{9}$$

where $\hat{I}$ represents the target image, and I represents the rendered image.

3) SSIM loss: We introduce SSIM loss to improve the quality of generated images:

$$L_{SSIM} = 1 - \text{SSIM}(\hat{I}, I). \tag{10}$$

Therefore, the final loss function is:

$$L_{\text{total}} = L_{\text{recon}} + \lambda_{\text{SSIM}} L_{\text{SSIM}} + \lambda_{\text{Smooth}} L_{\text{Smooth}} , \tag{11}$$

where λ_{SSIM} represents the weight of SSIM loss, and λ_{Smooth} represents the weight of Smooth loss.

V. Experiment

A. Implementation Details

We train our model using the Adam [46] optimizer, with a total of 30,000 iterations for each experiment. The learning rates for position, opacity, size, and rotation parameters of the Gaussian are set as follows: 1.6×10^{-4}, 0.025, 0.005, and 0.001, respectively. The hyperparameters of the loss function are tuned based on the number of training viewpoints. In the case of 30 views, the values of the hyperparameters are as follows: λ_{SSIM} is 0.1, λ_{Smooth} is 0.0043, and the opacity threshold T is set to 0.018. The proposed method is implemented in Python, using the PyTorch deep learning framework. All experiments are conducted on a single NVIDIA RTX 3090 GPU with 24GB memory.

B. Datasets

We use data provided by TiAVox [41], collected from 8 patients in Wuhan Union Hospital. For each patient's data, a total of 133 2D projection images were captured. Each image was acquired at different time points and perspectives. We normalize the time corresponding to the images from 0 to 1. The resolution of the image is 1024×1024 pixels.

Clinical DSA images often contain various sources of noise, such as bone signals, imaging artifacts, and other interferences. These factors significantly affect the quality of synthesized new views. As 4D DSA reconstruction focuses specifically on the vascular region, a Unet is used in TiAVox [41] to segment the blood vessel area and generate mask images of the blood vessel. By applying mask images, we obtain clear images containing only the blood vessel area, as shown in Fig. 3 (b).

C. Evaluation metrics

In this paper, we use Peak Signal to Noise Ratio (PSNR), SSIM, and Learned Perceptual Image Patch Similarity (LPIPS) [47] as the three metrics to evaluate the quality of the generated images by the model in the task of synthesizing new views. We use the VGG network [48] as the feature extraction backbone to calculate LPIPS.

D. Comparison with Existing Methods

We conducted experiments on novel view synthesis when the number of training views was 10, 20, 30, 40, 50, and 60, respectively. We selected DVGO [23], TiAVox [41], 3DGS [13], and 4DGS [18] as the four methods for comparison in experiments. Among them, DirectVoxGo and TiAVox use voxel grids to represent scenes, and 3DGS and 4DGS use 3D Gaussians to represent scenes. We found that modeling position offset in 4D DSA can potentially hinder model convergence. Therefore, in our comparative experiment, we removed the position offset of Gaussians modeled in 4DGS. For detailed experimental information, please refer to the V-E.1 section of the paper.

The results of the comparative experiment are shown in Fig. 5 and Table I. From Fig. 5, it is evident that the reconstruction results of DVGO exhibit noticeable noise and fail to effectively capture the intricate details of the blood vessels. TiAvox is capable of recovering the shape of the vessels, however, the opacity of the vessels appears less realistic, and the vessel details are relatively blurry. As shown in Fig. 5 (i) and (j), in regions with a significant overlap of blood vessels, the results obtained from 3DGS and 4DGS appear cluttered, and the background is not adequately clean. In 4D DSA reconstruction, the opacity of many Gaussians is initially low and gradually increases over time. 4DGS directly clipping Gaussians based on their initial opacity consequently limits its reconstruction results. In contrast, our proposed method accurately recovers the details of the blood vessels while generating more realistic results. From Table I, it is evident that our proposed method outperforms existing approaches in terms of evaluation metrics, particularly in the case of sparse viewpoint quantities (10 views). Our method produces higher-quality novel viewpoint images.

