Generalizable Neural Radiance Fields for Novel View Synthesis with Transformer

Dan Wang, Xinrui Cui, Septimiu Salcudean, Fellow, IEEE and Z. Jane Wang, Fellow, IEEE

Abstract—We propose a Transformer-based NeRF (TransNeRF) to learn a generic neural radiance field conditioned on observed-view images for the novel view synthesis task. By contrast, existing MLP-based NeRFs are not able to directly receive observed views with an arbitrary number and require an auxiliary pooling-based operation to fuse source-view information, resulting in the missing of complicated relationships between source views and the target rendering view. Furthermore, current approaches process each 3D point individually and ignore the local consistency of a radiance field scene representation. These limitations potentially can reduce their performance in challenging real-world applications where large differences between source views and a novel rendering view may exist. To address these challenges, our TransNeRF utilizes the attention mechanism to naturally decode deep associations of an arbitrary number of source views into a coordinate-based scene representation. Local consistency of shape and appearance are considered in the ray-cast space and the surrounding-view space within a unified Transformer network. Experiments demonstrate that our TransNeRF, trained on a wide variety of scenes, can achieve better performance in comparison to state-of-the-art image-based neural rendering methods in both scene-agnostic and per-scene finetuning scenarios especially when there is a considerable gap between source views and a rendering view.

I. INTRODUCTION

Novel view synthesis is a long-standing open problem concerned with the rendering of unseen views of a 3D scene given a set of observed views [1], [2], [3], [4], [5]. It is a fundamental and challenging problem in 3D modeling [6], [7] and computer animation [8]. The essence of novel view synthesis is to explore and learn a view-consistent 3D scene representation from a sparse set of input views. The early work focused on modeling 3D shapes by discrete geometric 3D representations, such as mesh surface [9], [10] point cloud [11] and voxel grid [12], [13]. Although explicit 3D geometry-based representations are intuitive, these representations are discrete and sparse. Therefore, they are incapable of learning high-resolution renderings with sufficient quality for complex real-world scenes.

The recent promising work, implicit continuous 3D coordinate-based representation [14], [15], [16], [17], has been shown to have the potential to reconstruct 3D scenes with complicated high-resolution geometry and appearance. However, most discrete [9], [10], [11], [12], [13] and continuous [14], [15], [16], [17] 3D representations require ground-truth 3D geometry information as supervision, which is difficult to achieve in real-world scenes. Subsequently, research work [18], [19], [20], [21] revisits this requirement by introducing differential approximation and rendering functions and optimizing the 3D representation only by multi-view images. But these approaches do not exploit the power of continuous 3D coordinate-based representation sufficiently and therefore only produce over-smoothed renderings.

The more recent impressive Neural Radiance Field (NeRF)-based models [1], as a sub-class of implicit continuous representation, introduce neural radiance field scene representations, which map a continuous 3D location and 2D viewing direction to its volume density and radiance color. Unlike previous models in implicit continuous representation, NeRF-based models can achieve high quality of novel view synthesis from observed images in realistic complex scenes. However, these models need to optimize a scene-specific 3D representation for each scene, which is time-consuming and does not learn the shared information amongst scenes. Subsequently, to learn prior knowledge in diverse scenes, researchers [22], [23].
Fig. 2: Example results from IBRNet [2] (b) vs. the proposed TransNeRF (c) and ground-truth rendering view (a). We train a single scene-agnostic TransNeRF model on a large hybrid multi-scene dataset. It can effectively generalize to novel scenes without per-scene finetuning. In the same experimental setting for the scene-agnostic scenario, the result of our TransNeRF (c) is more realistic, containing fewer artifacts when compared with IBRNet (b).

[2], [24] generalize the radiance field scene representation by receiving a pooling-based multi-view feature as a conditional input.

