

# FLOW DISTILLATION SAMPLING: REGULARIZING 3D GAUSSIANS WITH PRE-TRAINED MATCHING PRIORS

**Lin-Zhuo Chen** \*  
Nanjing University

**Kangjie Liu** \*  
Nanjing University

**Youtian Lin**  
Nanjing University

**Siyu Zhu**  
Fudan University

**Zhihao Li**  
Huawei Noah's Ark Lab

**Xun Cao**  
Nanjing University

**Yao Yao** †  
Nanjing University

## ABSTRACT

3D Gaussian Splatting (3DGS) has achieved excellent rendering quality with fast training and rendering speed. However, its optimization process lacks explicit geometric constraints, leading to suboptimal geometric reconstruction in regions with sparse or no observational input views. In this work, we try to mitigate the issue by incorporating a pre-trained matching prior to the 3DGS optimization process. We introduce Flow Distillation Sampling (FDS), a technique that leverages pre-trained geometric knowledge to bolster the accuracy of the Gaussian radiance field. Our method employs a strategic sampling technique to target unobserved views adjacent to the input views, utilizing the optical flow calculated from the matching model (Prior Flow) to guide the flow analytically calculated from the 3DGS geometry (Radiance Flow). Comprehensive experiments in depth rendering, mesh reconstruction, and novel view synthesis showcase the significant advantages of FDS over state-of-the-art methods. Additionally, our interpretive experiments and analysis aim to shed light on the effects of FDS on geometric accuracy and rendering quality, potentially providing readers with insights into its performance. Project page: <https://nju-3dv.github.io/projects/fds>.

## 1 INTRODUCTION

3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) has been widely applied to the field of 3D reconstruction and rendering, including novel view synthesis of static scenes (Kerbl et al., 2023; Yu et al., 2024a), mesh surface reconstruction (Guédon and Lepetit, 2024; Yu et al., 2024b), inverse rendering (Liang et al., 2024; Gao et al., 2023), object detection (Cao et al., 2024), and dynamic 3D reconstruction (Wu et al., 2024; Lin et al., 2024). However, in scenarios with less-observed areas, such as indoor scenes and unbounded scenes, radiance field optimization often suffers from overfitting to these limited input views (Li et al., 2024), resulting in unreliable and corrupted geometry reconstruction.

To mitigate the issue, recent research efforts (Li et al., 2024; Paliwal et al., 2024; Turkulainen et al., 2025) have focused on incorporating geometric priors from input views into the training process, thereby regulating the optimization of radiance field represented by 3D Gaussian points. For instance, DN-Splatter (Turkulainen et al., 2025) integrates sensor depth and normal cues into the reconstruction process. However, sensor depth acquisition is costly, and the depth prior information from pre-trained monocular deep models inevitably suffer from the scale ambiguity (Liu et al., 2023b). While the normal prior provides even better geometric details, the scale ambiguity still exists due to its monocular nature.

In contrast to monocular priors, pairwise matching priors can provide absolute scale information of the scene. In this paper, we introduce **Flow Distillation Sampling (FDS)**, an online method for distilling matching prior from a pre-trained optical flow model into the 3DGS training process. FDS aims to enhance the geometry quality of Gaussian radiance field by leveraging the matching prior

\*Equal contribution.

†Corresponding author.

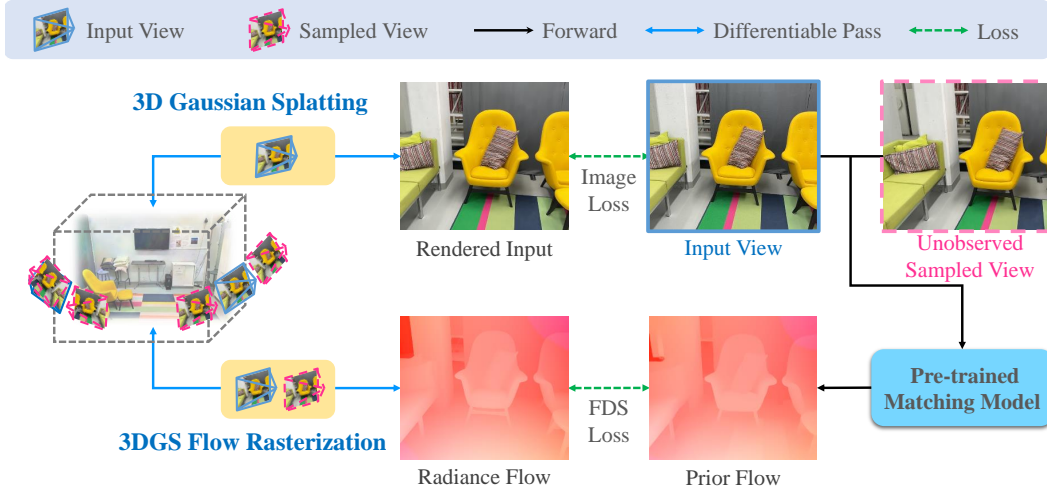


Figure 1. **Pipeline of the proposed FDS.** For each input view, we apply the FDS camera sampling scheme to generate corresponding unobserved sampled view. We then compute Radiance flow base on rendered depth and the Prior flow from matching prior model. Finally the Prior Flow is used to supervise Radiance flow, which enhances the geometric quality of Gaussian Radiance Field.

into the unobserved novel view. Specifically, we observe that the flow between the input view and the unobserved view generated by the match prior model (i.e., **Prior Flow**), can guide and refine the flow analytically calculated from the 3DGS geometry (i.e., **Radiance Flow**), improving the 3DGS reconstruction quality. Moreover, better 3DGS scene will lead to more accurate Prior Flow, creating a mutually reinforcing effect between two computed flow maps. This remains effective even when the radiance field is poorly optimized and the image rendered from the unobserved viewpoint is blurry during training. In addition, a camera sampling scheme is proposed to adaptively control the overlap between input view and sampled view for better Prior Flow calculation, which allows to leverage prior geometric knowledge more profoundly and thereby better enhance the 3DGS reconstruction quality.

