GRF: LEARNING A GENERAL RADIANCE FIELD FOR 3D SCENE REPRESENTATION AND RENDERING

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a simple yet powerful implicit neural function that can represent and render arbitrarily complex 3D scenes in a single network only from 2D observations. The function models 3D scenes as a general radiance field, which takes a set of posed 2D images as input, constructs an internal representation for each 3D point of the scene, and renders the corresponding appearance and geometry of any 3D point viewing from an arbitrary angle. The key to our approach is to explicitly integrate the principle of multi-view geometry to obtain the internal representations from observed 2D views, guaranteeing the learned implicit representations meaningful and multi-view consistent. In addition, we introduce an effective neural module to learn general features for each pixel in 2D images, allowing the constructed internal 3D representations to be remarkably general as well. Extensive experiments demonstrate the superiority of our approach.

1 INTRODUCTION

Understanding the precise 3D structure of a real-world environment and realistically re-rendering it from free viewpoints is a key enabler for many critical tasks, ranging from robotic manipulation and navigation to augmented reality. Classic approaches to recover the 3D scene geometry mainly include the structure from motion (SfM) (Ozyesil et al., 2017) and simultaneous localization and mapping (SLAM) (Cadena et al., 2016) pipelines. However, they are limited to reconstruct sparse and discrete 3D point clouds which are unable to contain geometric details.

The recent advances in deep neural networks have yielded rapid progress in 3D modeling and understanding. Most of them focus on the explicit 3D shape representations such as voxel grids (Choy et al., 2016), point clouds (Fan et al., 2017), or triangle meshes (Wang et al., 2018). However, these representations are discrete and sparse, limiting the recovered 3D structures to extremely low spatial resolution. In addition, these networks usually require large-scale 3D shapes for supervision, resulting in the trained models being over-fitting particular datasets and unable to generalize to novel scenes. In fact, it is also costly and even infeasible to collect high-quality 3D labels.

Encoding geometries into multilayer perceptrons (MLPs) (Mescheder et al., 2019; Park et al., 2019) recently emerges as a promising direction in 3D reconstruction and understanding. Its key advantage is the ability to model 3D structures in a continuous way instead of discrete, and therefore it has the potential to achieve unlimited spatial resolution in theory. However, most methods of this pipeline focus on individual objects and require ground truth 3D geometry for supervision. By introducing a recurrent neural network based renderer, SRNs (Sitzmann et al., 2019) is among the early work to learn implicit surface representations only from posed 2D images, but it fails to capture complicated scene geometries and renders over-smoothed images. Alternatively, by leveraging the volume rendering to synthesize new views for 2D supervision, the very recent NeRF (Mildenhall et al., 2020) directly encodes the radiance fields of complex 3D scenes within the weights of MLPs, achieving an unprecedented level of fidelity for challenging 3D scenes. Nevertheless, it has two major limitations: 1) since each 3D scene is encoded into all the weights of MLPs, the trained network (*i.e.,* a learned radiance field) can only represent a single scene, and is unable to generalize across novel scenarios; 2) because the shape and appearance of each spatial 3D location along a light ray is only optimized by the available pixel RGBs, the learned implicit representations of that location are lack of the general geometric patterns, resulting in the synthesized images being less photo-realistic.

In this paper, we propose a **g**eneral **r**adiance **f**ield (GRF), a simple yet powerful implicit function that can represent and render arbitrarily complex 3D scenes. Our GRF takes a set of posed 2D images, a 3D query point and its query viewpoint as input, and predicts the RGB value and volumetric density of that query point. Basically, this neural function learns to represent a 3D scene from sparse 2D observations, and infers the shape and appearance of that scene from previously unobserved viewpoints. Note that, the inferred shape and appearance of any particular 3D query point explicitly take into account its local geometric patterns from the available 2D observations. In particular, the proposed GRF consists of four components:

- Extracting the general 2D visual features for every light ray from the input 2D observations;
- Reprojecting the corresponding 2D features back to the query 3D point using the principle of multi-view geometry;
- Selecting and aggregating all the reprojected features for the query 3D point, while the visual occlusions are implicitly considered;
- Rendering the aggregated features of the query 3D point along a particular query viewpoint, and producing the corresponding RGB and volumetric density.

