SG-NeRF: Sparse-Input Generalized Neural Radiance Fields for Novel View Synthesis

Kuo Xu[1], Jie Li[2], Zhenqiang Li[1], Yangjie Cao[1]
[1]Zhengzhou University, [2]Shanghai Jiao Tong University

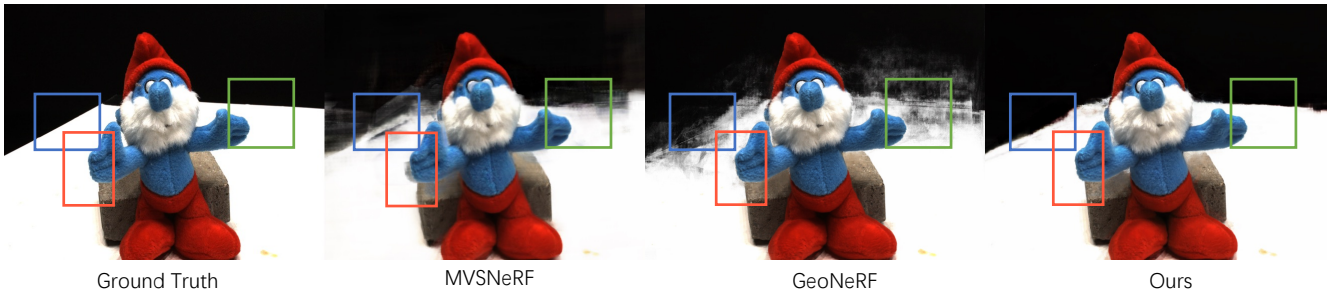

Fig.1. We conducted generalization tests on real scenes from the DTU dataset, without using real depth values. Our model demonstrated superior visual quality compared to existing generalization models, MVSNeRF[6] and GeoNeRF[17].

Abstract Traditional neural radiance fields for rendering novel views require intensive input images and pre-scene optimization, which limits their practical applications. We propose a generalization method to infer scenes from input images and perform high-quality rendering without pre-scene optimization named SG-NeRF. Firstly, we construct an improved multi-view stereo structure based on convolutional attention and multi-level fusion mechanism to obtain the geometric features and appearance features of the scene from the sparse input images, and then these features are aggregated by multi-head attention as the input of the neural radiance field. This strategy of utilizing neural radiance fields to decode scene features instead of mapping positions and orientations enables our method to perform cross-scene training as well as inference, thus enabling neural radiance fields to generalize for novel view synthesis on unseen scenes. We tested the generalization ability on DTU real unseen scenes, and our PSNR improved by 3.14 compared with the baseline method under the same input conditions. In addition, if the scene has dense input views available, the average PSNR can be improved by 1.04 through further refinement training in a short time, and a higher quality rendering effect can be obtained.

Keywords NeRF, MVS, Generalized, New View Synthesis

1 Introduction

Synthesizing novel views from given images has been a hot research topic in the fields of computer vision and computer graphics. This technology is also fundamental for achieving realistic AR/VR experiences. Recently, neural radiance field (NeRF)[26] techniques have gained significant attention due to their impressive rendering quality. NeRF and its subsequent works can achieve photo-realistic rendering of novel views, but they require a large number of images of a single scene as input and involve lengthy optimization processes to obtain accurate radiance fields, which limits their practical applicability.

Recent advancements have addressed these limitations. [39, 32, 20] propose methods that extract 2D features as additional inputs to the radiance field, reducing

the requirement for dense input views. DS-NeRF[10] introduces sparse depth information as additional supervision, improving rendering quality and speeding up the rendering process with fewer training views. DietNeRF[15] introduces semantic consistency loss as an auxiliary task, enabling training with fewer input views for a single scene. MVSNeRF[6] combines multi-view stereo (MVS) geometry with neural radiance fields, enhancing the generalization of the radiance field without the need for per-scene training. However, MVSNeRF cannot handle scene details and occlusions. MVS usually uses convolutional neural networks to extract the information and correlation between multiple views to estimate the depth of the scene. Benefiting from the inductive bias mechanism of convolutional neural networks, MVS can be trained and inferred across scenes and can accurately understand the 3D structure of the scene. MVS' understanding of scene 3D structure is input into NeRF as a priori, which can overcome the disadvantage that NeRF needs to be trained scene by scene, and enable NeRF to complete the task of new view synthesis after one forward propagation in a fully trained pipeline. MVSNeRF[6] has demonstrated the effectiveness of this idea. GeoNeRF[17] improves upon MVSNeRF but relies on supervised training with processed ground truth depth information from the DTU dataset[16] to enhance the performance of the geometric reasoning module. We use GeoNeRF as the baseline for comparison and make several improvements to it.

Specifically, we still adopt the idea of combining multi-view geometry with neural radiance field, so that neural radiance field can be trained and inferred across scenes. The difference is that we improve the module of constructing cost volume in traditional MVS technology, and expand the perceptual interaction between multi-level cost volume by means of multi-level cost volume fusion. More valuable spatial feature information is provided to the neural radiance field. In addition, we propose a deep self-supervised loss, which uses the depth information of MVS inference to distort the source view, reducing the dependence of the generalization model on the true depth information. Instead of the coarse-fine sampling strategy of original NeRF, we use a mixture of Gaussian-uniform sampling to directly utilize the depth information inferred by MVS to sample as many points near the object surface as possible, simplifying the neural radiance field inference rendering process and requiring no additional real depth information.

Our main contributions are as follows:

- Multi-level cost volume fusion module. This fusion module enhances the interaction between cost volume contexts and achieves high-quality geometry perception.
- Feature information decoding module. Decoding features instead of mapping location and orientation enhances the understanding of the scene and the generalization ability of the neural network.
- The structure of scene geometry reasoning and feature decoding enable our model learn to understand the scene from the source view, enable the model to train and reason across scenes.