The proposed FDS has been extensively evaluated on MushRoom (Ren et al., 2024), ScanNet (V2) (Dai et al., 2017), and Replica (Straub et al., 2019) datasets for the task of geometry reconstruction. We apply FDS to two commonly used baseline approaches, namely 3DGS (Kerbl et al., 2023) and 2DGS (Huang et al., 2024). The results demonstrate a significant improvement in geometry reconstruction accuracy. Additionally, we provide interpretive experiments and comprehensive analysis on the effectiveness of FDS, shedding light on its influence on the quality of geometric reconstruction and novel-view rendering. Our contributions are summarized as follows:

- FDS leverages matching prior information to recover absolute scale, significantly enhancing the geometric quality of the Gaussian radiance field.
- An adaptive camera sampling scheme is proposed to selectively choose unobserved views with controllable overlap with the input view, further improving the geometric quality of the Gaussian radiance field even in less-observed areas.
- By employing FDS and the adaptive camera sampling scheme, we bring significant improvements to state-of-the-art 3D geometry reconstruction approaches.

## 2 RELATED WORK

**Geometry Reconstruction Based on 3DGS:** To enhance the geometry of 3DGS representations, some studies (Gao et al., 2024a; Waczyńska et al., 2024; Lyu et al., 2024; Chen et al., 2023) focus on integrating mesh or SDF representations with 3DGS. (Gao et al., 2024a) improve the splitting of 3DGS under the guidance of mesh representation and deformation gradients. A hybrid representation that integrates 3DGS with mesh, allowing 3DGS to be modified as a mesh, is introduced in (Waczyńska et al., 2024). Mani-GS (Gao et al., 2024b) binds Gaussian and shape-aware triangular

mesh to manipulate 3DGS directly. 3DGSR (Lyu et al., 2024) combines a signed distance field (SDF) network with 3DGS to enhance geometry quality, aligning the SDF network’s geometry with that of 3DGS. NeuSG (Chen et al., 2023) utilizes point clouds from 3DGS to regulate NeuS, while its normals are also used to refine 3DGS. Additionally, NeuSG includes regularizers to ensure that 3DGS remains close to the surface. Some work (Guédon and Lepetit, 2024; Huang et al., 2024; Yu et al., 2024b; Chen et al., 2024) tends to extract geometry from 3DGS directly, benefiting from its fast training and rendering speed. Sugar (Guédon and Lepetit, 2024) proposes an efficient mesh extraction method from 3DGS, aligned with a regularization term during the training process and a refinement strategy. 2DGS (Huang et al., 2024) proposed a 2D surface modeling and two regularization losses which can preserve perspective-correct splatting and enhance geometry reconstruction. A ray-tracing-based volume rendering is introduced in GOF (Yu et al., 2024b), allowing the extraction of geometry from 3DGS directly. PGSR (Chen et al., 2024) proposes unbiased depth rendering, combined with single and multi-view regularization loss to preserve geometric consistency. However, in low-texture and less frequently observed areas, such as indoor scenes, 3DGS still tends to overfit to the limited input views, necessitating regulation through additional prior cues.

**Prior Regulation for Rendering:** Prior information is usually incorporated in ill-posed problems, including sparse view novel view synthesis, dynamic scene reconstruction, and mesh reconstruction. For sparse novel view synthesis tasks, most works utilize the prior depth information (Deng et al., 2022; Roessle et al., 2022; Song et al., 2024; Wang et al., 2023b; Zhu et al., 2023; Xiong et al., 2023; Paliwal et al., 2024), semantic information (Jain et al., 2021; Wynn and Turmukhambetov, 2023; Xiong et al., 2023), and matching information (Paliwal et al., 2024; Lao et al., 2024) to constrain the optimization process of 3DGS. Dynamic scene reconstruction is another challenging task which requires reconstructing the scene geometry and object motion at the same time. Therefore, optical flow priors are crucial as they can help distinguish between camera motion and object motion (Liu et al., 2023b) while providing motion priors between frames (Liu et al., 2023a; Gao et al., 2021; Li et al., 2023; Wang et al., 2023a; Guo et al., 2023; Tian et al., 2023). For the mesh reconstruction task, geometry quality is also enhanced by depth priors (Wei et al., 2021; Yu et al., 2022; Turkulainen et al., 2025) or normal priors (Yu et al., 2022; Turkulainen et al., 2025). However, using optical flow model priors to obtain metric depth priors to help geometric reconstruction, while leveraging unobserved regions to enhance the quality of limited view reconstruction, has not yet been explored.

### 3 METHOD

Our FDS regulates the optimization of Gaussian radiance field by incorporating matching priors from the pretrained deep model. The generation of Radiance Flow and our proposed FDS loss, along with the equipped camera sampling scheme, are detailed in Sec. 3.1 and Sec. 3.2, respectively.

#### 3.1 3D GAUSSIAN SPLATTING AND RADIANCE FLOW

We utilize 3DGS as an example to demonstrate how Radiance Flow is generated. 3DGS (Kerbl et al., 2023) employs a collection of 3D Gaussians to represent the 3D scene. The expression of  $i$ -th 3D Gaussian distribution shows below:

$$\mathcal{G}_i(\mathbf{X}) = e^{-\frac{1}{2}(\mathbf{X}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{X}-\boldsymbol{\mu}_i)}, \quad (1)$$

where  $\boldsymbol{\Sigma}_i = \mathbf{R}_i \mathbf{S}_i \mathbf{S}_i^T \mathbf{R}_i^T$ , and  $\boldsymbol{\mu}_i$  represents the position of the Gaussian, which is optimized during training. Both  $\mathbf{S}_i$  and  $\mathbf{R}_i$  are represented by a 3D scaling vector  $\mathbf{s}_i$  and a quaternion  $\mathbf{q}_i$ , respectively. In addition to the above parameters, each Gaussian  $\mathbf{P}_i$  has extra learnable attributes, including opacity  $\alpha_i$  and color feature  $\mathbf{f}_i$ .

To render a color for pixel  $x$ , a volume rendering based method similar to NeRF (Mildenhall et al., 2021) is adapted:

$$C(x) = \sum_{i \in N} c_i \hat{\alpha}_i \prod_{j=1}^{i-1} (1 - \hat{\alpha}_j), \quad (2)$$

where  $c_i$  is the color of each Gaussian point, represented by spherical harmonics.  $\hat{\alpha}$  is the blending weight for each 2D Gaussian point which is the projection of 3D Gaussian points on the image plane.  $N$  is the number of Gaussian points. Similarly, the depth of pixel  $x$  is rendered using alpha blending.

$$D(x) = \frac{\sum_{i \in N} d_i \hat{\alpha}_i \prod_{j=1}^{i-1} (1 - \hat{\alpha}_j)}{\sum_{i \in N} \hat{\alpha}_i \prod_{j=1}^{i-1} (1 - \hat{\alpha}_j)}, \quad (3)$$

where  $d_i$  is the distance from the target camera to the Gaussian Point  $P_i$ . We can also render Radiance Flow between two views of camera  $m, n$  using their camera poses and position of Gaussian points  $\mu_i$ . As mentioned above, we can project pixel  $x = (u_1, v_1)$  in  $m$ -th view image to the  $n$ -th view by its corresponding depth and their pose transformation:

$$D^n(u_2, v_2) \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} = \mathbf{K} T_m^n \mathbf{K}^{-1} D^m(u_1, v_1) \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix}, \quad (4)$$

where  $T_m^n$  is the relative transformation from  $m$ -th view to  $n$ -th view,  $\mathbf{K}$  is the intrinsic matrix, and  $D^m(u_1, v_1)$  is the rendered depth of  $(u_1, v_1)$  in  $m$ -th view.