These four components enable our GRF to be significantly different from all existing 3D scene representation approaches. 1) Compared with the classic SfM/SLAM systems, our GRF can represent the 3D scene structure with smooth and continuous surfaces. 2) Compared with the neural approaches based on explicit voxel grids, point clouds and meshes, our GRF learns continuous 3D representations without requiring 3D data for training. 3) Compared with the existing implicit representation methods such as SDF (Park et al., 2019), SRNs (Sitzmann et al., 2019) and NeRF (Mildenhall et al., 2020), our GRF can represent arbitrarily complicated 3D scenes and has remarkable generalization to novel scenarios. In addition, the learned 3D representations carefully consider the general geometric patterns for every 3D spatial location, allowing the rendered views to be exceptionally realistic with fine-grained details. Our key contributions are:

- We propose a general radiance field to implicitly represent the 3D scene structure and appearance purely from a sparse set of 2D images. It has remarkable generalization to novel scenes.
- We explicitly integrate the principle of multi-view geometry to learn geometric details for each 3D query point along every query light ray. This allows the synthesized 2D views to be superior.
- We demonstrate significant improvement over baselines on three large-scale datasets and provide intuition behind our design choices through extensive ablation studies.

2 Related Work

Classic Multi-view Geometry. Classic approaches to reconstruct 3D geometry from images mainly include SfM and SLAM systems, which firstly extract and match hand-crafted geometric local features and then apply bundle adjustment for both shape and camera motion estimation (Hartley & Zisserman, 2004). Although they can recover visually satisfactory 3D models, the reconstructed shapes are usually sparse, discrete point clouds. In contrast, our GRF learns a neural function to implicitly represent the continuous 3D structures from images. Notably, however, the principle of classic multi-view geometry is explicitly integrated into our GRF to learn accurate and general features for every spatial location.

Geometric Deep Learning The recent advances of deep neural nets have yielded impressive progress in recovering explicit 3D shapes from either single or multiple images, including the voxel grid (Choy et al., 2016; Yang et al., 2019), octree (Riegler et al., 2017; Christian et al., 2017), point cloud (Fan et al., 2017; Qi et al., 2017) and triangle mesh (Wang et al., 2018; Groueix et al., 2018; Nash et al., 2020) based approaches. However, most of these methods only focus on individual 3D objects, whilst only few pipelines (Song et al., 2017; Tulsiani et al., 2018; Gkioxari et al., 2019) attempt to learn the structures for complex 3D scenes. Although these neural networks can predict realistic 3D structures of objects and scenes, they have two limitations. First, almost all of them require ground truth 3D labels to supervise the networks, resulting in the learned representations being unable to generalize to novel real-world scenes. Second, since the recovered 3D shapes are discrete, they are unable to preserve high-resolution geometric details. Being quite different, our GRF learns to infer continuous 3D shapes only from posed 2D images which can be cheaply acquired and also allow better generalization across real-world scenes.

Neural Implicit 3D Representations. The implicit representation of 3D shapes recently emerges as a promising direction to recover 3D geometries. It is initially formulated as level sets by optimizing deep neural nets which map *xyz* locations to an occupancy field (Mescheder et al., 2019) or a distance function (Park et al., 2019). The subsequent work (Niemeyer et al., 2020; Sitzmann et al., 2019; Liu et al., 2019b) introduce differentiable rendering functions, allowing 2D images to supervise the networks, but only producing decent results on individual objects. Using neural radiance fields instead, the latest NeRF (Mildenhall et al., 2020) and the succeeding NSVF (Liu et al., 2020), NeRF-wild (Martin-Brualla et al., 2020), GRAF (Schwarz et al., 2020) demonstrate impressive results to represent complex 3D scenes. However, each neural network (*i.e.,* a radiance field) can only represent a single scene, and is unable to generalize across novel scenarios. Additionally, it does not take into account the local geometric patterns for spatial locations, thereby the rendered 2D images being less realistic and sub-optimal. Uniquely, our GRF is a general radiance field which maps any set of images to the corresponding implicit 3D scene structure with geometric details.

Novel View Synthesis and Neural Rendering Novel view synthesis involves generating unseen views of a scene captured by multiple images. Existing methods usually learn an embedding of a scene and then estimate a new image given a viewing angle, including GAN based methods (Goodfellow et al., 2014; Radford et al., 2016), variational auto-encoders (Kingma & Welling, 2014), autoregressive models (Oord et al., 2016), and other generative frameworks (Eslami et al., 2018). Although photo-realistic single images can be generated, these methods tend to learn the manifold of 2D images, instead of exploiting the underlying 3D geometry for consistent multi-view synthesis.

Neural rendering techniques (Fried et al., 2020; Kato et al., 2020) have recently been actively investigated and integrated into 3D reconstruction pipelines, where there are no ground truth 3D data available but only 2D images for supervision. To render the discrete voxel grids (Yan et al., 2016; Rezende et al., 2016; Tulsiani et al., 2017), point clouds (Insafutdinov & Dosovitskiy, 2018), meshes (Kato et al., 2018; Chen et al., 2019; Liu et al., 2019a), and implicit surfaces (Liu et al., 2019b; Remelli et al., 2020), most of these techniques are designed with differentiable and approximate functions, but sacrifice the sharpness of synthesized images and the accuracy of estimated 3D shapes. By contrast, our GRF leverages the existing volume rendering method (Mildenhall et al., 2020) which is naturally differentiable and can accurately render the RGB per light ray.