We also conducted a comparison of rendering speed, training time, and storage size among different methods, as shown in Table II. The Gaussian Splatting-based method exhibits significantly faster rendering speed compared to the voxel grid-based method. Our method extends 3DGS to dynamic scenes in 4D DSA while maintaining a high rendering speed. Regarding training time, our model can complete the training process within 6 minutes, maintaining a high training speed. In terms of storage overhead, the proposed model achieves the minimum storage cost. This is because the proposed model can effectively represent the scene using only a small number of Gaussians, without relying on any additional neural networks.

E. Ablation study

1) Modeling position offset experiment: To verify whether it is suitable to model the displacement of objects in the 4D DSA reconstruction problem, we conducted experiments on two dynamic 3DGS methods, Deformable 3DGS [17], and 4DGS, that both use deformable fields to model object displacements. In Deformable 3DGS, we introduced a coefficient w for position displacement δx. When w is set to 0, the model generates a large number of point clouds, making it challenging to discern the shape of the blood vessels clearly from these point clouds. The model can render 2D DSA images normally. When $0 < w <= 0.3$, the point cloud shape is abnormal, but 2D DSA images can be rendered. When $w > 0.3$, most of the point cloud is clipped, leaving only very few points, and 2D DSA images cannot be rendered normally. In 4DGS, when we remove the position displacement δx of the point cloud, the point cloud shape can be well generated and the DSA image can be rendered at the same time. When the position offset is introduced, the shape of the blood vessels is partially missing, and the image cannot be rendered normally. This experiment confirms that modeling positional offset in 4D DSA reconstruction may prevent the model from converging properly, thereby affecting the rendering of images. Therefore, when comparing to 4DGS, we removed the position offset in 4DGS.

2) Ablation study for Opacity offset table and Randomly pruning Gaussians: We conducted ablation experiments to verify the effectiveness of the proposed model. First, we verified the effectiveness of the introduced opacity offset table and random pruning strategy, as shown in Table III. After removing the opacity offset table, the model is still capable of fitting new viewpoint images, but the quality of the rendered images significantly deteriorates. Simultaneously, it can be observed from the table that randomly pruning a certain proportion of Gaussians can also enhance the quality of rendered images. Therefore, we conducted further research on the impact of randomly pruning Gaussians at different proportions on memory usage and the quality of the generated images. The experimental results are shown in Fig. 6. We compared the experimental results with pruning proportions of 4%, 8%, 10%, and 12%. Taking 8% as an example, we observed that compared to the results without random pruning strategy, the number of Gaussians decreased from 120,000 to 40,000, the model memory reduced from 33M to 11M, and the PSNR of the generated images improved from 31.69 to 31.81. This phenomenon suggests that in Gaussian splatting, the random initialization and Gaussian splitting duplication strategies may lead to the generation of excessive Gaussians to fit the scene, resulting in overfitting issues. Randomly pruning a certain proportion of Gaussians can help alleviate overfitting problems and reduce memory overhead.

3) Ablation study for Loss function: In this ablation experiment, we validated the effectiveness of each loss term. We compared the impact of including the Smooth loss versus excluding it on the results. The results are shown in Table IV and Fig. 7. It can be observed that when there is a significant number of viewpoints, the improvement from incorporating the Smooth loss term is not substantial. However, when the number of viewpoints is limited, such as 10 or 20, the inclusion of the Smooth loss term effectively enhances the reconstruction metrics. This suggests that the Smooth loss term can effectively mitigate overfitting issues that arise when