Next, we calculate  $F^{m \rightarrow n}(x)$  for pixel  $x = (u_1, v_1)$  from view  $m$  to view  $n$ :

$$F^{m \rightarrow n}(x) = \begin{bmatrix} u_2 - u_1 \\ v_2 - v_1 \end{bmatrix}. \quad (5)$$

For other type of Gaussian radiance field such as 2DGS (Huang et al., 2024), we only need to replace the alpha blending based depth  $D(x)$  with corresponding formulation.

### 3.2 FLOW DISTILLATION SAMPLING

Given a collection of images  $\{I^i\}_{i=1,2,\dots,N}$ , Gaussian Radiance Field typically employs the following loss function for rendering optimization:

$$L = \frac{1}{B} \sum_{i=1}^B (1 - \lambda) L_1 + \lambda L_{D-SSIM} + \lambda_{normal} L_n, \quad (6)$$

where  $B$  denotes batch size, and  $L_n$  represents the normal consistency loss from (Huang et al., 2024). However, when  $N$  is small, this representation suffers from overfitting (Li et al., 2024; Paliwal et al., 2024).

We propose the Flow Distillation Sampling (FDS) method to incorporate pretrained matching priors into the rendering optimization process, thereby mitigating overfitting and improving rendering performance.

The camera sampling scheme and loss function design in FDS are introduced in Sec. 3.2.1 and Sec. 3.2.2.

#### 3.2.1 CAMERA SAMPLING SCHEME

During the training process, FDS randomly samples unobserved camera views nearby the input view and then incorporates the matching priors into these views. To utilize the matching prior information between input view and sampled view more profoundly, we propose a camera sampling scheme that ensures sufficient movement to perceive the geometry of the scene while avoiding excessively abrupt motions that could make it difficult for the prior model to match.

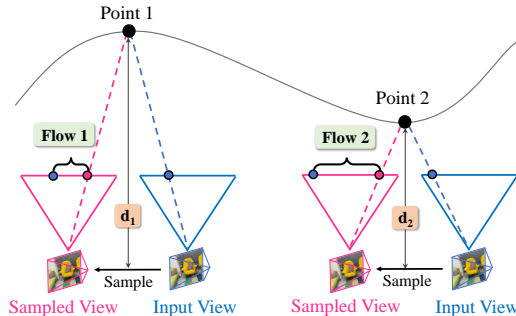


Figure 2. **Explanation of depth-adaptive translation radius.** A fixed-radius camera sampling scheme may result in significantly different flow values (Flow 1 and Flow 2) in areas with varying depth ( $d_1$  and  $d_2$ ).

Given the world to camera transformation matrix  $T^i$  for the  $i$ -th input view, a small translation disturbance  $\mathbf{t}$  and rotation disturbance  $\mathbf{R}$  are applied to get the sampled transformation matrix  $T^s$ , expressed as:

$$T^s = ((T^i)^{-1}E)^{-1} \quad (7)$$

where  $E = [\mathbf{R}, \mathbf{t}]$  denotes the transformation matrix from the sampled view to the input view, and  $\mathbf{t}$  is a translation vector satisfying  $|\mathbf{t}| = \epsilon_t$ ,  $\mathbf{t} = (t_1, t_2, t_3)$ , indicating that unobserved view is sampled along a small spiral circle with a radius of  $\epsilon_t$ . As noted in (Bian et al., 2021), the rotation flow in image warping is independent of depth. Therefore, we set the rotation part to identity matrix. So that  $E$  is a pure-translation transformation. For the translation radius  $\epsilon_t$ , as illustrated in Fig. 2, using a uniform radius for all views leads to varying flows due to the differences in depths; the closer the depth, the larger the flow. To ensure flow consistency across all views while maintaining controllable overlap between the input views and sampled views, we hope to implement a depth-adaptive radius, which can help to preserve the same flow between all input views and their sampled views.

According to (4), the pure translation transformation (Bian et al., 2021) between the input view  $i$  and its sampled view  $s$  is shown below:

$$D^s(u_2, v_2) \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} = D^i(u_1, v_1) \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} + \mathbf{K} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \quad (8)$$

where  $\mathbf{K}$  is the intrinsic matrix of the camera, after solving the above equation (Bian et al., 2021), we get:

$$\begin{bmatrix} u_2 \\ v_2 \end{bmatrix} = \begin{bmatrix} \frac{D_i(u_1, v_1)u_1 + f_x t_1 + c_x t_3}{D_i(u_1, v_1) + t_3} \\ \frac{D_i(u_1, v_1)u_1 + f_y t_2 + c_x t_3}{D_i(u_1, v_1) + t_3} \end{bmatrix} \quad (9)$$

We set  $t_3 = 0$  in our camera sampling scheme and assume camera intrinsic parameters:  $f_x \approx f_y = f$ . The radiance flow  $\mathbf{F}^{i \rightarrow s}(u_1, v_1) = \begin{bmatrix} u_2 - u_1 \\ v_2 - v_1 \end{bmatrix}$  from the input view  $i$  to its sampled view  $s$  is shown below:

$$\mathbf{F}^{i \rightarrow s}(u_1, v_1) = \begin{bmatrix} \frac{f}{D_i(u_1, v_1)} t_1 \\ \frac{f}{D_i(u_1, v_1)} t_2 \end{bmatrix} \quad (10)$$

We aim to keep the value of  $\|\mathbf{F}^{i \rightarrow s}(u_1, v_1)\|_2$  constant for the pixel  $x = (u_1, v_1)$  during each camera sampling. By setting  $\|\mathbf{F}^{i \rightarrow s}(u_1, v_1)\|_2 = \sigma$  and incorporating this with (10), we get:

$$\epsilon_t = \sqrt{t_1^2 + t_2^2} = \sigma \frac{D_i(u_1, v_1)}{f} \quad (11)$$

Thus, the radius of translation in our camera sampling is defined as  $\epsilon_t = \sigma \frac{D_i(u_1, v_1)}{f}$  which helps maintain stable flow. The parameter  $\sigma$  can be tuned as a hyperparameter, which represents the mean radiance flow between the input view and its sampled view. Given that pixel depths vary within an image, we use the mean depth  $\bar{D}_i$  of the image and set the radius of our translation  $\epsilon_t = \sigma \frac{\bar{D}_i}{f}$ .

$$\mathbf{t} = \begin{bmatrix} \sigma \frac{\bar{D}_i}{f} \sin(2\pi\xi) \\ \sigma \frac{\bar{D}_i}{f} \cos(2\pi\xi) \\ 0 \end{bmatrix} \quad (12)$$

Where  $\xi \sim U(0, 1)$  is a uniform random value between  $[0, 1]$  during training.