2 Related Work

2.1 Multi-View Stereo.

Multi-view stereo is a classic problem in computer vision, aiming to recover the dense geometric representation of a scene given multiple views with overlapping regions. Traditional methods [9, 12, 19, 27] have made extensive exploration in solving the multi-view stereo problem. Recent approaches [19, 27, 37] have introduced deep learning techniques to address the MVS

problem. MVSNet[37] builds a cost volume on the scanning planes of the source views and applies 3D convolutional neural networks for post-processing to obtain the depth information of the scene. This approach significantly improves the quality of 3D reconstruction compared to traditional methods. However, the major limitation of this approach is the requirement for a large amount of memory space. R-MVSNet[38] is an improvement over MVSNet by changing the process of regularizing the cost volume from simultaneous regularization at multiple depths to sequential regularization at individual depths, leveraging the output of the previous depth to reduce memory consumption and enhance model scalability. Some methods [7, 13, 36] introduce a cascaded architecture that progressively refines the constructed cost volume, reducing memory consumption and obtaining depths at different scales without sacrificing accuracy. We also utilize such a cascaded architecture, where the initial depth interval of the cost volume is related to the predicted depths from the previous level, enhancing the interaction between different levels of cost volumes.

2.2 Self-attention

Self-attention is a specific implementation of the attention mechanism, introduced by[42]. Fundamentally, self-attention remains focused on addressing the issue of varying points of interest in the input when predicting outputs at different positions in sequence problems. It represents one way to implement the attention mechanism. Initially, the attention mechanism was employed to tackle the issue of polysemy in machine translation. In a given sentence, the words at different positions are not entirely independent; they encompass certain contextual information. The incorporation of an attention layer associates the information of elements at different positions, thereby facilitating information interaction.

In the field of computer vision, the self-attention mechanism is frequently applied in image segmentation to enhance image understanding and processing capabilities [43, 44, 45]. It pays simultaneous attention to both local and global information when processing images. For a set of images, just like words in a sentence, there is an abundance of contextual information between them. This contextual information provides a priori conditions for the accurate synthesis of new perspectives. However, the weight of the information provided by different perspectives for the same position is not entirely the same. The self-attention mechanism offers a theoretical basis for calculating these weights. We utilize the self-attention mechanism to calculate the weight information that neighboring perspectives contribute to a new perspective, achieving high-quality view synthesis. The effectiveness of this mechanism has been confirmed by works such as [6, 17].

2.3 New View synthesis.

In previous works, various methods have been explored for view synthesis, including light field-based approaches [18, 31, 5], image-based rendering techniques [3, 4, 29], and deep learning-based methods [41, 8, 24, 30]. Image-based methods typically learn a blending weight based on ray-space proximity or approximate geometry to perform weighted blending of pixel colors from the source views to generate the colors of the target view. Their synthesis quality relies on the image quality of the source views and is limited by occlusions. Synthesizing radiance fields on meshes [14] or point clouds [1, 23] has the advantage of synthesizing new views using a small set of reference views, but they are often limited by the quality of 3D reconstruction. In the case of non-Lambertian surfaces, the colors of the same point can vary across different views, and this multi-view inconsistency often leads to failure in 3D re-

construction on these surfaces.

Our approach combines traditional MVS techniques with neural rendering techniques by taking spatial features corresponding to sampled points as prior input and decoding colors and densities from scene features corresponding to arbitrary 3D positions. We simulate continuous radiance fields using ray projection techniques and obtain the final pixel colors using volume rendering techniques, enabling realistic view synthesis.

2.4 Neural Scene Representations.

Recently, Ben et al. proposed the use of neural networks to encode scenes as a 5D neural radiance field (NeRF) [26]. NeRF optimizes this neural radiance field to render realistic novel views of a fixed scene. Subsequent works [2, 11, 21] have improved upon NeRF but still require hours or days of optimization per scene. GRF [32] directly takes 2D feature representations of sampled points and ray directions as input, replacing the 3D coordinates in the 5D neural radiance field. PixelNeRF [39] introduced the use of convolutional layers to process the input images and modify the NeRF structure. It incorporates image features as additional inputs, similar to residual connections, allowing the network to be trained across scenes and synthesize new views from a sparse set of images (one or a few). IBRNet [34] proposed a generic interpolation function that aggregates density features of sampled points on the same ray using transformer modules. It requires the input of source view colors and directions and its synthesis quality is limited by the quality of the source views. GNT [33] heavily relies on attention mechanisms to fuse multi-view features and directly predicts the pixel colors of the reference view without volume rendering. GNT [33] uses attention mechanisms to achieve a ray-based learnable scene-adaptive rendering, eliminating the need for per-scene optimization. We believe that the generalization capability of the radiance field mainly stems from the model's inference of the scene. Specifically, the rendering of new views, without per-scene optimization, depends on prior input obtained from the source views, including 3D spatial features and global 2D features. To enhance the model's inference capability, we supervise the geometric reasoning module using the inferred depth and the final rendered depth as pseudo-ground truth values, aiming to construct a more accurate geometric neural field.

3 Method

We train SG-NeRF across scenes and divide the rendering of scenes into two stages.The first stage builds the geometric reasoning module, and the second stage performs scene rendering.Specifically, we first process the 2D features in the channel and spatial dimensions, then use these processed 2D features to construct the cost volume, and then fuse the cost volume as the 3D prior information of the reference view. We describe the geometric reasoning module in detail in Section 3.1. In the second phase, as the rendering phase, we use the NeRF network to build a decoding module that uses the 3D features of the first phase as an additional prior guide to predict the density and color information of the spatial sampling points. At the same time, we use the rough depth information predicted in the first stage for fine sampling, avoiding the additional time consumption caused by NeRF hierarchical sampling. We describe this sampling method in 3.2.1 and our decoding module in 3.2.2. The overall flow chart is shown in Fig.2.