3 GRF

3.1 Overview

Our GRF models the complex 3D scenes including the geometry and appearance as a neural function $f_{\mathbf{W}}$ where $\mathbf{W}$ represent learnable parameters. This function takes a set of K images together with their viewpoints $\{(\mathcal{I}_1, \mathcal{V}_1) \cdots (\mathcal{I}_k, \mathcal{V}_k) \cdots (\mathcal{I}_K, \mathcal{V}_K)\}$, which are sampled from a complex 3D scene, a query 3D point $p = \{x_p, y_p, z_p\}$, and its query viewpoint $\mathcal{V}_p = \{x_p^v, y_p^v, z_p^v\}$ as input, and then predicts the RGB value $\{r_p, g_p, b_p\}$ and the volumetric density d_p of that query 3D point p observed from the query viewpoint $\mathcal{V}_p$. Formally, it is defined as below:

$$(r_p, g_p, b_p, d_p) = f_{\mathbf{W}}\left(\Big\{(\mathcal{I}_1, \mathcal{V}_1) \cdots (\mathcal{I}_k, \mathcal{V}_k) \cdots (\mathcal{I}_K, \mathcal{V}_K)\Big\}, \{x_p, y_p, z_p\}, \{x_p^v, y_p^v, z_p^v\}\right) \quad (1)$$

Basically, this function is a general radiance field (GRF) which parameterizes arbitrary 3D scenes observed by the input K images. In the meantime, it returns both the appearance and geometry, when being queried at any location p from any viewpoint $\mathcal{V}$ in 3D space.

As illustrated in Figure 1, the proposed GRF firstly extracts general features for each light ray hitting through every pixel, and then reprojects those features back to the query 3D point p. After that, the corresponding RGB value and volumetric density of p are inferred from those features given a specific query viewing angle. This simple design of GRF follows the principle of classic multi-view geometry (Hartley & Zisserman, 2004), therefore guaranteeing the learned implicit representations meaningful and multi-view consistent. To realize our neural function $f_{\mathbf{W}}$, we introduce four components: 1) A feature extractor for every 2D pixel; 2) A reprojector to transform 2D features to 3D space; 3) An aggregator to obtain general features for a 3D point; and 4) A neural renderer to infer the appearance and geometry for that 3D point. All these components are connected and trained end-to-end with arbitrary scenes.

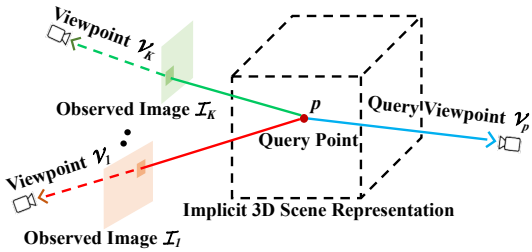


Figure 1: Overview of the proposed general radiance field.

3.2 Extracting General Features for 2D Pixels

The whole input set of observed 2D images together describes the geometry and appearance of a 3D scene, while each pixel of those images describes a specific point of that scene. This module aims to extract the general features of each pixel, so to learn the regional description and geometric patterns for each light ray. A naive approach is to directly use the raw *rgb* values as the pixel features. However, this is sub-optimal because the raw *rgb* values are sensitive to lighting conditions, environmental noise, etc. In order to learn more general and robust patterns for each pixel, we turn to use a more powerful encoder-decoder based convolutional neural network (CNN). As shown in Figure 2, our CNN module is designed with the follow two features:

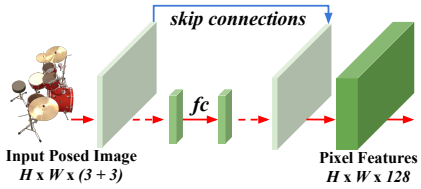


Figure 2: The CNN module to extract pixel features.

- Instead of directly feeding raw RGB images into the CNN module, we stack the corresponding viewpoint to each pixel of the image. This allows the learned pixel features to be explicitly aware of its relative position in the 3D scene space.
- We use skip connections between the encoder and decoder to preserve high frequency local features for each pixel, while optionally integrating a couple of fully connected (fc) layers in the middle of the CNN module to learn global features. The mixture of hierarchical features tends to to be more general and representative, effectively aiding the network in practice.

Details of the CNN module is presented in the appendix and all input images share the same CNN module. Note that, there are many ways to construct a pixel feature extractor, but identifying an optimal CNN module here is out the scope of this paper.