### 3.2.2 FLOW DISTILLATION SAMPLING LOSS

Specifically, to optimize the rendered results of the  $i$ -th view, every iteration, we sample an unobserved view  $s$  near the  $i$ -th input training camera to render the color image  $C^s$ . The Radiance Flow  $F^{i \rightarrow s}(\mathbf{p})$ , guided by Equ. (5), benefits from the introduced matching priors. We define the Prior Flow  $\bar{F}^{i \rightarrow s}$  as:

$$\bar{F}^{i \rightarrow s} = \mathcal{M}_\theta(I^i, C^s), \quad (13)$$

where  $\theta$  represents the parameters of pretrained network  $\mathcal{M}$ ,  $I^i$  is the rgb ground truth of view  $i$ , and  $C^s$  is the rendered image of sampled view  $s$ . Instead of using  $C^i$  to generate Prior Flow, we find that using  $I^i$  can help us to remove the floaters in  $C^s$  as shown in Tab. 4. However, both  $F^{i \rightarrow s}$  and  $\bar{F}^{i \rightarrow s}$  face challenges:

- $F^{i \rightarrow s}$ , derived from Equ. (5), is inaccurate due to the incorrect positioning of Gaussian points.
- $\bar{F}^{i \rightarrow s}$ , derived from Equ. (13), suffers from the blurred rendering quality of  $C^s$ .

Despite these inaccuracies, we observe that  $\bar{F}^{i \rightarrow s}$  is more robust and precise during training compared with  $F^{i \rightarrow s}$ . Motivated by this observation, we aim to utilize  $\bar{F}^{i \rightarrow s}$  to refine  $F^{i \rightarrow s}$ , which can make  $F^{i \rightarrow s}$  more accurate, enhancing the positioning of Gaussian points and subsequently improving the rendering quality of  $C^s$ . This process also leads to a more accurate  $\bar{F}^{i \rightarrow s}$ . Based on these observations, we propose the FDS loss, which distills matching priors from a pretrained deep model through mutual refinement of the two flows:

$$L_{fds} = \|\bar{F}^{i \rightarrow s} - F^{i \rightarrow s}\|_2 \quad (14)$$

---

#### Algorithm 1 Flow Distillation Sampling

---

**Input:** A batch of input training image:  $\{I_i\}_{i=1}^N$ , Transformation Matrix:  $\{T_i\}_{i=1}^N$ , Prior Matching Network  $\mathcal{M}_\theta$ , Gaussian Points  $\{P_i\}_{i=1}^M$  with  $\{r_i, t_i, f_i, \mu_i, \alpha_i\}$ .

**Output:**  $L_{fds}$

- 1: **for**  $i$  in  $\{1, 2, \dots, B\}$  **do**
- 2:    $\xi \sim U(0, 1)$ ,  $\mathbf{R} \leftarrow \mathbf{I}$
- 3:    $t_1 \leftarrow \sigma \frac{D^i}{f^i} \sin(2\pi\xi)$ ,  $t_2 \leftarrow \sigma \frac{D^i}{f^i} \cos(2\pi\xi)$ ,  $t_3 \leftarrow 0$
- 4:    $\mathbf{E} \leftarrow [\mathbf{R}, \mathbf{t}]$ ,  $\mathbf{T}^s \leftarrow ((\mathbf{T}^i)^{-1} \mathbf{E})^{-1}$
- 5:    $C^s, D^s \leftarrow \text{Render}(\mathbf{T}^s, \mathbf{P})$
- 6:    $C^i, D^i \leftarrow \text{Render}(\mathbf{T}^i, \mathbf{P})$
- 7:    $\mathbf{X}^s \leftarrow \mathbf{K} \mathbf{T}_i^s \mathbf{K}^{-1} \mathbf{D}^i (\mathbf{X}^i) \mathbf{X}^i / \mathbf{D}(\mathbf{X}^s)$
- 8:    $F^{i \rightarrow s} \leftarrow \mathbf{X}^s - \mathbf{X}^i$
- 9:    $\bar{F}^{i \rightarrow s} \leftarrow \mathcal{M}_\theta(I^i, C^s)$
- 10:    $L_{fds} \leftarrow L_{fds} + 1/B \|\bar{F}^{i \rightarrow s} - F^{i \rightarrow s}\|_2$
- 11: **return**  $L_{fds}$

---

To maintain training stability and reduce computational complexity, we detach  $\bar{F}^{i \rightarrow s}$  from  $P_i$  when we calculate loss. This prevents the propagation of gradients, as computing them is resource-intensive and time-consuming, as noted in (Poole et al., 2022). Additionally, this helps to ensure that  $\bar{F}^{i \rightarrow s}$  is not directly influenced by the less reliable  $F^{i \rightarrow s}$ .

In summary, the procedures of our proposed Flow Distillation Sampling are presented in Algorithm 1. The overall training loss is:

$$L = \frac{1}{B} \sum_{i=1}^B (1 - \lambda) L_1 + \lambda L_{D-SSIM} + \lambda_{normal} L_n + \lambda_{fds} L_{fds}. \quad (15)$$