3.1 Geometric reasoning

3.1.1 Building Cost Volumes

Given N adjacent source images $\{\mathrm{I_i}\}_{i=1}^{\mathbf{N}} \in \mathbb{R}^{3\times \mathrm{H}\times \mathrm{W}}$, we first extract multi-level feature information from the

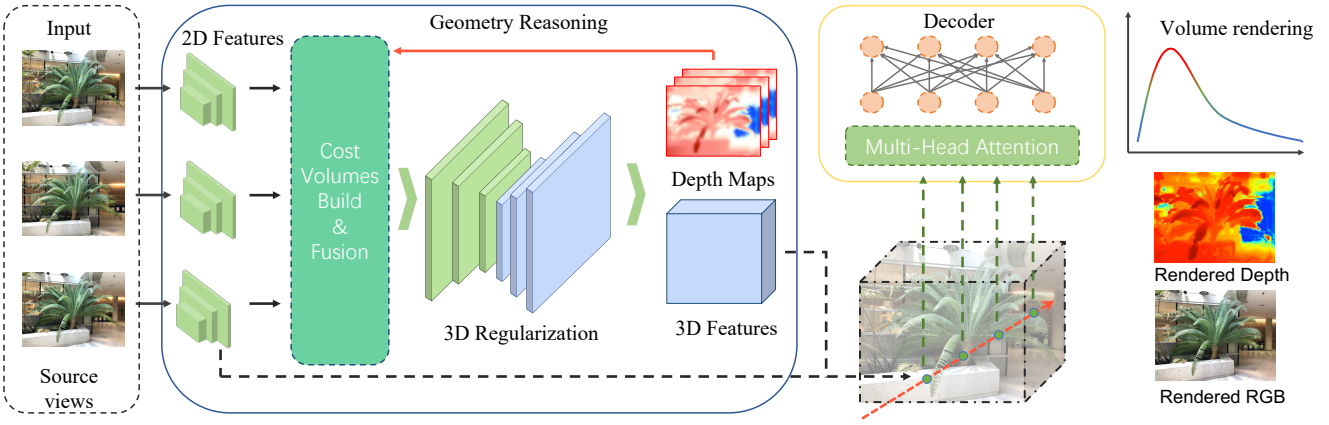

Fig.2. The entire inference pipeline can be summarized as follows: For the target reference view rendering, we first select N neighboring views based on camera parameters and input them into a geometric reasoning model as source views. A U-Net with convolutional attention modules is employed to extract multi-level 2D features from these source views, which are used for constructing and fusing multi-level cost volumes. Next, at level l, the cost volumes of the source views are regularized to obtain predicted depth maps and 3D features F_i^l for each source view. These predicted depth maps are used to guide the construction of the next-level cost volumes and ray sampling. Finally, the multi-level 3D features corresponding to the spatial sampling points, along with the full-resolution 2D features, are fed into the decoding module. Through a multi-head attention mechanism, the feature information from different source views is aggregated and separately passed to the color decoding network and density decoding network for decoding.

images to construct a cost volume for the source view [17]. In MVS [37], this multi-scale structure is more helpful for inferring scene depth information, and previous work [21, 32] has demonstrated the effectiveness of multi-scale structures. For the extracted 2D scene features, we use a convolutional block attention module to enhance the scene-related features and suppress irrelevant features, reducing the loss of detail in small-scale feature information. Specifically, we improve the model's representational capacity by adaptively adjusting the weights of different channel and spatial position feature maps.

First, we process the channel dimension of the feature maps using a channel attention module, selectively enhancing the representation ability of each channel. The specific implementation can be divided into the following steps: We begin by performing global max pooling and global average pooling on the feature map channels to obtain two channel value pooling weights W_{C_1} and W_{C_2}, respectively. Then, we feed these two weights into a shared neural network, obtaining the weight coefficients W_C for the channel dimension, performing dot product operation with the original feature map to obtain channel-weighted feature map f_c^l.

$$\begin{aligned} \mathrm{W}_C &= \mathrm{MLP}(\mathrm{W}_{C_1}, \mathrm{W}_{C_2}) \\ f_c^l &= \mathrm{W}_C \otimes f_i^l \end{aligned} \tag{1}$$

After obtaining the channel-weighted feature f_c^l, we also obtain two spatial pooling weights W_{S_1} and W_{S_2} by performing global max pooling and global average pooling on the spatial dimension. We concatenate the two weights obtaining the weight coefficients W_S for the spatial dimension. We use these two weights to process the channel-weighted feature map, obtaining multi-scale features $\hat{f}_i^l$ based on convolutional attention.

$$\begin{aligned} \mathrm{W}_S &= \mathrm{CNN}(\mathrm{W}_{S_1}, \mathrm{W}_{S_2}) \\ \hat{f}_i^l &= \mathrm{W}_S \otimes f_c^l \end{aligned} \tag{2}$$

This attention-based multi-scale feature approach helps to aggregate more valuable feature information into the cost volumes, thereby enabling the geometric neural field to provide more valuable local spatial features.

We also adopted the cascaded cost volume construction proposed by CastMVSNet. By using the camera

parameters $[K, R, t]$, we can find the K nearest views to view I_i among the N source views and perform homography warping to obtain the multi-level cost volume V_i^l based on group correlation for the source view I_i.

$$\mathcal{H}_k(z) = K_k \cdot \left(R_k \cdot R_i^T + \frac{(t_i - t_k) \cdot n_i^T}{z} \right) \cdot K_i^{-1} \quad (3)$$

$$\mathrm{V}_i^l(u, v, z) = \mathbf{G}(\hat{f}_k^l \cdot (\mathcal{H}_k(z) \cdot [u, v, 1]^T)_k^{\mathrm{K}} \quad (4)$$

In the equation, $\mathcal{H}_k(z)$ represents the homography matrix that warps the k-th image to the reference view I_i. (u, v, z) represents the spatial coordinates of a point in 3D space. $\mathbf{G}(\cdot)$ computes the group correlation among the K images.

3.1.2 *Geometric Neural Field*

This group-based cost volume encodes the appearance of the scene from different input viewpoints, capturing the variations in appearance caused by geometry and viewpoint changes. During the construction of the scene cost volume, the connections between different-level cost volumes are established based on the relationships between the feature maps extracted by the U-net at different levels. To strengthen the interaction between cost volumes at different levels, we propose a cost volume fusion module that integrates the current cost volume with the cost volume from the previous level. The small-scale cost volume is composed of the scene features with large receptive field, which often contains more abstract spatial information of the scene, but it is easy to ignore some details of the scene.Based on this reason, we adopt a method similar to UNet to fuse large-scale cost volume with small-scale cost volume, and enhance the perception ability of small-scale cost volume to scene details. Specifically, we first perform trilinear interpolation on the cost volume from the previous level to match the width, height, and depth dimensions of the current level's cost volume. Then, we use a convolutional layer to adjust the channel dimensions to match the current level's cost volume, ensuring consistency in size. Finally, the cost volumes from the previous level and the current level are concatenated and fused. This cost volume fusion module is illustrated in Fig.3.