3.3 Reprojecting 2D Features to 3D Space

Considering the extracted pixel features are a compact description of the light ray emitting from the camera center up to the 3D scene surface, we naturally reproject the pixel features back to the 3D space along the light ray. Since there are no depth scans paired with RGB images, it is impossible to determine which particular 3D surface point the pixel features belong to. In this module, we preliminarily regard the pixel features as the representation of every location along the light ray in 3D space. With this simple formulation, every 3D point can theoretically have a copy of its corresponding 2D pixel features from each 2D image. Formally, given a 3D point p, an observed 2D view $\mathcal{I}_k$ together with its viewpoint $\mathcal{V}_k$ including extrinsics and intrinsics, the corresponding 2D pixel features $\mathbf{F}_p^k$ is retrieved by the reprojection operation below:

$$\mathbf{F}_p^k = \mathcal{P}\Big(\{\mathcal{I}_k, \mathcal{V}_k\}, \{x_p, y_p, z_p\}, \mathbf{I}_k\Big) \tag{2}$$

where the function $\mathcal{P}()$ follows the principle of multi-view geometry (Hartley & Zisserman, 2004) and $\mathbf{I}_k$ represents the image features extracted by the CNN module in Section 3.2. However, since the pixels of 2D images are discrete and bounded within a certain spatial size, while the 3D points are continuous in the scene space, after the 3D point p is projected to the plane of image $\mathcal{I}_k$, we apply two approximations to deal with the following issues.

- If the point lies inside of the image, we simply select the nearest pixel and duplicate its features to the 3D point. Note that, more advanced techniques may be applied to address the discretization issue, such as linear interpolation or designing a kernel function.

- If the point lies outside of the image, we assign a zero vector to the 3D point, which means there is no information observed. In fact, we empirically find that the nearest interpretation can also achieve good performance, but it is only applicable for relatively small-scale scenes.

Overall, the above simple reprojection operation explicitly retains the extracted 2D pixel features back to 3D space via the principle of geometry, and the module is naturally differentiable.

3.4 Obtaining General Features for 3D Points

Given a query 3D point p, it is able to retrieve a feature vector from each input image. However, given a set of input images, it is challenging to obtain a final feature vector for the point p, because:

- The total number of input images for each 3D scene is variable and there is no order for images. As a consequence, the retrieved feature vectors are also unordered and have arbitrary size.
- Since there are no depth scans paired with the input RGB images, it is unable to decide which feature vectors are the truly descriptions of the query point due to visual occlusions. Ideally, these features can be aware of the relative distance to the query point and then selected automatically.

To tackle these critical issues, we formulate this problem as an attention aggregation process. In particular, given the query 3D point p, its query viewpoint $\mathcal{V}_p$, and the set of retrieved pixel features $\{\mathbf{F}_p^1 \cdots \mathbf{F}_p^k \cdots \mathbf{F}_p^K\}$:

- For each retrieved feature vector $\mathbf{F}_p^k$, we firstly use shared MLPs to integrate the information of query point p, generating a new feature vector $\hat{\mathbf{F}}_p^k$ which is aware of the relative distance to the query point p. Formally, it is defined as:

$$\hat{\mathbf{F}}_p^k = MLPs(\mathbf{F}_p^k \oplus [x_p, y_p, z_p]), \quad (\oplus \text{ means concatenation in neural nets}) \tag{3}$$

- After obtaining the new set of position-aware features $\{\hat{\mathbf{F}}_p^1 \cdots \hat{\mathbf{F}}_p^k \cdots \hat{\mathbf{F}}_p^K\}$, we use the existing attention aggregation methods such as AttSets (Yang et al., 2020) and Slot Attention (Locatello et al., 2020) to compute a unique feature vector $\bar{\mathbf{F}}_p$ for the query 3D point p. Basically, the attention mechanism learns a unique weight for all input features and then aggregates them together. Note that, the selected attention mechanisms are permutation invariant with regard to the input set of feature vectors and can process an arbitrary number of elements. Formally, it is defined as:

$$\bar{\mathbf{F}}_p = \mathcal{A}\big(\hat{\mathbf{F}}_p^1 \cdots \hat{\mathbf{F}}_p^k \cdots \hat{\mathbf{F}}_p^K\big), \quad (\mathcal{A} \text{ is an existing attention function}) \tag{4}$$

Above all, for very query 3D point p, its final features $\bar{\mathbf{F}}_p$ explicitly preserve the general geometric patterns retrieved from the input 2D observations and are also made aware of the particular 3D point location within the scene space. This allows the 3D point features $\bar{\mathbf{F}}_p$ to be general and representative for its own geometry and appearance.