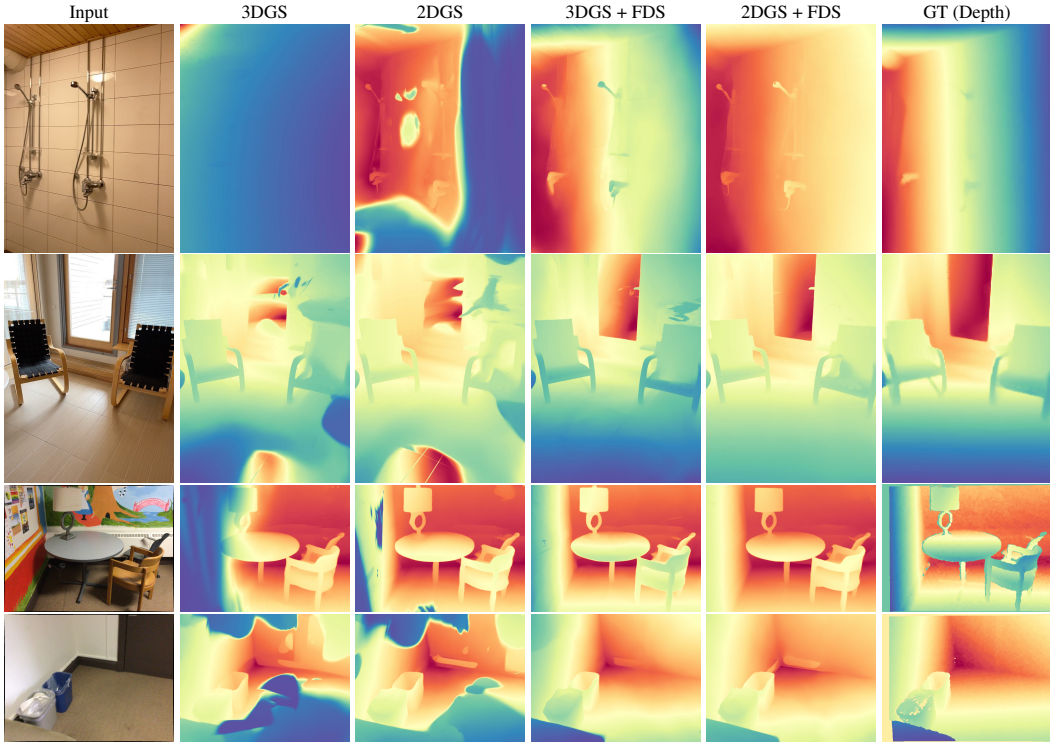


Figure 3. **Comparison of depth reconstruction on Mushroom and ScanNet datasets.** The original 3DGS or 2DGS model equipped with FDS can remove unwanted floaters and reconstruct geometry more precisely.

## 4 EXPERIMENTS

### 4.1 SETUPS

#### 4.1.1 IMPLEMENTATION DETAILS

We apply our FDS method to two types of 3DGS: the original 3DGS, and 2DGS (Huang et al., 2024). The number of iterations in our optimization process is 35,000. We follow the default training configuration and apply our FDS method after 15,000 iterations, then we add normal consistency loss for both 3DGS and 2DGS after 25000 iterations. The weight for FDS,  $\lambda_{fds}$ , is set to 0.015, the  $\sigma$  is set to 23, and the weight for normal consistency is set to 0.15 for all experiments. We removed the depth distortion loss in 2DGS because we found that it degrades its results in indoor scenes. The Gaussian point cloud is initialized using Colmap for all datasets. We tested the impact of using Sea Raft (Wang et al., 2024) and Raft (Teed and Deng, 2020) on FDS performance. Due to the blurriness of the ScanNet dataset, additional prior constraints are required. Thus, we incorporate normal prior supervision on the rendered normals in ScanNet (V2) dataset by default. The normal prior is predicted by the Stable Normal model (Ye et al., 2024) across all types of 3DGS. The entire framework is implemented in PyTorch (Paszke et al., 2019), and all experiments are conducted on a single NVIDIA 4090D GPU.

#### 4.1.2 DATASETS AND METRICS

We evaluate our method for 3D reconstruction and novel view synthesis tasks on **Mushroom** (Ren et al., 2024), **ScanNet (v2)** (Dai et al., 2017), and **Replica** (Straub et al., 2019) datasets, which feature challenging indoor scenes with both sparse and dense image sampling. The Mushroom dataset is an indoor dataset with sparse image sampling and two distinct camera trajectories. We train our model on the training split of the long capture sequence and evaluate novel view synthesis on the test split of the long capture sequences. Five scenes are selected to evaluate our FDS, including "coffee room", "honka", "kokko", "sauna", and "vr room". ScanNet(V2) (Dai et al., 2017) consists of 1,613 indoor scenes with annotated camera poses and depth maps. We select 5 scenes from the ScanNet (V2)

Table 1. 3D Reconstruction and novel view synthesis results on Mushroom dataset. \* Represents that FDS uses the Raft model.

Method	Acc↓	Comp↓	C-L1↓	NC↑	F-Score↑	Abs Rel↓	PSNR↑	SSIM↑	LPIPS↓	Time↓
GOF	0.1812	0.1093	0.1453	0.6292	0.3665	0.2380	21.37	0.7762	0.3132	≈1.4h
PGSR	0.0971	0.1420	0.1196	0.7193	0.5105	0.1723	22.13	0.7773	0.2918	≈1.2h
3DGS	0.1167	0.1033	0.1100	0.7954	0.3739	0.1214	24.18	0.8392	0.2511	≈0.8h
3DGS + FDS*	0.0569	0.0676	0.0623	0.8105	0.6573	0.0603	24.72	0.8489	0.2379	≈1.3h
3DGS + FDS	<b>0.0527</b>	<b>0.0565</b>	<b>0.0546</b>	<b>0.8178</b>	<b>0.6958</b>	<b>0.0568</b>	<b>24.76</b>	<b>0.8486</b>	<b>0.2381</b>	≈1.3h
2DGS	0.1078	0.0850	0.0964	0.7835	0.5170	0.1002	23.56	0.8166	0.2730	≈0.8h
2DGS + FDS*	0.0689	0.0646	0.0667	0.8042	0.6582	0.0589	23.98	0.8255	0.2621	≈1.3h
2DGS + FDS	<b>0.0615</b>	<b>0.0534</b>	<b>0.0574</b>	<b>0.8151</b>	<b>0.6974</b>	<b>0.0561</b>	<b>24.06</b>	<b>0.8271</b>	<b>0.2610</b>	≈1.3h

Table 2. 3D Reconstruction and novel view synthesis results on ScanNet dataset.