While these image-conditioned NeRFs [22], [23], [2], [24] can generalize across different scenes, previous studies of NeRFs have not investigated in depth the relationship between observed source views and the target rendering view which is essential for novel view synthesis. The reasons for this is that previous NeRF approaches were built on multi-layer perceptrons networks (MLPs) which are incapable of receiving and processing an arbitrary number of observed views; consequently, they use an auxiliary pooling-based model to aggregate multi-view features. This ignores in-depth and high-level complicated relationships across observed views and the rendering view.

This limitation impairs the potential of NeRFs to explore and learn a view-consistent 3D scene representation from observed views, especially in practical applications where the observed source views might be captured at camera poses that are very different from the camera pose of the target view. As illustrated in Fig [1], when camera poses of source views are similar to the rendering view, source views and the target view are distributed in a local region in 3D scene representation space, making it possible to approximate their relationship by a linear function as in previous techniques [22], [23], [2], [24]. However, as the difference between observed views and the rendered view increases, the correlation becomes more complicated, making it challenging for these approaches to synthesize a novel view. Therefore, in this scenario, prior MLP-based NeRFs [22], [23], [2], [24], using the pooling-based function to fuse the multi-view, fail to resolve this challenge.

To tackle this unmet need, in this paper, we propose a Transformer-based NeRF framework (TransNeRF) utilizing its powerful attention mechanism to learn a general continuous 3D scene representation from an arbitrary number of observed views. The learning process of TransNeRF is divided into two stages: the first deals with the surrounding-view space and the second deals with in the ray-cast space, shown in Fig [3]. In the surrounding-view space, unlike previous MLP-based work requiring an auxiliary pooling-based operation to fuse source views, our framework is only built on the Transformer-based network and leverages its attention mechanism to integrate deep relationships between the rendering view and observed views as a coordinate-based scene representation. Specifically, in our attention layers at our Density-ViewDecoder and Color-ViewDecoder, TransNeRF can naturally decode observed views and spatial location information for a query 3D point in the rendering ray direction into its density and directional color representations. In the ray-cast space, when rendering a query 3D-point, our TransNeRF considers adjacent 3D points on the target ray simultaneously by attention layers in Density-RayDecoder and Color-RayDecoder. In contrast, in prior work each 3D point is processed independently. In this paper, taking advantage of the attention mechanism, our proposed approach enhances the local consistency of a 3D scene representation in both the ray-cast space and the surrounding-view space by a unified Transformer-based network. Therefore, the proposed TransNeRF has the capability to learn a more comprehensive neural radiance field that can effectively generalize to novel scenes without per-scene finetuning, as shown in Fig. 2.

Our contributions can be summarized as follows:

- We introduce TransNeRF, a unified Transformer-based architecture, to model a general neural radiance field from observed views for novel view synthesis, whereas previous MLP-based NeRFs need an auxiliary pooling model to aggregate multi-view information when dealing with an arbitrary number of observed source views.
- TransNeRF, utilizing the attention mechanism, integrates information of the projected 2D pixels in the surrounding source views and the neighboring 3D points along the query ray when rendering the density and radiance color of a query 3D point. It gives our model a better understanding of the shape and appearance consistency in a local neural radiance field.
- TransNeRF can explore complicated relationships between observed views and the rendering view and fuses the learned high-level multi-view information into
coordinate-based 3D scene representation by our attention layers.

- Experiments demonstrate that in both scene-agnostic and per-scene finetuning experimental settings, TransNeRF achieves more realistic rendering results than previous state-of-the-art methods, especially when source views are captured at camera poses with a considerable difference to the camera pose of the rendering view.

In this paper, we present a generic Transformer-based NeRF framework with a high capacity in modeling radiance field scene representation from observed source view images. Similar to MLP-based NeRFs, our framework can be easily extended to advanced TransNeRF derivatives, e.g. from NeRF to NeRF-W [25] by adding appearance and transient variables as inputs to model uncertainty in the wild scenarios. Here, we only focus on the vanilla TransNeRF and hope this work will serve as a blueprint for future work.

II. RELATED WORK

A. Novel View Synthesis

The goal of novel view synthesis is to reconstruct unseen views of a scene from its observed multiple images. For this task, solutions can be separated into three categories: novel view rendering by image-based representation, discrete 3D shape-based representation, and implicit continuous 3D spatial coordinate-based representation.