3.5 Rendering 3D Features

With the learned features $\bar{\mathbf{F}}_p$ for any query 3D point p, this module aims to infer its volumetric density d_p in 3D space and the corresponding RGB values $\{r_p, g_p, b_p\}$ given the query viewpoint $\mathcal{V}_p$. In particular, we follow the simple MLP based design of NeRF (Mildenhall et al., 2020). To maintain the multi-view consistency of all 3D points in the rendering process, the volumetric density is modeled as a function of the point features, while the RGB is modeled as a function of both point features and the query viewpoint. Formally, it is defined as:

$$d_p = MLPs(\bar{\mathbf{F}}_p) \tag{5}$$

$$(r_p, g_p, b_p) = MLPs\Big(\bar{\mathbf{F}}_p, \{x_p^v, y_p^v, z_p^v\}\Big) \tag{6}$$

To render the color of any query ray passing through the 3D scene, the classic volume rendering (Kajiya & Herzen, 1984) is applied and the integral is estimated by sampling a discrete set of 3D points along the query ray (Mildenhall et al., 2020). Eventually, novel 2D images can be directly synthesized from our GRF by querying a bunch of light rays. This allows the entire network to be trainable only with a set of posed 2D images, without requiring 3D data.

Table 1: Comparison of the PSNR (in dB) and SSIM of reconstructed images in the ShapeNetv2 dataset by our GRF, the deterministic GQN (Eslami et al., 2018), TCO (Tatarchenko et al., 2015), WRL (Worrall et al., 2017) and SRNs (Sitzmann et al., 2019). The higher the scores, the better the synthesized novel views. **Note that, our GRF solves a harder problem than SRNs, because GRF infers the novel scene representation in a single forward pass, while SRNs cannot.**

	50 Images (Group 1)		2 Images (Group 2)		1 Image (Group 3)	
	Chairs	Cars	Chairs	Cars	Chairs	Cars
TCO	24.31 / 0.92	20.38 / 0.83	21.33 / 0.88	18.41 / 0.80	21.27 / 0.88	18.15 / 0.79
WRL	24.57 / 0.93	19.16 / 0.82	22.28 / 0.90	17.20 / 0.78	22.11 / 0.90	16.89 / 0.77
dGQN	22.72 / 0.90	19.61 / 0.81	22.36 / 0.89	18.79 / 0.79	21.59 / 0.87	18.19 / 0.78
SRNs	26.23 / **0.95**	26.32 / **0.94**	**24.48 / 0.92**	**22.94 / 0.88**	**22.89 / 0.91**	**20.72 / 0.85**
GRF (Ours)	**27.10 / 0.95**	**27.21 / 0.94**	22.65 / 0.88	22.06 / 0.86	21.25 / 0.86	19.84 / 0.81

SRNs GRF (Ours) Ground Truth

Figure 3: Qualitative results of SRNs and our GRF for 50-view reconstruction of chairs and cars.

3.6 Implementation

The proposed four modules are connected and trained end-to-end. In the implementation, the CNN module may use different number of layers to extract pixel features according to the complexity of datasets. In the 3D point feature aggregation module, we use either AttSets (Yang et al., 2020) or Slot Attention (Locatello et al., 2020) to find a trade-off between computation and accuracy for different datasets. In the neural rendering module, all the designs of neural layers and other settings simply follow NeRF (Mildenhall et al., 2020). In our network, all the 3D locations and RGB values are processed by the positional encoding proposed by NeRF (Mildenhall et al., 2020). The L2 loss between the rendered RGB and the ground truth is used to optimize the whole neural network. All details are presented in the appendix.

4 Evaluation

4.1 Experiments on ShapeNetv2 Dataset

Following the experimental settings of SRNs (Sitzmann et al., 2019), we firstly evaluate our GRF on the chair and car classes of ShapeNetv2. Particularly, the chair has 4612 objects for training, 662 for validation and 1317 for testing, while the car has 2151 objects for training, 352 for validation and 704 for testing. Each training object has randomly sampled 50 images with a resolution of 128×128 pixels. We conduct the following three groups of experiments.

- Group 1: Novel-view synthesis of seen objects in the training split of the same category. In particular, we train two separate models on the training split of chairs and cars, and these two models are separately tested on the trained objects to generate completely novel views. During testing, the model is fed with 50 trained views and infers another 251 novel views for evaluation.
- Group 2: Novel-view synthesis of unseen objects in the testing split of the same category. The trained two models in Group 1 are tested on novel objects of the same category. During testing, the model is fed with 2 novel views of each novel object, inferring 251 novel views for evaluation.
- Group 3: Similar to Group 2, but only 1 novel view is fed into the model for novel view synthesis.

Basically, the experiments of Group 1, 2 and 3 are exactly the same as in SRNs (Sitzmann et al., 2019) which is also the state of the art on this ShapeNetv2 dataset. Table 1 compares the quantitative results of our GRF and four baselines. Note that, the recent NeRF and NSVF are not scalable to be trained on large number of scenes simultaneously because each scene is encoded into the network

parameters. Our method outperforms SRNs on 50-view reconstruction, demonstrating the strong capability of our GRF for 3D scene representation. Figure 3 shows the qualitative results.