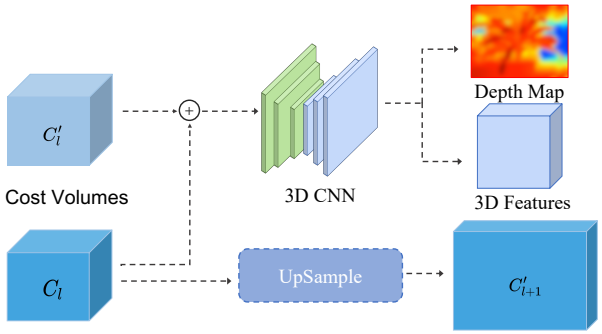

Fig.3. Before regularization, the cost volume C_l is first concatenated and fused with the upsampled cost volume C_l' from the previous level. Afterward, regularization is performed to obtain the predicted depth map and scene 3D features. Additionally, the cost volume C_l is upsampled to generate C_{l+1}', which serves as the input for regularization in the next level.

In the cost volume regularization stage, the traditional MVS method directly predicts the depth information of the scene and only interprets the scene geometry. Our aim is to perceive scene geometry versus appearance across scenes, so different from traditional MVS is to generate meaningful geometric neural fields F_i^l while inferential scene depth. The geometric neural field F_i^l is input into the subsequent decoding network for decoding as the geometric understanding of the scene. The inferential scene depth is used for the sampling prior of subsequent ray-casting steps of neural radiation fields. Our geometric reasoning module does not use the real depth information to supervise, in order to constrain it, we use the predicted depth and camera parameters to distort K neighboring views of the source view I_i, and calculate the photometric consistency loss between the source view and the distorted neighboring view.

3.2 Feature Decoding

3.2.1 Gaussian-Uniform mixture sampling

After constructing the geometric neural field, we use ray casting techniques to render new views. We simulate N_r rays based on the camera parameters of the reference view I_0 and sample discrete points along the rays for rendering the final ray colors. To enhance the correlation between the sampling point positions and the spatial depth, we use inverse warp to distort the predicted depth maps from the source views to the reference view. By performing this inverse warp, smaller distorted depth values are overlaid on top of larger distorted depth values, resulting in a fused depth map for the reference view. This fused depth map serves as the coarse predicted depth $\hat{D}$, providing a prior guidance for the fine sampling of the point positions.

First, we uniformly sample S_c points along each camera ray to cover the entire depth range.

$$t_k \sim \mathcal{U}\left[t_n + \frac{k-1}{K}\left(t_f - t_n\right), t_n + \frac{k}{K}\left(t_f - t_n\right)\right] \quad (5)$$

In the equation, t_f and t_n represent the far and near boundaries of the scene, t_k represents the sampling interval for ray casting.

Subsequently, guided by the coarse predicted depth, we sample candidate points following a Gaussian distribution. These candidate points are sampled in a way that takes into account the estimated depth information, allowing us to capture the variations in scene geometry more effectively. Assuming the pixel coordinate is denoted as $P = (u, v)$ and the predicted ray depth is denoted as $z_p = \hat{D}(u, v)$, sample S_f fine sample points using the following formula:

$$\begin{aligned} t_k &\sim \mathcal{N}(z_p, s_p^2) \\ s_p &= \frac{min(|z_p - t_f|, |z_p - t_n|)}{3} \end{aligned} \quad (6)$$

z_p and s_p are the mean and standard deviation of the proposed normal distribution, respectively. By using these values, we can optimize geometric features more effectively by sampling more candidate points near the object surface. This differentiable sampling method also contributes to better convergence of the geometric neural field.

3.2.2 Aggregation feature and Decoding

Using the camera parameters, each point X on the ray is projected onto each source view and bilinear interpolation is performed to obtain corresponding multi-level 3D features. These multi-level features are then merged to form the final geometric field feature F_i. As for the scene's 2D features, only the full-size 2D feature with $l = 0$ is retained and subjected to bilinear interpolation. This full-size 2D feature encompasses a global understanding of the scene and also provides a mask to determine if the sampling point is projected outside the source view. Finally, these aggregated feature values $\hat{F}_i$ serve as the input to the decoding module.

$$\hat{F}_i = [\{F_i^l\}_{l=0}^2, \{f_i^l\}_{l=0}] \quad (7)$$

Now that the vector $\hat{F}_i$ contains all the necessary data of the scene, it can be used to learn the scene appearance and predict the density of spatial sampling points. NeRF employs a fully connected neural network (MLP) to map the coordinate vector and direction vector of a spatial point to its corresponding color and density, resulting in overfitting to the specific scene. This overfitting restricts NeRF's ability to train and render across different scenes. In contrast, SG-NeRF constructs the geometric neural field of the scene during the scene feature inference stage, using the bias-inductive power of convolutional neural networks for cross-scene training. After obtaining the scene feature vector $\hat{F}_i$, we first compute the mean and variance of the full-sized 2D features as the view-independent token. Contains the 3D features and 2D features of the

source view, serves as the view-dependent tokens. Since the contribution of different source views to the new view is not the same, we use the multi-head attention mechanism proposed in Transformer to aggregate the tokens from different views and obtain the attention-weighted feature from different views. The effectiveness of this mechanism has been demonstrated by [17]. Once the scene feature vector is obtained, two separate feature decoding networks (MLPs) are used to decode the color and density of the spatial sampling points. The color decoding network takes the view-dependent vector as input, while the density decoding network takes the global view-independent vector token as input.

$$\boldsymbol{c_n}, \boldsymbol{\sigma_n} = \mathrm{MLP}(\mathrm{MHA}((\mathrm{mean}(f_i^0), \mathrm{var}(f_i^0))_{i=1}^{\mathrm{N}}, \hat{F}_i), \triangle d) \tag{8}$$