Method	Acc↓	Comp↓	C-L1↓	NC↑	F-Score↑	Abs Rel↓	PSNR↑	SSIM↑	LPIPS↓
GOF	1.8671	0.0805	0.9738	0.5622	0.2526	0.1597	21.55	0.7575	0.3881
PGSR	0.2928	0.5103	0.4015	0.5567	0.1926	0.1661	21.71	0.7699	0.3899
3DGS	0.4867	0.1211	0.3039	0.7342	0.3059	0.1227	22.19	0.7837	0.3907
3DGS + FDS	<b>0.2458</b>	<b>0.0787</b>	<b>0.1622</b>	<b>0.7831</b>	<b>0.4482</b>	<b>0.0573</b>	<b>22.83</b>	<b>0.7911</b>	<b>0.3826</b>
2DGS	0.2658	0.0845	0.1752	0.7504	0.4464	0.0831	22.59	0.7881	0.3854
2DGS + FDS	<b>0.1457</b>	<b>0.0679</b>	<b>0.1068</b>	<b>0.7883</b>	<b>0.5459</b>	<b>0.0432</b>	<b>22.91</b>	<b>0.7928</b>	<b>0.3800</b>

dataset, uniformly sampling one-tenth of the views, following the approach in (Guo et al., 2022). To further improve the geometry rendering quality of 3DGS, Replica (Straub et al., 2019) contains small-scale real-world indoor scans. We evaluate our FDS on five scenes from Replica: office0, office1, office2, office3 and office4, selecting one-tenth of the views for training. The results for Replica are provided in the supplementary materials. To evaluate the rendering quality and geometry of 3DGS, we report PSNR, SSIM, and LPIPS for rendering quality, along with Absolute Relative Distance (Abs Rel) as a depth quality metrics. Additionally, for mesh evaluation, we use metrics including Accuracy, Completion, Chamfer-L1 distance, Normal Consistency, and F-scores.

## 4.2 RESULTS

### 4.2.1 DEPTH RENDERING AND NOVEL VIEW SYNTHESIS

The comparison results on Mushroom and ScanNet are presented in Tab. 1 and Tab. 2, respectively. Due to the sparsity of sampling in the Mushroom dataset, challenges are posed for both GOF (Yu et al., 2024b) and PGSR (Chen et al., 2024), leading to their relative poor performance on the Mushroom dataset. Our approach achieves the best performance with the FDS method applied during the training process. The FDS significantly enhances the geometric quality of 3DGS on the Mushroom dataset, improving the "abs rel" metric by more than 50%. We found that Sea Raft (Wang et al., 2024) outperforms Raft (Teed and Deng, 2020) on FDS, indicating that a better optical flow model can lead to more significant improvements. Additionally, the render quality of RGB images shows a slight improvement, by 0.58 in 3DGS and 0.50 in 2DGS, benefiting from the incorporation of cross-view consistency in FDS. On the Mushroom dataset, adding the FDS loss increases the training time by half an hour, which maintains the same level as baseline. Similarly, our method shows a notable improvement on the ScanNet dataset as well using Sea Raft (Wang et al., 2024) Model. The "abs rel" metric in 2DGS is improved nearly 50%. This demonstrates the robustness and effectiveness of the FDS method across different datasets.

The qualitative comparisons on the Mushroom and ScanNet dataset are illustrated in Fig. 3. As seen in the first row of Fig. 3, both the original 3DGS and 2DGS suffer from overfitting, leading to corrupted geometry generation. Our FDS effectively mitigates this issue by supervising the matching relationship between the input and sampled views, helping to recover the geometry. FDS also improves the refinement of geometric details, as shown in other rows. By incorporating the matching prior through FDS, the quality of the rendered depth is significantly improved.

### 4.2.2 MESH EXTRACTION

To further demonstrate the improvement in geometry quality, we applied methods used in (Turkulainen et al., 2025) to extract meshes from the input views of optimized 3DGS and 2DGS. The



Table 3. Ablation study on geometry priors.

Method	Acc.↓	Comp ↓	C-L1 ↓	NC ↑	F-Score ↑	Abs Rel ↓	PSNR ↑	SSIM ↑	LPIPS ↓
2DGS	0.1078	0.0850	0.0964	0.7835	0.5170	0.1002	23.56	0.8166	0.2730
2DGS+Depth	0.0862	0.0702	0.0782	0.8153	0.5965	0.0672	23.92	0.8227	0.2619
2DGS+MVDepth	0.2065	0.0917	0.1491	0.7832	0.3178	0.0792	23.74	0.8193	0.2692
2DGS+Normal	0.0939	0.0637	0.0788	<b>0.8359</b>	0.5782	0.0768	23.78	0.8197	0.2676
2DGS+FDS	<b>0.0615</b>	<b>0.0534</b>	<b>0.0574</b>	0.8151	<b>0.6974</b>	<b>0.0561</b>	<b>24.06</b>	<b>0.8271</b>	<b>0.2610</b>
2DGS+Depth+FDS	0.0561	0.0519	0.0540	0.8295	0.7282	0.0454	<b>24.22</b>	<b>0.8291</b>	<b>0.2570</b>
2DGS+Normal+FDS	<b>0.0529</b>	<b>0.0450</b>	<b>0.0490</b>	0.8477	<b>0.7430</b>	<b>0.0443</b>	24.10	0.8283	0.2590
2DGS+Depth+Normal	0.0695	0.0513	0.0604	<b>0.8540</b>	0.6723	0.0523	24.09	0.8264	0.2575
2DGS+Depth+Normal+FDS	<b>0.0506</b>	<b>0.0423</b>	<b>0.0464</b>	<b>0.8598</b>	<b>0.7613</b>	<b>0.0403</b>	<b>24.22</b>	<b>0.8300</b>	<b>0.0403</b>

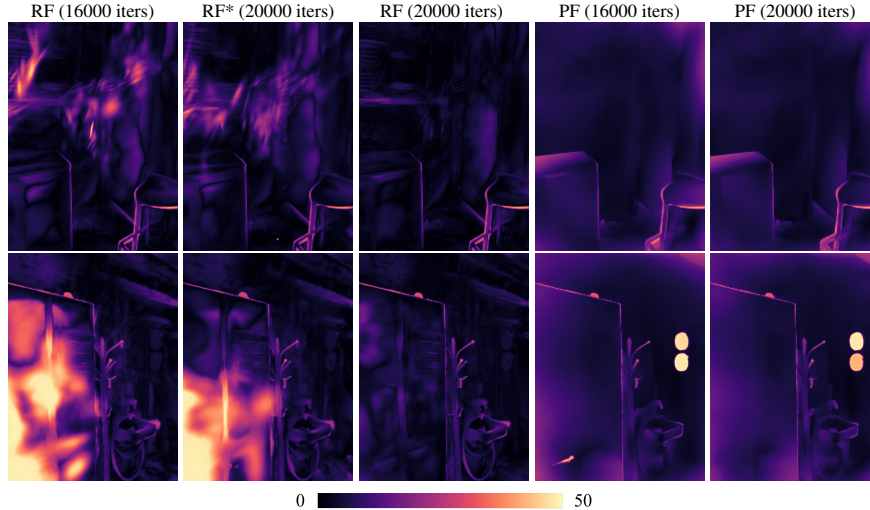


Figure 4. The error map of Radiance Flow and Prior Flow. RF: Radiance Flow, PF: Prior Flow, \* means that there is no FDS loss supervision during optimization.

comparison results are presented in Tab. 1. With the integration of FDS, the mesh quality is significantly enhanced compared to the baseline, featuring fewer floaters and more well-defined shapes.