Noticeably, our GRF still lags behind SRNs for novel scene reconstruction in both Group 2 and 3. Fundamentally, this is because our GRF solves a much harder problem than SRNs. In particular, our network directly infers the novel scene representation in a single forward pass, while SRNs needs to be retrained on all novel scenes to optimize the latent code. As a result, the experiments of Group 2 and 3 are significantly in favour of SRNs.

4.2 Experiments on Synthetic-NeRF Dataset

We further evaluate our approach for on the more complex dataset Synthetic-NeRF (Mildenhall et al., 2020). It consists of path-traced images of 8 synthetic objects with much more complicated geometry and realistic non-Lambertian materials than ShapeNet objects. For each scene, 100 views are rendered for training, another 200 novel views for testing. Each image has a high resolution of 800×800 pixels. For fair comparison, our experimental settings strictly follow the state of the art approach NSVF (Liu et al., 2020).

Table 2 presents the quantitative results of our GRF and five baselines. Our method outperforms the state of the art NSVF on all metrics. Figure 4 shows the qualitative results.

Table 2: The average scores of PSNR, SSIM and LPIPS (Zhang et al., 2018) for our GRF, SRNs (Sitzmann et al., 2019), NV (Lombardi et al., 2019), LLFF (Mildenhall et al., 2019), NeRF (Mildenhall et al., 2020) and NSVF (Liu et al., 2020) on the Synthetic-NeRF dataset.

	PSNR↑	SSIM↑	LPIPS↓
SRNs	22.26	0.846	0.170
NV	26.05	0.893	0.160
LLFF	24.88	0.911	0.114
NeRF	31.01	0.947	0.081
NSVF	31.75	0.954	**0.048**
GRF(Ours)	**32.02**	**0.960**	**0.048**

4.3 Experiments on Real-world Complex Scenes

In this section, we evaluate the novel view synthesis on complex real-world scenes captured by a handheld cellphone. Particularly, there are 8 scenes with roughly forward-facing images, 5 from LLFF (Mildenhall et al., 2019) and 3 from NeRF (Mildenhall et al., 2020). Each scene has 20 to 62 images, 1/8 of which for testing. All images have high resolution of 1008×756 pixels. Our experimental settings are the same as NeRF (Mildenhall et al., 2020).

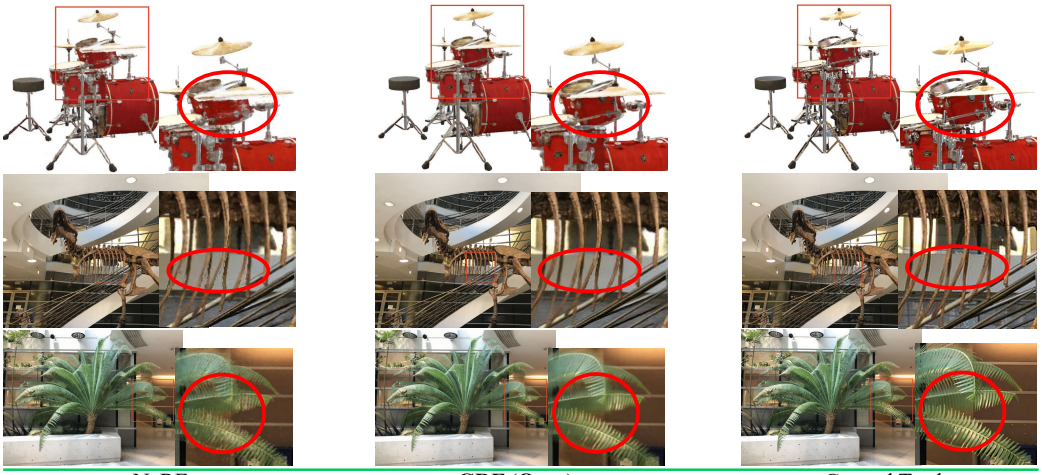
Table 3 compares the quantitative results of our GRF and three baselines. Note that, the recent NSVF (Liu et al., 2020) is unable to process these forward-facing scenes because the predefined voxels cannot represent the unbounded 3D space. Clearly, our method surpasses the state of the art by large margins, especially over the SSIM and LPIPS metrics. Compared with PSNR which only measures the accuracy of per pixel on average, the metrics SSIM and LPIPS favor high quality of photorealism, highlighting the superiority of our GRF to generate truly realistic images. Figure 4 compares the qualitative results of our method and NeRF. As highlighted by the red circles, our GRF can generate fine-grained and natural geometries in pixel level, while NeRF produces many artifacts. This clearly demonstrates our GRF indeed learns general and precise pixel features from the observed 2D images for 3D scene representation and rendering.