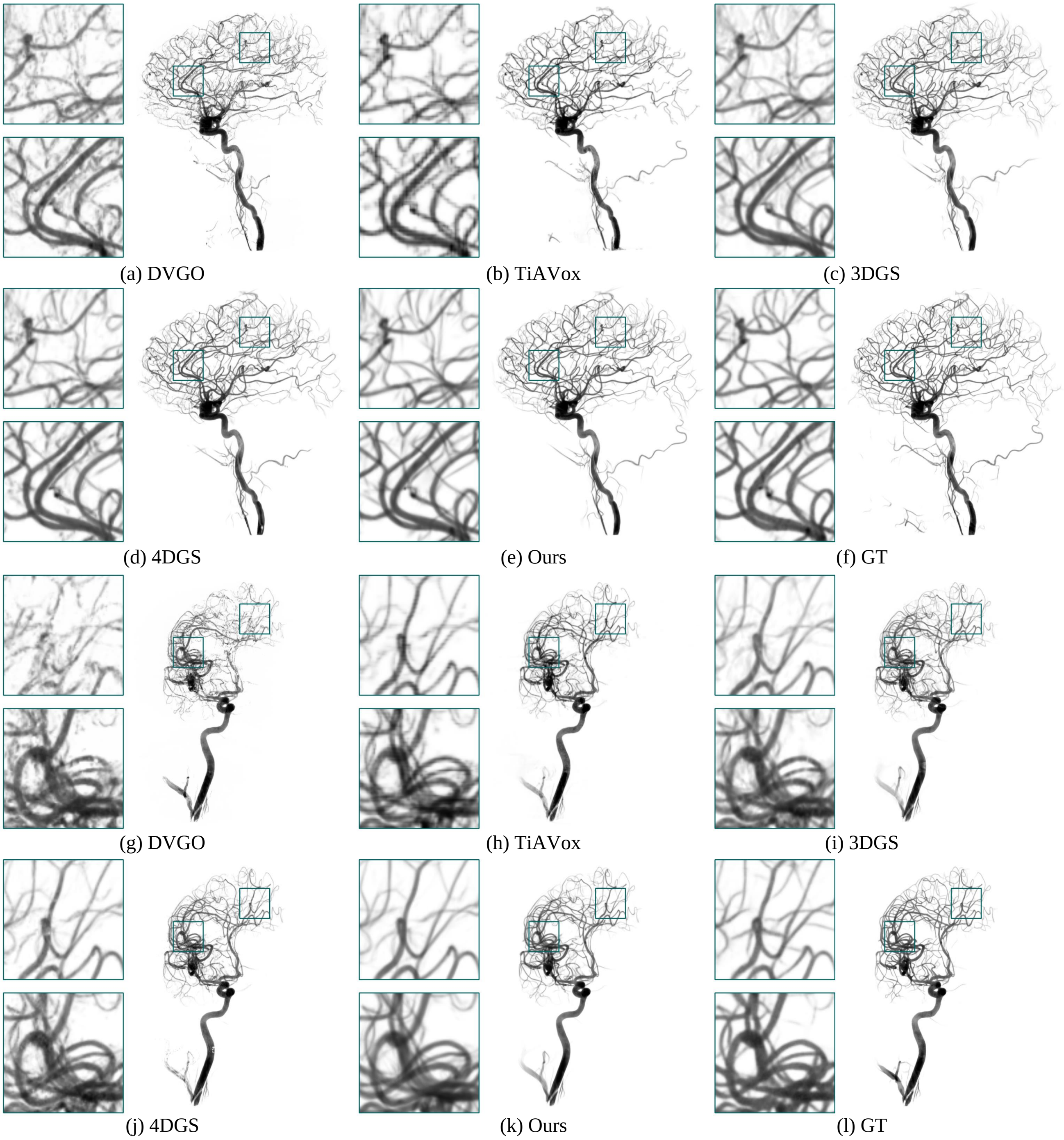
(a) DVGO (b) TiAVox (c) 3DGS (d) 4DGS (e) Ours (f) GT (g) DVGO (h) TiAVox (i) 3DGS (j) 4DGS (k) Ours (l) GT

Fig. 5. Visualization results of each model in the comparative experiment under 30 training viewpoints.

the number of views is restricted. From Fig. 7, it is also evident that the introduction of the Smooth loss in the model is beneficial for fitting the scene. Furthermore, we removed the SSIM loss term and noticed a decrease in the evaluation metrics. From Fig. 7, it can be observed that without the inclusion of SSIM loss, the model fails to capture the finer details of blood vessels accurately.