### 4.3 ABLATION STUDY

**Ablation study on geometry priors:** To highlight the advantage of incorporating matching priors, we incorporated various types of priors generated by different models into 2DGS. These include a monocular depth estimation model (Depth Anything v2) (Yang et al., 2024), a two-view depth estimation model (Unimatch) (Xu et al., 2023), and a monocular normal estimation model (StableNormal) (Ye et al., 2024). We adapt the scale and shift-invariant loss in Midas (Birkl et al., 2023) for monocular depth supervision and L1 loss for two-view depth supervision. We use Sea Raft (Wang et al., 2024) as our default optical flow model. The comparison results on Mushroom dataset are shown in Tab. 3. We observe that the normal prior provides accurate shape information, enhancing the geometric quality of the radiance field. The multi-view depth prior, hindered by the limited feature overlap between input views, fails to offer reliable geometric information. We test average "Abs Rel" of multi-view depth prior, and the result is 0.19, which performs worse than the "Abs Rel" results rendered by original 2DGS. From the results, it can be seen that depth order information provided by monocular depth improves reconstruction accuracy. Meanwhile, our FDS achieves the best performance among all the priors, and by integrating all three components, we obtained the optimal results.

**Ablation study on FDS:** In this section, we present the design of our FDS method through an ablation study on the Mushroom dataset to validate its effectiveness. The optional configurations of FDS are outlined in Tab. 4. Our base model is the 2DGS equipped with FDS, and its results are shown in the first row. The goal of this analysis is to evaluate the impact of various strategies on FDS sampling and loss design. We observe that when we replace  $I_i$  in (13) with  $C_i$ , as shown in the second row, the geometric quality of 2DGS deteriorates. Using  $I_i$  instead of  $C_i$  help us to remove the

Table 4. Ablation study on FDS strategies.

$\mathcal{M}_\theta(X, C^s)$		Loss			Metric		
$X = C^i$	$X = I^i$	Next Input view	Sampled view	Fixed Sampled view	Abs Rel ↓	F-score ↑	NC ↑
✓	✓		✓		<b>0.0561</b>	<b>0.6974</b>	<b>0.8151</b>
	✓	✓	✓		0.0839	0.6242	0.8030
	✓			✓	0.0877	0.6091	0.7614
	✓				0.0724	0.6312	0.8015

floaters in  $C^s$ , which are also remained in  $C^i$ . We also experiment with modifying the FDS loss. For example, in the third row, we use the next training input view as the sampling view, and replace the render result of next training view with ground truth image of its input view. Due to the significant movement between images, the Prior Flow fails to accurately match the pixel between them, leading to a further degradation in geometric quality. Finally, we attempt to fix the sampling view and found that this severely damaged the geometric quality, indicating that random sampling is essential for the stability of the mean error in the Prior Flow.

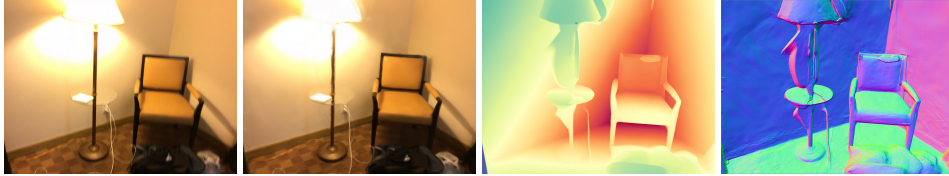


Figure 5. Limitation of FDS.

**Interpretive Experiments:** To demonstrate the mutual refinement of two flows in our FDS, for each view, we sample the unobserved views multiple times to compute the mean error of both Radiance Flow and Prior Flow. We use Raft (Teed and Deng, 2020) as our default optical flow model for visualization. The ground truth flow is calculated based on Eq. (4) and Eq. (5) utilizing ground truth depth in dataset. We introduce our FDS loss after 16000 iterations during optimization of 2DGS. The error maps are shown in Fig. 4. Our analysis reveals that Radiance Flow tends to exhibit significant geometric errors, whereas Prior Flow can more accurately estimate the geometry, effectively disregarding errors introduced by floating Gaussian points.

#### 4.4 LIMITATION AND FURTHER WORK

Firstly, our FDS faces challenges in scenes with significant lighting variations between different views, as shown in the lamp of first row in Fig. 5. Incorporating exposure compensation into FDS could help address this issue. Additionally, our method struggles with reflective surfaces and motion blur, leading to incorrect matching. In the future, we plan to explore the potential of FDS in monocular video reconstruction tasks, using only a single input image at each time step.

## 5 CONCLUSIONS

In this paper, we propose Flow Distillation Sampling (FDS), which leverages the matching prior between input views and sampled unobserved views from the pretrained optical flow model, to improve the geometry quality of Gaussian radiance field. Our method can be applied to different approaches (3DGS and 2DGS) to enhance the geometric rendering quality of the corresponding neural radiance fields. We apply our method to the 3DGS-based framework, and the geometry is enhanced on the Mushroom, ScanNet, and Replica datasets.

## ACKNOWLEDGEMENTS

This work was supported by National Key R&D Program of China (2023YFB3209702), the National Natural Science Foundation of China (62441204, 62472213), and Gusu Innovation & Entrepreneurship Leading Talents Program (ZXL2024361)