Table 3: Comparison of the average PSNR, SSIM and LPIPS scores of our GRF, SRNs (Sitzmann et al., 2019), LLFF (Mildenhall et al., 2019) and NeRF (Mildenhall et al., 2020) in the challenging real-world dataset.

	PSNR↑	SSIM↑	LPIPS↓
SRNs	22.84	0.668	0.378
LLFF	24.13	0.798	0.212
NeRF	26.50	0.811	0.250
GRF(Ours)	**26.52**	**0.834**	**0.182**

4.4 Ablation Study

To evaluate the effectiveness of the key components of our GRF, we conduct 3 groups of ablation experiments on the car category in ShapeNetv2 dataset. In particular, we train on the entire training split then randomly select 500 objects from the training split for testing. During testing, the model



NeRF **GRF (Ours)** Ground Truth

Figure 4: Qualitative results of NeRF and our GRF on the Synthetic-NeRF (the 1st row) and real-world datasets (the 2nd and 3rd rows).

infers 50 novel views for each of these 500 objects for evaluation. Basically, this setting is similar to the experiment of Group 1 in Section 4.1.

- Group 1: The viewpoints of the input images are removed from the CNN module. The CNN module only takes the pixel RGB as input, without being explicitly aware of the relative position of the pixel features.
- Group 2: The decoder of the CNN module is removed and each input image is encoded as a global feature vector. For any 3D query point which is rightly projected into the image boundary, that global feature vector is retrieved and reprojected to the query point. Basically, this ablation experiment is to validate how the pixel local features would benefit the learned 3D point features.
- Group 3: The advanced attention module is replaced by the simple max-pooling to aggregate the set of reprojected pixel features. This ablation aims to investigate how the visual occlusion can be better addressed with the soft attention mechanism.
- Group 4: The full model is trained and tested with the same settings for comparison.

Table 4: The average scores of PSNR and SSIM for ablated GRF in the subset of car category.

	PSNR↑	SSIM↑
(1) Remove Input Viewpoints	20.13	0.807
(2) Remove Pixel Local Features	20.23	0.818
(3) Replace Attention by Maxpooling	24.88	0.914
(4) The Full Model	**27.16**	**0.942**

Table 4 compares the scores of PSNR and SSIM for all ablated models. From this, we can see that: 1) The greatest impact is caused by the removal of image viewpoints from the CNN module and the miss of pixel local features to represent 3D points. It highlights that obtaining the position-aware and precise pixel features is crucially to represent 3D scenes from 2D images. 2) Using max-pooling to select the reprojected pixel features is detrimental to the performance, as it is sub-optimal to address the visual occlusions from different viewing angles. Note that, although the ablation experiments are conducted on a subset of cars, the PSNR and SSIM scores of the full model is comparable with the main results of Group 1 in Section 4.1. Therefore, the randomly selected objects are representative and the ablation reliable.

5 Conclusion

The proposed method models complex 3D scenes as a general radiance field. We demonstrated that it can learn general and robust 3D point features from a set of 2D observations. By using the principle of multi-view geometry to precisely map 2D pixel features back to 3D space and leveraging the attention mechanism to implicitly address the visual occlusions, our GRF can synthesize truly realistic 2D novel views. However, there are still limitations that lead to the future work. (1) More advanced CNN modules can be designed to learn better pixel features. (2) Depth scans can be integrated into the network to explicitly address visual occlusions.

REFERENCES

Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, Jose Neira, Ian D. Reid, and John J. Leonard. Past, Present, and Future of Simultaneous Localization and Mapping: Towards the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

Wenzheng Chen, Jun Gao, Huan Ling, Edward J. Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to Predict 3D Objects with an Interpolation-based Differentiable Renderer. *Advances in Neural Information Processing Systems*, pp. 9609–9619, 2019.

Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction. *European Conference on Computer Vision*, pp. 628–644, 2016.

H Christian, Shubham Tulsiani, and Jitendra Malik. Hierarchical Surface Prediction for 3D Object Reconstruction. *International Conference on 3D Vision*, pp. 412–420, 2017.

S.M. Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S. Morcos, Marta Garnelo, Avraham Ruderman, Andrei A. Rusu, Ivo Danihelka, Karol Gregor, David P. Reichert, Lars Buesing, Theophane Weber, Oriol Vinyals, Dan Rosenbaum, Neil Rabinowitz, Helen King, Chloe Hillier, Matt Botvinick, Daan Wierstra, Koray Kavukcuoglu, and Demis Hassabis. Neural Scene Representation and Rendering. *Science*, 360(6394):1204–1210, 2018.

Haoqiang Fan, Hao Su, and Leonidas Guibas. A Point Set Generation Network for 3D Object Reconstruction from a Single Image. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 605–613, 2017.