Note that to enhance the mapping ability of the decoding network, we combine the relative direction vectors of the sampling point $\triangle d$ as residual items, and the encoding method of the direction vector is consistent with that in NeRF [26]. After decoding the density and color of the space sampling point, we use traditional volume rendering techniques [22] to render the color and depth values of the ray.

$$\hat{\boldsymbol{c}} = \sum_{n=1}^{S} \exp\left(-\sum_{k=1}^{n-1} \boldsymbol{\sigma}_k\right) \left(1 - \exp\left(-\boldsymbol{\sigma_n}\right)\right) \boldsymbol{c_n} \tag{9}$$

The formula for rendering the color is slightly modified to obtain the depth value of the ray.

$$\hat{\boldsymbol{d}} = \sum_{n=1}^{S} \exp\left(-\sum_{k=1}^{n-1} \boldsymbol{\sigma_k}\right) \left(1 - \exp\left(-\boldsymbol{\sigma_n}\right)\right) z_n \tag{10}$$

3.3 Loss Function

For the color loss, we follow the same approach as the original NeRF. We calculate the mean squared error to measure the difference between the rendered color and the true pixel color.

$$\mathcal{L}_c = \frac{1}{|R|} \sum_{r \in R} \left\| \hat{\boldsymbol{c}}(r) - c_{gt}(r) \right\|^2 \tag{11}$$

Using only the final rendered color loss for supervision is insufficient to constrain the entire pipeline of geometric reasoning. Therefore, we propose the photometric consistency loss of the self-supervised module as well as the deep inference loss to supervise the geometric inference module. Specifically, during the geometric neural field inference stage, in addition to obtaining the scene's 3D features, we also obtain the inferred depth map of the scene. We use this depth map to perform an inverse warp of the neighboring views of the source view. This process generates the warped source view and a binary mask, M, which masks out invalid pixels outside the source view. The photometric consistency loss is then calculated by comparing the differences between the warped view and the real source view:

$$\begin{aligned} \mathcal{L}_{PC} = \sum_{j=1}^{K} \frac{1}{\|M_j\|_i} \Big(& \left\| (\hat{I}_i^j - I_i) \odot M_j \right\|_2 + \\ & \left\| (\nabla \hat{I}_i^j - \nabla I_i) \odot M_j \right\|_2 \Big) \end{aligned} \tag{12}$$

In the equations, the symbol ∇ represents the pixel-wise gradient, and $\odot$ represents the pixel-wise multiplication. This loss measures the consistency of pixel intensities between the warped and real views.Additionally, we utilize the fused depth from the sampling stage of the reference view, which determines the efficiency of our fine sampling. To enhance the consistency between the distorted source view and the reference view in terms of depth, we use the rendered depth as the pseudo-real depth value for the reference view and warp this depth value to each source view. We minimize the difference between the predicted depth in the source view and the pseudo-depth value in the op-

Table 1. We tested SG-NeRF and other generalization models on the unseen scenes from the DTU[16], LLFF[25], and Synthetic datasets[26]. We used three metrics for qualitative comparison: PSNR (higher is better), SSIM (higher is better), and LPIPS (lower is better). Bold indicates the **best results**, and underlining represents the second best results.

Method	Settings	Real Data (DTU / LLFF)			Synthetic Data		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
pixelNeRF[39]	No per-scene Optimization	19.31/11.16	0.789/0.486	0.382/0.671	7.39	0.658	0.411
IBRNet [34]		20.01/23.38	0.803/0.789	0.347/0.229	25.11	0.902	0.108
MVSNeRF[6]		20.10/20.30	0.812/0.726	0.338/0.317	23.62	0.897	0.176
GeoNeRF[17]		21.77/25.00	0.847/0.823	0.217/0.183	**28.14**	**0.936**	**0.090**
Ours		**24.91/25.21**	**0.891/0.836**	**0.195/0.173**	27.79	0.926	0.119
NeRF[21]	Per-scene Optimization	**27.01/25.97**	**0.901**/0.870	0.263/0.236	**30.63**	**0.962**	0.093
MVSNeRF[6]		21.97/25.45	0.847/**0.877**	0.226/0.192	27.07	0.931	0.168
GeoNeRF[17]		23.78/25.81	0.897/0.841	**0.176**/0.173	28.94	0.941	0.077
Ours		25.80/25.85	0.898/0.853	0.188/**0.156**	28.91	0.943	**0.070**

timization process.

$$\mathcal{L}_{DC} = smooth_{L_1}(\hat{D}(r) - z(r)) \tag{13}$$

Here, $\hat{D}(r)$ represents the depth value obtained from volume rendering, and $z(r)$ represents the depth map from the convolutional operation on the cost volume. The final loss formulation can be represented as follows:

$$L = \mathcal{L}_c + \lambda_{pc} \sum_{i=1}^{N} \mathcal{L}_{PC}/N + \lambda_{dc} \times \mathcal{L}_{DC} \tag{14}$$

λ_{pc} and λ_{dc} are weighting factors that balance the influence of each loss term in the overall optimization process.

4 Experiment

Datasets. We trained our model on real forward-facing datasets from LLFF[25], IBRNet[34], and the DTU dataset[16]. The camera parameters for the real forward-facing scenes were obtained from the COLMAP[28] . In total, there were 5689 images used for training, which came from 102 indoor and outdoor forward-facing scenes (35 scenes from LLFF and 67 scenes from IBRNet), as well as 88 real DTU scenes. Unlike GeoNeRF, we did not utilize real depth data from DTU for training. Instead, we relied solely on RGB images for self-supervision.We conducted testing on a subset of the LLFF dataset, which consists of 8 real-world forward-facing scenes, as well as 8 synthetic scenes. Additionally, we performed testing on 15 real scenes from the DTU dataset. We also conducted fine-tuning and testing on these datasets to further improve the performance of our model.

During training, we randomly selected one image as the reference view and simulated partial rays based on its camera parameters. To conserve memory, we resized the resolution of each image to 640×480.

Implementation Details. We trained SG-NeRF for 40 epochs, where each epoch involved iterating through all training views. During training, we randomly selected one image from the training dataset as the reference view. From the reference view, we emitted 512 rays and sampled 128 points along each ray, including 96 coarse samples and 32 fine samples. We trained our code on a single 3090ti GPU. The initial training for cross-scene initialization took approximately 4 days. Once training was completed, there was no need to train for each scene separately. A single forward pass was sufficient to synthesize the reference view from the source views.