## REFERENCES

- Jia-Wang Bian, Huangying Zhan, Naiyan Wang, Tat-Jun Chin, Chunhua Shen, and Ian Reid. Auto-rectify network for unsupervised indoor depth estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(12):9802–9813, 2021. 5
- Reiner Birkl, Diana Wofk, and Matthias Müller. Midas v3.1 – a model zoo for robust monocular relative depth estimation. *arXiv preprint arXiv:2307.14460*, 2023. 9
- Yang Cao, Yuanliang Ju, and Dan Xu. 3dgs-det: Empower 3d gaussian splatting with boundary guidance and box-focused sampling for 3d object detection, 2024. URL <https://arxiv.org/abs/2410.01647>. 1
- Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *arXiv preprint arXiv:2406.06521*, 2024. 3, 8
- Hanlin Chen, Chen Li, and Gim Hee Lee. Neusg: Neural implicit surface reconstruction with 3d gaussian splatting guidance. *arXiv preprint arXiv:2312.00846*, 2023. 2, 3
- Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5828–5839, 2017. 2, 7
- Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12882–12891, 2022. 3
- Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Int. Conf. Comput. Vis.*, pages 5712–5721, 2021. 3
- Jian Gao, Chun Gu, Youtian Lin, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. Relightable 3d gaussian: Real-time point cloud relighting with brdf decomposition and ray tracing. *arXiv preprint arXiv:2311.16043*, 2023. 1
- Lin Gao, Jie Yang, Bo-Tao Zhang, Jia-Mu Sun, Yu-Jie Yuan, Hongbo Fu, and Yu-Kun Lai. Mesh-based gaussian splatting for real-time large-scale deformation. *arXiv preprint arXiv:2402.04796*, 2024a. 2
- Xiangjun Gao, Xiaoyu Li, Yiyu Zhuang, Qi Zhang, Wenbo Hu, Chaopeng Zhang, Yao Yao, Ying Shan, and Long Quan. Mani-gs: Gaussian splatting manipulation with triangular mesh. *arXiv preprint arXiv:2405.17811*, 2024b. 2
- Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024. 1, 3
- Haoyu Guo, Sida Peng, Haotong Lin, Qianqian Wang, Guofeng Zhang, Hujun Bao, and Xiaowei Zhou. Neural 3d scene reconstruction with the manhattan-world assumption. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 8
- Xiang Guo, Jiadai Sun, Yuchao Dai, Guanying Chen, Xiaoqing Ye, Xiao Tan, Errui Ding, Yumeng Zhang, and Jingdong Wang. Forward flow for novel view synthesis of dynamic scenes. In *Int. Conf. Comput. Vis.*, pages 16022–16033, 2023. 3
- Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH Anal. Conf.*, 2024. doi: 10.1145/3641519.3657428. 2, 3, 4, 7
- Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Int. Conf. Comput. Vis.*, pages 5885–5894, 2021. 3
- Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):1–14, 2023. 1, 2, 3

- Yixing Lao, Xiaogang Xu, Xihui Liu, Hengshuang Zhao, et al. Corresnerf: Image correspondence priors for neural radiance fields. *Adv. Neural Inform. Process. Syst.*, 36, 2024. 3
- Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. *arXiv preprint arXiv:2403.06912*, 2024. 1, 4
- Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4273–4284, 2023. 3
- Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. Gs-ir: 3d gaussian splatting for inverse rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 21644–21653, 2024. 1
- Youtian Lin, Zuozhuo Dai, Siyu Zhu, and Yao Yao. Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024. 1
- Yu-Lun Liu, Chen Gao, Andréas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13–23, June 2023a. 3
- Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13–23, 2023b. 1, 3
- Xiaoyang Lyu, Yang-Tian Sun, Yi-Hua Huang, Xiuzhe Wu, Ziyi Yang, Yilun Chen, Jiangmiao Pang, and Xiaojuan Qi. 3dgsr: Implicit surface reconstruction with 3d gaussian splatting. *ACM Trans. Graph.*, 43(6):1–12, 2024. 2, 3
- Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 3
- Avinash Paliwal, Wei Ye, Jinhui Xiong, Dmytro Kotovenko, Rakesh Ranjan, Vikas Chandra, and Nima Khademi Kalantari. Coherentgs: Sparse novel view synthesis with coherent 3d gaussians. *arXiv*, 2024. 1, 3, 4
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inform. Process. Syst.*, 32, 2019. 7
- Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 6
- Xuqian Ren, Wenjia Wang, Dingding Cai, Tuuli Tuominen, Juho Kannala, and Esa Rahtu. Mushroom: Multi-sensor hybrid room dataset for joint 3d reconstruction and novel view synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4508–4517, 2024. 2, 7
- Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12892–12901, 2022. 3
- Jiuhn Song, Seonghoon Park, Honggyu An, Seokju Cho, Min-Seop Kwak, Sungjin Cho, and Seungryong Kim. Därf: Boosting radiance fields from sparse input views with monocular depth adaptation. *Adv. Neural Inform. Process. Syst.*, 36, 2024. 3
- Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 2, 7, 8

- Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Eur. Conf. Comput. Vis.*, pages 402–419. Springer, 2020. 7, 8, 10
- Fengrui Tian, Shaoyi Du, and Yueqi Duan. Mononerf: Learning a generalizable dynamic radiance field from monocular videos. In *Int. Conf. Comput. Vis.*, pages 17903–17913, 2023. 3
- Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025. 1, 3, 8
- Joanna Waczyńska, Piotr Borycki, Sławomir Tadeja, Jacek Tabor, and Przemysław Spurek. Games: Mesh-based adapting and modification of gaussian splatting. *arXiv preprint arXiv:2402.01459*, 2024. 2
- Chaoyang Wang, Lachlan Ewen MacDonald, Laszlo A Jeni, and Simon Lucey. Flow supervision for deformable nerf. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 21128–21137, 2023a. 3
- Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Int. Conf. Comput. Vis.*, pages 9065–9076, 2023b. 3
- Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *European Conference on Computer Vision*, pages 36–54. Springer, 2024. 7, 8, 9
- Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *Int. Conf. Comput. Vis.*, pages 5610–5619, 2021. 3
- Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 20310–20320, 2024. 1
- Jamie Wynn and Daniyar Turmukhambetov. Diffusionerf: Regularizing neural radiance fields with denoising diffusion models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4180–4189, 2023. 3
- Haolin Xiong, Sairisheek Muttukuru, Rishi Upadhyay, Pradyumna Chari, and Achuta Kadambi. Sparsegs: Real-time 360  $\{\backslash\deg\}$  sparse view synthesis using gaussian splatting. *arXiv preprint arXiv:2312.00206*, 2023. 3
- Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezafofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023. 9
- Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024. 9
- Chongjie Ye, Lingteng Qiu, Xiaodong Gu, Qi Zuo, Yushuang Wu, Zilong Dong, Liefeng Bo, Yuliang Xiu, and Xiaoguang Han. Stablenormal: Reducing diffusion variance for stable and sharp normal. *ACM Transactions on Graphics (TOG)*, 2024. 7, 9
- Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Adv. Neural Inform. Process. Syst.*, 35:25018–25032, 2022. 3
- Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 19447–19456, 2024a. 1
- Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient and compact surface reconstruction in unbounded scenes. *arXiv preprint arXiv:2404.10772*, 2024b. 1, 3, 8
- Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. *arXiv preprint arXiv:2312.00451*, 2023. 3