4) Ablation study for pruning strategies: We conducted ablation experiments on different pruning strategies, where the Gaussians were pruned based on their initial opacity, average opacity, or maximum opacity. Specifically, when the opacity of a Gaussian drops below a certain threshold, it is pruned. The experimental results are shown in Table V. From the table, it is evident that directly pruning Gaussians based on their initial opacity leads to subpar reconstruction performance. However, when using the maximum opacity and average opacity of Gaussians with an appropriate threshold, both methods yield higher reconstruction quality. This further highlights the advantages of our approach over the deformable field-based method. With our method, we can easily obtain the opacity of Gaussians at a specific time, allowing us to control the density of Gaussians accordingly.

5) Ablation study for Temporal resolutions: We also investigated the influence of the length of the opacity offset table

TABLE I
NOVEL VIEW SYNTHESIS RESULT (8 PATIENTS)

VIEWS	10			20			30		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
DirectVoxGo [23]	24.85	0.905	0.115	26.86	0.917	0.108	28.23	0.933	0.098
TiAVox [41]	28.57	0.939	0.091	30.46	0.951	0.077	31.23	0.959	0.065
3DGS [13]	27.23	0.924	0.095	29.82	0.947	0.063	31.25	0.961	0.050
4DGS [18]	27.36	0.927	0.089	30.64	0.953	0.058	31.53	0.962	0.050
TOGS (Ours)	**29.74**	**0.947**	**0.063**	**31.32**	**0.957**	**0.052**	**32.17**	**0.968**	**0.041**
VIEWS	40			50			60		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
DirectVoxGo [23]	29.18	0.940	0.090	29.82	0.942	0.086	30.75	0.951	0.077
TiAVox [41]	31.51	0.962	0.063	31.94	0.962	0.065	32.29	0.965	0.062
3DGS [13]	31.87	0.965	0.048	32.59	0.968	0.040	33.18	0.972	0.036
4DGS [18]	31.93	0.965	0.049	32.71	0.970	0.039	33.22	0.974	0.034
TOGS (Ours)	**32.59**	**0.970**	**0.039**	**33.24**	**0.972**	**0.036**	**33.82**	**0.975**	**0.033**

TABLE II
COMPARISON OF MODEL SPEED AND STORAGE OCCUPANCY (30 TRAINING VIEWS)

Method	Render speed↑ (FPS)	Train time↓ (minute)	Storage↓ (MB)
DirectVoxGo [23]	1.72	21.1	1650
TiAVox [41]	1.46	**4.9**	901
3DGS [13]	**384**	5.0	80
4DGS [18]	75	20.0	172
TOGS (Ours)	317	6.0	**14**