For each epoch, we used the Adam optimizer with an initial learning rate of 5e-4. We employed the ReduceLROnPlateau learning rate strategy, dynamically

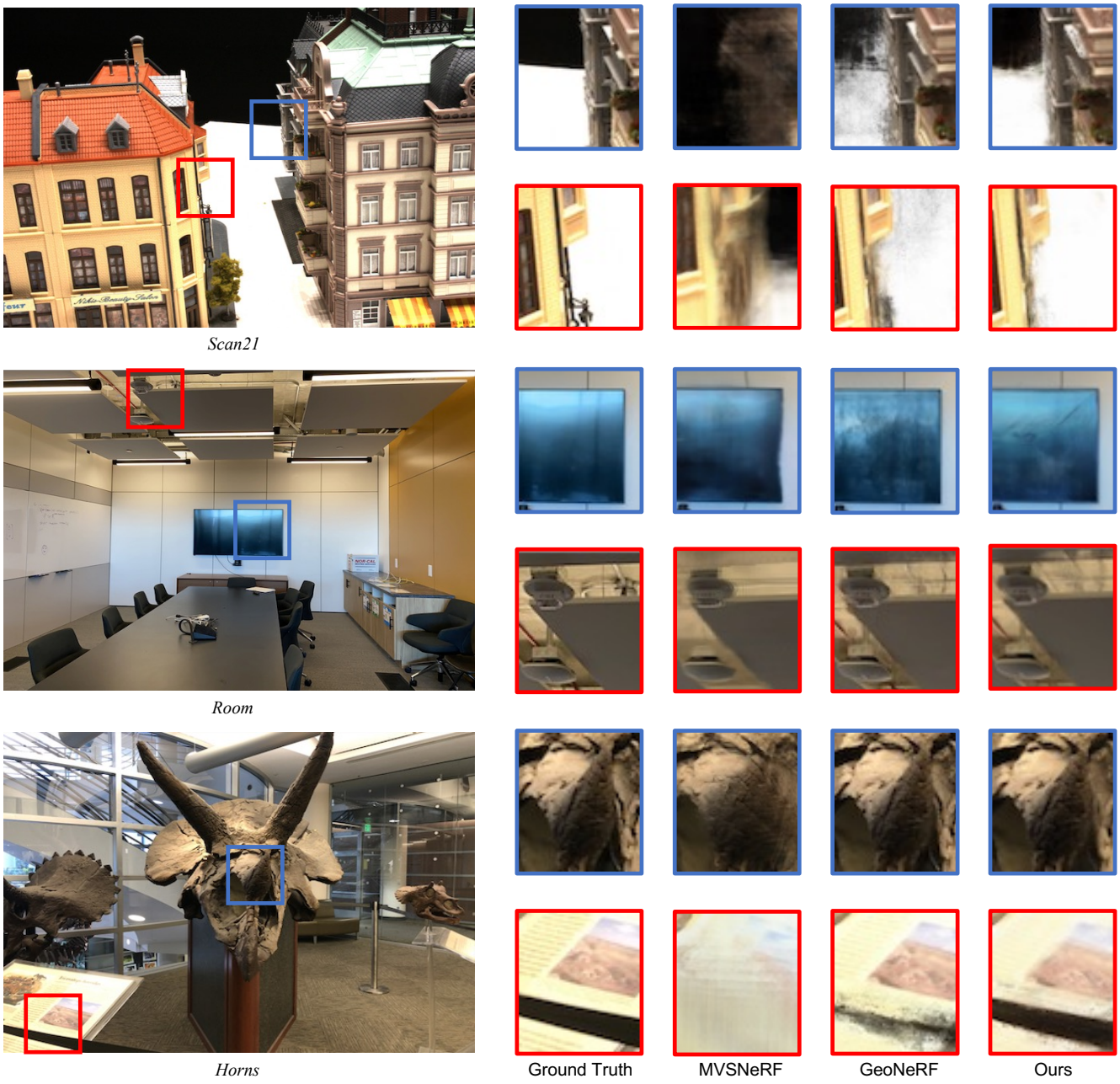

Fig. 4. We test the generalization synthesis effect of our model on LLFF (*Room*, *Horns*) and DTU (*scan21*) datasets. When we performed the generalization test on the DTU dataset, we did not utilize the processed depth information in the DTU dataset to verify the generalization ability of our model under the low-information condition. Compared with other generalization models, our model performs better in detail and alleviates the artifacts in weak texture regions.

adjusting the learning rate based on the average PSNR obtained from each epoch.

4.1 Experimental Results

To evaluate the generalization ability of our model, we compared it with the original NeRF and other well-known open-source generalized NeRF models: PixelNeRF[39], IBRNet[34], MVSNeRF[6], and GeoNeRF[17]. In the generalization capability tests, we primarily use RGB images as the original input, without the need to include the corresponding depth information of the scene. This approach is based on our ob-

servation that the purpose of introducing Multi-View Stereo (MVS) technology is to enable the network to infer the 3D information of the scene. Using depth information as an input undermines the effectiveness of this module and limits the practical application of the model. The original experimental results of GeoNeRF, which exhibited performance jumps on both synthetic datasets and the DTU dataset (where depth information was inputted), highlight this issue. Conversely, by integrating the depth information predicted from source views as supplementary input for the reference view, we have demonstrated that our module functions effectively even without depth inputs. As shown in Table 1., we tested these models on an unseen test dataset and quantitatively compared them based on PSNR, SSIM [35], and LPIPS[40] metrics. The results indicate that our model outperforms the others in terms of performance. When tested on the DTU dataset without utilizing real depth information, our model performs the best, demonstrating the effectiveness of the geometric neural field. Fig.1. and Fig.4. showcases the rendering results in unseen scenes, where our model better preserves scene details and exhibits fewer artifacts compared to others.