A Tewari O Fried, J Thies V Sitzmann, S Lombardi K Sunkavalli, and R Martin-brualla T Simon J Saragih. State of the Art on Neural Rendering. *Computer Graphics Forum*, 39(2):701–727, 2020.

Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh R-CNN. *IEEE International Conference on Computer Vision*, pp. 9785–9795, 2019.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A Papier-Mache Approach to Learning 3D Surface Generation. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 216–224, 2018.

Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

Eldar Insafutdinov and Alexey Dosovitskiy. Unsupervised Learning of Shape and Pose with Differentiable Point Clouds. *Advances in Neural Information Processing Systems*, pp. 2802–2812, 2018.

James T. Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH Computer Graphics*, 18(3), 1984.

Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3D Mesh Renderer. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3907–3916, 2018.

Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable Rendering: A Survey. *arXiv:2006.12057*, 2020.

Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *International Conference on Learning Representations*, 2014.

Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural Sparse Voxel Fields. *Advances in Neural Information Processing Systems*, 2020.

Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft Rasterizer: A Differentiable Renderer for Image-based 3D Reasoning. *IEEE International Conference on Computer Vision*, pp. 7708–7717, 2019a.

Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to Infer Implicit Surfaces without 3D Supervision. *Advances in Neural Information Processing Systems*, pp. 8293–8304, 2019b.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-Centric Learning with Slot Attention. *Advances in Neural Information Processing Systems*, 2020.

Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Transactions on Graphics*, 38(4), 2019. ISSN 15577368. doi: 10.1145/3306346.3323020.

Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. *arXiv:2008.02268*, 2020.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy Networks: Learning 3D Reconstruction in Function Space. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4455–4465, 2019.

Ben Mildenhall, Pratul Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescrip-tive sampling guidelines. *SIGGRAPH*, 2019.

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. *European Conference on Computer Vision*, 2020.

Charlie Nash, Yaroslav Ganin, S. M. Ali Eslami, and Peter W. Battaglia. PolyGen: An Autoregressive Generative Model of 3D Meshes. *International Conference on Machine Learning*, 2020.

Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable Volumetric Rendering: Learning Implicit 3D Representations without 3D Supervision. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3504–3515, 2020.

Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional Image Generation with PixelCNN Decoders. *Advances in Neural Information Processing Systems*, pp. 4790–4798, 2016. ISSN 10495258.

Onur Ozyesil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A Survey of Structure from Motion. *Acta Numerica*, 26:305–364, 2017.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 165–174, 2019.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660, 2017.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *International Conference on Learning Representations*, 2016.

Edoardo Remelli, Artem Lukoianov, Stephan R. Richter, Benoît Guillard, Timur Bagautdinov, Pierre Baque, and Pascal Fua. MeshSDF: Differentiable Iso-Surface Extraction. *arXiv:2006.03997*, 2020.

Danilo Jimenez Rezende, S. M. Ali Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised Learning of 3D Structure from Images. *Advances in Neural Information Processing Systems*, pp. 4996–5004, 2016.

Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. OctNet: Learning Deep 3D Representations at High Resolutions. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3577–3586, 2017.

Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. *arXiv:2007.02442*, 2020.

Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations. *Advances in Neural Information Processing Systems*, pp. 1119–1130, 2019.

Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas Funkhouser. Semantic Scene Completion from a Single Depth Image. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1746–1754, 2017.

Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Single-view to Multi-view: Reconstructing Unseen Views with a Convolutional Network. *arXiv:1511.06702*, 2015.

Shubham Tulsiani, Tinghui Zhou, Alexei A. Efros, and Jitendra Malik. Multi-view Supervision for Single-view Reconstruction via Differentiable Ray Consistency. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2626–2634, 2017.

Shubham Tulsiani, Saurabh Gupta, David Fouhey, Alexei A. Efros, and Jitendra Malik. Factoring Shape, Pose, and Layout from the 2D Image of a 3D Scene. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 302–310, 2018.

Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images. *European Conference on Computer Vision*, pp. 52–67, 2018.

Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow. Interpretable transformations with encoder-decoder networks. *IEEE International Conference on Computer Vision*, pp. 5726–5735, 2017.

Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective Transformer Nets: Learning Single-View 3D Object Reconstruction without 3D Supervision. *Advances in Neural Information Processing Systems*, pp. 1696–1704, 2016.

Bo Yang, Stefano Rosa, Andrew Markham, Niki Trigoni, and Hongkai Wen. Dense 3D Object Reconstruction from a Single Depth View. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(12):2820 – 2834, 2019.

Bo Yang, Sen Wang, Andrew Markham, and Niki Trigoni. Robust Attentional Aggregation of Deep Feature Sets for Multi-view 3D Reconstruction. *International Journal of Computer Vision*, 128: 53–73, 2020.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.