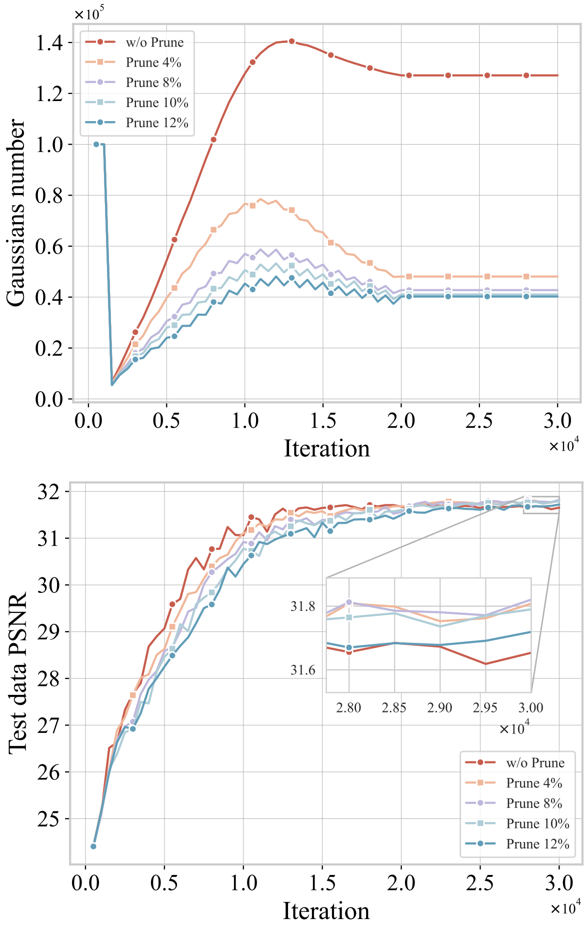

Fig. 6. The impact of randomly pruning Gaussians at different proportions on the number of Gaussians and the quality of generated images (30 training views).

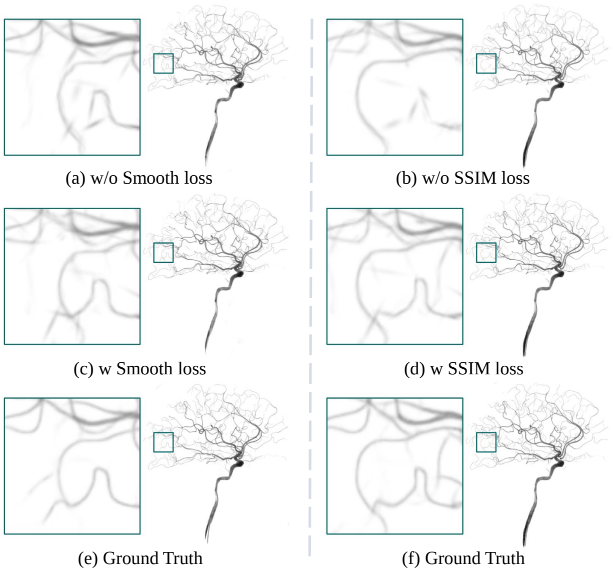
(a) w/o Smooth loss
(b) w/o SSIM loss
(c) w Smooth loss
(d) w SSIM loss
(e) Ground Truth
(f) Ground Truth

Fig. 7. Ablation study of loss function (30 training views).

on the reconstruction results, and the experimental results are shown in Table VI. From the table, it can be observed that the length of the opacity offset table should be chosen appropriately. When there are a larger number of training views, a longer opacity offset table can better capture the temporal variations of opacity. However, when there are fewer training views, using a longer opacity offset table may result in severe overfitting and a decline in reconstruction performance. Therefore, in this paper, we opted for a length of 5 for the opacity offset table.

VI. LIMITATIONS AND FUTURE WORK

Although the proposed method achieves good results under the 4D DSA new perspective synthesis task, there are still some limitations that need to be addressed in future work. The 3DGS-based methods are prone to overfitting issues in sparse-view scenarios. In this paper, We solve this problem by introducing a Smooth loss term in the loss function and randomly pruning some Gaussians. However, in extremely sparse views, such as only four training views, the model still suffers from severe overfitting issues.