In order to test the ability of our generalization model to synthesize novel views when dense views are available, we refine the training on a specific scene from the NeRF synthetic dataset and compare it with the original NeRF. Our results show that our method achieves comparable results to the original NeRF in a short amount of time with refined training. Compared with other generalization models, our model shows the best performance after refined training, second only to the original NeRF with full training. Figu.5. shows the optimized rendering results of NeRF for each scene and the results of SG-NeRF's refined training.

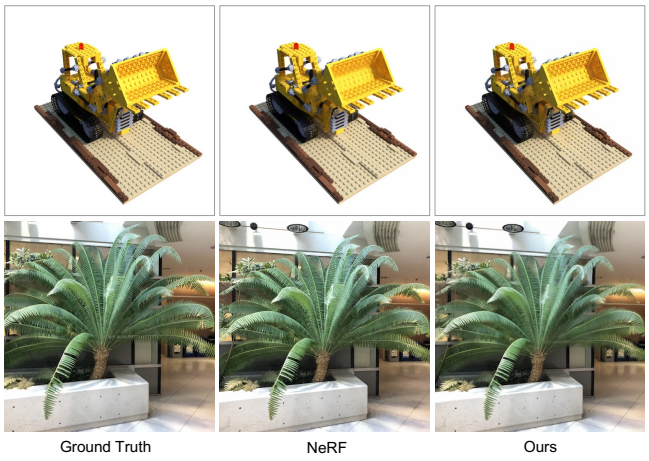

Fig.5. We showcased the synthesis results of our model on new views after short fine-tuning, achieving performance comparable to the original NeRF model.

4.2 Ablation Study

Fig.5. shows the synthesis results in our generalization model retained for more details on the scene, proves that we improve the effectiveness of the geometry of neural field. To prove the validity of the other modules in the model, we conducted an ablation study of our generalized model on the LLFF dataset.Table 2. shows our ablation results, which include: (a) no self-supervised loss is used to constrain the geometric neural field, (b) only the points on the line are uniformly sampled, (c) the attention mechanism of the decoding module is removed.

It can be seen from the results in the table that the model can show the best effect when the improved module is fully used.(a) When self-supervised loss is not used, the geometric inference module lacks constraints on spatial geometry at the beginning of the process, which causes our convergence process to slow down further and affects our final synthesis quality.It can be observed from Fig.6. that the predicted depth without self-supervised depth loss is prone to blur, detail loss and other problems. (b) When we do not use Gaussian uniform mixture sampling, we do the equivalent of just performing the coarse sampling phase of

NeRF without sampling more points on the object surface, which causes the resultant new view to lose some detail. (c) Remove the attention mechanism, which has the biggest impact on the model.The lack of attention mechanism causes all source views to provide equivalent features, whereas for different new perspectives, the source view with the closest perspective should provide more weighted features. In Fig.7., we show the results of the new view synthesis after removing the attention mechanism, and the overall quality degradation can be clearly observed, including large deviations in color information and false artifacts.

Table 2. Ablation study of the key components of SG-NeRF. The evaluation is performed on the real forward-facing LLFF

Settings	LLFF Data		
	PSNR ↑	SSIM ↑	LPIPS ↓
a.No self-supervised loss	24.33	0.821	0.186
b.No mixture sampling	24.84	0.826	0.184
c.No attention mechanism	17.70	0.620	0.375
d.Full SG-NeRF	**25.21**	**0.836**	**0.173**

Table 3. Quantitative analysis of different numbers of source views on LLFF dataset.

Number of source views	PSNR	SSIM	LPIPS
4	20.92	0.786	0.205
5	23.52	0.818	0.176
6	25.21	0.836	0.173

Table 4. Quantitative analysis of skipping the nearest K neighboring views on LLFF dataset.

K	0	2	3	4
PSNR	25.21	23.48	23.27	22.85
SSIM	0.836	0.792	0.782	0.770
LPIPS	0.173	0.218	0.229	0.240

Quantitative analysis of skipping the nearest K neighboring views on LLFF dataset.

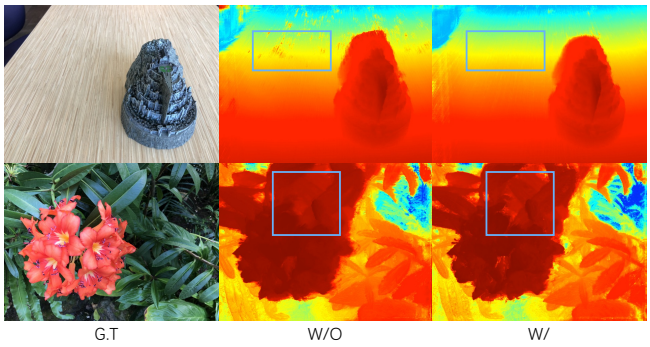

Fig.6. New view synthesis results after removing depth loss.

Fig.7. New view synthesis results after removing the attention mechanism. A significant mass loss can be observed.

4.3 The Influence of Source View Count

We investigated the impact of the number and quality of source views on our model to analyze its robustness to source views. As shown in Table 3. , we evaluated the influence of different numbers of source views on our model. The results demonstrate that even with a small number of source views, our model can still synthesize realistic new viewpoint images.

Table 4. showcases the robustness of our model when there is a significant viewpoint difference between the source views and the reference view. We discarded the K nearest neighboring views to the reference view and used the remaining neighboring views as source views for rendering. As the viewpoint difference between the source views and the reference view increases, the effective information provided by the source views

decreases. In this scenario, our model does not exhibit a significant performance drop.

5 Conclusion

We introduce SG-NeRF, a few-view novel view synthesis method that can render realistic novel views for complex scenes without per-scene optimization. Our approach enhances the performance of traditional multi-view geometry architectures using convolutional attention modules and a cost volume fusion mechanism. It constructs a geometric neural field for scene representation and assists the neural network inferring the scene. Multi-head Attention is used to aggregate information from the source views, enabling the synthesis of realistic images from new viewpoints. We believe that more advanced multi-view stereo geometry techniques may extend the application of our method to surround-shooting source views and reduce artifacts in weakly textured regions.

References

[1] K. A. Aliev, A. Sevastopolsky, M. Kolos, D. Ulyanov, and V. Lempitsky. Neural point-based graphics. In *European Conference on Computer Vision*, 2020.

[2] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan. Mip-nerf: A mul- tiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pages 5855-5864.