TABLE III
ABLATION STUDY FOR OPACITY OFFSET TABLE AND RANDOMLY PRUNING GAUSSIANS

VIEWS	10			30			50		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
w/o opacity offset table	27.93	0.930	0.076	30.95	0.961	0.050	31.71	0.965	0.042
w/o randomly pruning Gaussians	28.56	0.933	0.070	31.63	0.963	0.045	32.24	0.964	0.041
all	**28.68**	**0.935**	**0.070**	**31.81**	**0.965**	**0.043**	**32.55**	**0.969**	**0.037**

TABLE IV
ABLATION STUDY FOR LOSS FUNCTION

VIEWS	10			30			50		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
w/o Smooth loss	27.79	0.924	0.077	31.62	0.965	0.042	32.54	0.968	0.037
w/o SSIM loss	28.58	0.934	0.071	31.59	0.963	0.047	32.26	0.966	0.042
all	**28.68**	**0.935**	**0.070**	**31.81**	**0.965**	**0.043**	**32.55**	**0.969**	**0.037**

TABLE V
ABLATION STUDY FOR PRUNING STRATEGY (30 VIEWS)

Threshold T	0.005			0.01			0.015			0.02		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Initial opacity	29.03	0.944	0.071	28.96	0.943	0.072	28.90	0.943	0.072	28.93	0.943	0.070
Mean opacity	31.75	**0.964**	**0.043**	**31.80**	**0.965**	**0.043**	31.72	0.965	0.044	31.65	0.964	0.045
Max opacity	**31.75**	0.963	0.044	31.80	0.965	0.043	**31.80**	**0.965**	**0.043**	**31.81**	**0.965**	**0.043**

TABLE VI
THE EXPERIMENTAL RESULTS AT DIFFERENT TEMPORAL RESOLUTIONS.

VIEWS	Temporal resolution	PSNR↑	SSIM↑	LPIPS↓
10	3	28.61	0.934	0.071
	5	**28.64**	**0.934**	**0.069**
	7	28.49	0.933	0.070
	10	28.43	0.929	0.072
30	3	31.67	0.964	0.044
	5	31.81	0.965	0.043
	7	31.89	0.966	0.042
	10	**32.05**	**0.967**	**0.041**

Rendering images at different time frames from the same viewpoint remains a challenging problem. Due to the limitation of having only one image per time frame for a given viewpoint in the dataset, there is a lack of supervision from other time frames. As a result, rendering images from other time frames can suffer from issues such as blurring and ghosting. In the future, there is a possibility of incorporating depth prior information [49], adding extra supervision [50], and exploring other approaches to address this issue.

In addition to addressing the aforementioned challenges, future work should further explore efficient 3D/4D representation methods specifically tailored to the unique field of medical imaging. This paper has employed a straightforward approach of using explicit opacity offset tables to model the complex signal intensity variations of contrast agents. In the future, it is crucial to explore more efficient methods for modeling the intensity variations of contrast agents. Additionally, the current Gaussian Splatting method still faces challenges such as overfitting and difficulties in surface extraction [51]. Future research should focus on refining the Gaussian Splatting method or exploring more efficient representation and rendering techniques.

VII. CONCLUSION

In this paper, to achieve high-quality and real-time rendering in 4D DSA reconstruction under sparse views, we propose TOGS. We introduced opacity offset tables in 3DGS to model the radiation signal intensity changes of contrast agents in 4D DSA. By querying the opacity offset table, we obtain the opacity of the Gaussian at a given moment. Afterward, the image at that moment can be rendered using splatting. We introduced a Smoothing loss and implemented random pruning to control the density of Gaussians. By implementing these two strategies, we successfully alleviated the issue of model overfitting and improved the quality of rendered images. The proposed method effectively enhances the quality of new viewpoint synthesis in 4D DSA. Moreover, it has the advantages of real-time rendering and low memory overhead. In the future, we will further explore more efficient representations and rendering methods for 4D DSA reconstruction.

REFERENCES

[1] K. Ruedinger, S. Schafer, M. Speidel, and C. Strother, “4d-dsa: development and current neurovascular applications,” *American Journal of Neuroradiology*, vol. 42, no. 2, pp. 214–220, 2021.

[2] S. Lang, P. Gölitz, T. Struffert, J. Rösch, K. Rössler, M. Kowarschik, C. Strother, and A. Doerfler, “4d dsa for dynamic visualization of cerebral vasculature: a single-center experience in 26 cases,” *American Journal of Neuroradiology*, vol. 38, no. 6, pp. 1169–1176, 2017.

[3] L. A. Feldkamp, L. C. Davis, and J. W. Kress, “Practical cone-beam algorithm,” *Josa a*, vol. 1, no. 6, pp. 612–619, 1984.

[4] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[5] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, “Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5855–5864, 2021.

[6] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, “Nerf++: Analyzing and improving neural radiance fields,” *arXiv preprint arXiv:2010.07492*, 2020.

[7] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Zip-nerf: Anti-aliased grid-based neural radiance fields," *arXiv preprint arXiv:2304.06706*, 2023.

[8] A. Molaei, A. Aminimehr, A. Tavakoli, A. Kazerouni, B. Azad, R. Azad, and D. Merhof, "Implicit neural representation in medical imaging: A comparative survey," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 2381–2391, October 2023.

[9] R. A. Drebin, L. Carpenter, and P. Hanrahan, "Volume rendering," *ACM Siggraph Computer Graphics*, vol. 22, no. 4, pp. 65–74, 1988.

[10] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–15, 2022.

[11] Q. Xu, Z. Xu, J. Philip, S. Bi, Z. Shu, K. Sunkavalli, and U. Neumann, "Point-nerf: Point-based neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5438–5448, 2022.

[12] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," in *European Conference on Computer Vision*, pp. 333–350, Springer, 2022.

[13] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics (ToG)*, vol. 42, no. 4, pp. 1–14, 2023.

[14] Z. Yang, H. Yang, Z. Pan, X. Zhu, and L. Zhang, "Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting," *arXiv preprint arXiv:2310.10642*, 2023.

[15] Z. Li, Z. Chen, Z. Li, and Y. Xu, "Spacetime gaussian feature splatting for real-time dynamic view synthesis," *arXiv preprint arXiv:2312.16812*, 2023.

[16] J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan, "Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis," *arXiv preprint arXiv:2308.09713*, 2023.

[17] Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, and X. Jin, "Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction," *arXiv preprint arXiv:2309.13101*, 2023.

[18] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang, "4d gaussian splatting for real-time dynamic scene rendering," *arXiv preprint arXiv:2310.08528*, 2023.

[19] Y. Liang, N. Khan, Z. Li, T. Nguyen-Phuoc, D. Lanman, J. Tompkin, and L. Xiao, "Gaufre: Gaussian deformation fields for real-time dynamic novel view synthesis," *arXiv preprint arXiv:2312.11458*, 2023.

[20] Y. Lin, Z. Dai, S. Zhu, and Y. Yao, "Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle," *arXiv preprint arXiv:2312.03431*, 2023.

[21] K. Gao, Y. Gao, H. He, D. Lu, L. Xu, and J. Li, "Nerf: Neural radiance field in 3d vision, a comprehensive review," *arXiv preprint arXiv:2210.00379*, 2022.

[22] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5501–5510, 2022.

[23] C. Sun, M. Sun, and H.-T. Chen, "Direct voxel grid optimization: Superfast convergence for radiance fields reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5459–5469, 2022.

[24] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, "Nerfies: Deformable neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5865–5874, 2021.

[25] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10318–10327, 2021.

[26] K. Park, U. Sinha, P. Hedman, J. T. Barron, S. Bouaziz, D. B. Goldman, R. Martin-Brualla, and S. M. Seitz, "Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields," *arXiv preprint arXiv:2106.13228*, 2021.

[27] J. Fang, T. Yi, X. Wang, L. Xie, X. Zhang, W. Liu, M. Nießner, and Q. Tian, "Fast dynamic radiance fields with time-aware neural voxels," in *SIGGRAPH Asia 2022 Conference Papers*, pp. 1–9, 2022.

[28] A. Cao and J. Johnson, "Hexplane: A fast representation for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 130–141, 2023.