[3] G. Chaurasia, S. Duchene, O. Sorkine-Hornung, and G. Drettakis. Depth synthesis and local warps for plausi- ble image-based navigation. *Acm Transactions on Graphics*, 2013, 32(3):1-12.

[4] G. Chaurasia, O. Sorkine, and G. Drettakis. Silhouette-aware warping for image-based rendering. *Computer Graphics Fo- rum*, 30(4):1223-1232, 2011.

[5] A. Chen, M. Wu, Y. Zhang, N. Li, J. Lu, S. Gao, and J. Yu. Deep surface light fields. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 2018.

[6] A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, and H. Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. 2021.

[7] S. Cheng, Z. Xu, S. Zhu, Z. Li, L. E. Li, R. Ramamoorthi, and H. Su. Deep stereo using adaptive thin volume representation with uncertainty awareness. 2019.

[8] I. Choi, O. Gallo, A. Troccoli, M. H. Kim, and J. Kautz. Ex- treme view synthesis. In *International Conference on Com- puter Vision*.

[9] J. S. De and P. Viola. Poxels : Probabilistic voxelized volume reconstruction. 1999.

[10] K. Deng, A. Liu, J. Y. Zhu, and D. Ramanan. Depth- supervised nerf: Fewer views and faster training for free. 2021.

[11] T. DeVries, M. A. Bautista, N. Srivastava, G. W. Taylor, and J. M. Susskind. Unconstrained scene generation with locally conditioned radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14304- 14313, 2021.

[12] Y. Furukawa and J. Ponce. Accurate, dense, and robust mul- tiview stereopsis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(8):p.1362-1376, 2010.

[13] X. Gu, Z. Fan, S. Zhu, Z. Dai, and P. Tan. Cascade cost vol- ume for high-resolution multi-view stereo and stereo match- ing. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[14] J. Huang, J. Thies, A. Dai, A. Kundu, and T. Funkhouser. Adversarial texture optimization from rgb-d scans. 2020.

[15] A. Jain, M. Tancik, and P. Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. 2021.

[16] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and H. Aans. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 406-413, 2014.

[17] M. M. Johari, Y. Lepoittevin, and F. Fleuret. Geonerf: Gen- eralizing nerf with geometry priors. 2021.

[18] N. K. Kalantari, T. C. Wang, and R. Ramamoorthi. Learning- based view synthesis for light field cameras, 2016.

[19] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. *Springer-Verlag*, 2002.

[20] J. Li, Z. Feng, Q. She, H. Ding, C. Wang, and G. H. Lee. Mine: Towards continuous depth mpi with nerf for novel view synthesis. 2021.

[21] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Pro- ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210-7219, 2021.

[22] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graph- ics*, 1(2):99-108, 1995.

[23] M. Meshry, D. B. Goldman, S. Khamis, H. Hoppe, R. Pandey, N. Snavely, and R. Martin-Brualla. Neural rerendering in the wild. *IEEE*, 2019.

[24] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalan- tari, R. Ramamoorthi, R. Ng, and A. Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics*, 38(4):1-14, 2019.

[25] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalan- tari, R. Ramamoorthi, R. Ng, and A. Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):1- 14, 2019.

[26] 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. 2020.

[27] J. L. Schnberger, E. Zheng, M. Pollefeys, and J. M. Frahm. Pixelwise view selection for unstructured multi-view stereo. *Springer, Cham*, 2016.

[28] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104-4113, 2016.

[29] S. N. Sinha, D. Steedly, and R. Szeliski. Piecewise planar stereo for image-based rendering. In *IEEE 12th International Conference on Computer Vision*, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009, 2009.

[30] P. P. Srinivasan, R. Tucker, J. T. Barron, R. Ramamoorthi, R. Ng, and N. Snavely. Pushing the boundaries of view extrapolation with multiplane images. *IEEE*, 2019.

[31] P. P. Srinivasan, T. Wang, A. Sreelal, R. Ramamoorthi, and R. Ng. Learning to synthesize a 4d rgbd light field from a single image. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[32] A. Trevithick and B. Yang. Grf: Learning a general radiance field for 3d representation and rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15182-15192, 2021.

[33] P. Wang, X. Chen, T. Chen, S. Venugopalan, Z. Wang, et al. Is attention all nerf needs? *arXiv preprint* arXiv:2207.13298, 2022.

[34] Q. Wang, Z. Wang, K. Genova, P. P. Srinivasan, H. Zhou, J. T. Barron, R. Martin-Brualla, N. Snavely, and T. Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690-4699, 2021.

[35] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simon- celli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image process- ing*, 13(4):600-612, 2004.

[36] J. Yang, W. Mao, J. M. Alvarez, and M. Liu. Cost volume pyramid based depth inference for multi-view stereo. In *2020 IEEE/CVF Conference on Computer Vision and Pat- tern Recognition (CVPR)*, 2020.

[37] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan. Mvsnet: Depth inference for unstructured multi-view stereo. *European Con- ference on Computer Vision (ECCV)*, 2018.

[38] Y. Yao, Z. Luo, S. Li, T. Shen, T. Fang, and L. Quan. Recur- rent mvsnet for high-resolution multi-view stereo depth in- ference. *Computer Vision and Pattern Recognition (CVPR)*, 2019.

[39] A. Yu, V. Ye, M. Tancik, and A. Kanazawa. pixelnerf: Neural radiance fields from one or few images. 2020.

[40] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a percep- tual metric. In *Proceedings of the IEEE conference on com- puter vision and pattern recognition*, pages 586-595, 2018.

[41] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Transactions on Graphics*, 37(4):1-12, 2018.

[42] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. *Advances in neural information processing systems*, 2017.

[43] Guo M H, Xu T X, Liu J J, et al. Attention mechanisms in computer vision: A survey. *Computational visual media*, 8(3): 331-368, 2022.

[44] Shmatko A, Ghaffari Laleh N, Gerstung M, et al. Artificial intelligence in histopathology: enhancing cancer research and clinical oncology. *Nature cancer*, 3(9): 1026-1038, 2022.

[45] Li Y, Mao H, Girshick R, et al. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision (ECCV)*, 2022: 280-296.