[29] S. Fridovich-Keil, G. Meanti, F. R. Warburg, B. Recht, and A. Kanazawa, "K-planes: Explicit radiance fields in space, time, and appearance," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12479–12488, 2023.

[30] R. Shao, Z. Zheng, H. Tu, B. Liu, H. Zhang, and Y. Liu, "Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16632–16642, 2023.

[31] K. Ruedinger, E. Harvey, S. Schafer, M. Speidel, and C. Strother, "Optimizing the quality of 4d-dsa temporal information," *American Journal of Neuroradiology*, vol. 40, no. 12, pp. 2124–2129, 2019.

[32] K.-K. Chen and C.-J. Lin, "Estimating pulsatile flow velocity using four-dimensional digital subtraction angiography," in *2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 1–4, IEEE, 2023.

[33] Z. Zhang, X. Liang, X. Dong, Y. Xie, and G. Cao, "A sparse-view ct reconstruction method based on combination of densenet and deconvolution," *IEEE transactions on medical imaging*, vol. 37, no. 6, pp. 1407–1417, 2018.

[34] Y. Han and J. C. Ye, "Framing u-net via deep convolutional framelets: Application to sparse-view ct," *IEEE transactions on medical imaging*, vol. 37, no. 6, pp. 1418–1429, 2018.

[35] H. Zhao, Z. Zhou, F. Wu, D. Xiang, H. Zhao, W. Zhang, L. Li, Z. Li, J. Huang, H. Hu, *et al.*, "Self-supervised learning enables 3d digital subtraction angiography reconstruction from ultra-sparse 2d projection views: a multicenter study," *Cell Reports Medicine*, vol. 3, no. 10, 2022.

[36] H. Zhang, B. Liu, H. Yu, and B. Dong, "Metainv-net: Meta inversion network for sparse view ct image reconstruction," *IEEE Transactions on Medical Imaging*, vol. 40, no. 2, pp. 621–634, 2020.

[37] R. Zha, Y. Zhang, and H. Li, "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.

[38] A. Corona-Figueroa, J. Frawley, S. Bond-Taylor, S. Bethapudi, H. P. Shum, and C. G. Willcocks, "Mednerf: Medical neural radiance fields for reconstructing 3d-aware ct-projections from a single x-ray," in *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 3843–3848, IEEE, 2022.

[39] Y. Wang, Y. Long, S. H. Fan, and Q. Dou, "Neural rendering for stereo 3d reconstruction of deformable tissues in robotic surgery," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 431–441, Springer, 2022.

[40] C. Yang, K. Wang, Y. Wang, X. Yang, and W. Shen, "Neural lerplane representations for fast 4d reconstruction of deformable tissues," *arXiv preprint arXiv:2305.19906*, 2023.

[41] Z. Zhou, H. Zhao, J. Fang, D. Xiang, L. Chen, L. Wu, F. Wu, W. Liu, C. Zheng, and X. Wang, "Tiavox: Time-aware attenuation voxels for sparse-view 4d dsa reconstruction," *arXiv preprint arXiv:2309.02318*, 2023.

[42] Y. Li, X. Fu, S. Zhao, R. Jin, and S. K. Zhou, "Sparse-view ct reconstruction with 3d gaussian volumetric representation," *arXiv preprint arXiv:2312.15676*, 2023.

[43] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross, "Surface splatting," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 371–378, 2001.

[44] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4104–4113, 2016.

[45] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[47] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

[48] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[49] Z. Zhu, Z. Fan, Y. Jiang, and Z. Wang, "Fsgs: Real-time few-shot view synthesis using gaussian splatting," *arXiv preprint arXiv:2312.00451*, 2023.

[50] Y. Lao, X. Xu, Z. Cai, X. Liu, and H. Zhao, "Corresnerf: Image correspondence priors for neural radiance fields," *arXiv preprint arXiv:2312.06642*, 2023.

[51] A. Guédon and V. Lepetit, "Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering," *arXiv preprint arXiv:2311.12775*, 2023.