**Image-based representation.** The image-representation-based approaches leverage GAN models and auto-encoders [26] to learn a global embedding from multi-view images and render an unseen view given its camera parameters. However, these image-representation-based models are limited by the lack of understanding of the underlying 3D shape representation. In contrast, the two 3D-representation-based types reconstruct novel views by exploiting multi-view consistent 3D shape representation.

**Discrete 3D shape-based representation.** For novel view synthesis, the discrete 3D-shape representation involves point cloud [11], mesh surface [9], voxel grid [12], [13], [19] approaches. The point-cloud-based representation [11] reconstructed from multi-view images is typically sparse, making it difficult to infer the missing information. The mesh-based representation [9], [10] requires a template mesh with fixed topology and rigid texture parameterizations for the category-specific shape. In complex real-world scenes, it is challenging to fulfill this requirement. The volume-based representation [12], [13] can easily leverage CNNs to obtain predicted voxel grids. Nevertheless, it causes discretization artifacts from low-resolution voxel grids and limits its resolution-scale ability with computational and memory constraints.

**Implicit continuous 3D spatial coordinate-based representation.** Recently, another promising branch implicitly represents geometric and appearance information of a 3D point in a scene as a function of its continuous spatial coordinates. Therefore this implicit continuous 3D shape representation has the potential to render higher-resolution complicated geometry and appearance of scenes without storing an entire scene representation. In this direction, research work modeling occupancy fields [14], [15] or signed distance functions [16], [17] requires ground-truth 3D information for supervision, limiting its real-world applications. To relax this requirement, researchers [20], [21] leverage differential rendering functions to allow 2D images as only supervision information for 3D representation. However, these models are unable to fully exploit the advantage of the implicit continuous representation, which limits it to simple geometric and appearance scenarios.

**Neural radiance field scene representation.** More recently, the impressive neural radiance fields (NeRF) [11] have shown a solid ability to synthesize novel views by representing continuous scenes as 5D radiance fields in MLPs. Nevertheless, NeRF optimizes each scene representation independently, not exploring the shared information amongst scenes and being time-consuming. To solve this, researchers proposed models, such as PixelNeRF [22], MVSNeRF [23], IBRNet [2], which receives as conditional inputs multiple observer views to learn a general neural radiance field. Followed the design principle of divide-and-conquer, they have two separate components: a CNN feature extractor for a single image and a MLP as a NeRF network. For a single-view stereo, in these models, CNNs map an image to a feature grid, and MLPs map a query 5D coordinate and its corresponding CNN feature to a single volume density and view-dependent RGB color. For a multi-view stereo, since CNN and MLP are unable to process an arbitrary number of input views, they first process the coordinates and corresponding features in each view coordinate frame independently and obtain an image-conditioned intermediate representation for each view. Next, they use an auxiliary pooling-based model to aggregate these intermediate representations across the views within the NeRF network. In the 3D understanding, multiple views provide additional information about the scene.

Nonetheless, pooling-based fusion models in these methods can barely explore the complex relationship across multiple views for 3D scene understanding. Furthermore, processing each 3D point independently ignores the local consistency of a 5D radiance field in a scene. To tackle it, we propose an encoder-decoder Transformer framework, TransNeRF, to represent the neural radiance field scene. Compared with the pooling-based multi-view representation in previous work, TransNeRF can explore deep relationships amongst multiple views and aggregate multi-view information into the coordinate-based scene representation by the attention mechanism in a single Transformer-based NeRF. Furthermore, TransNeRF can learn the local geometry consistency of shape and appearance in a scene by considering the corresponding information in the ray-cast space and the surrounding-view space.

B. Transformer

Transformer recently emerged as a promising network framework and achieved impressive performance in natural language processing [27] and computer vision [28], [29], [30]. The main idea behind this approach is to utilize the multi-head self-attention operation to explore the dependence within input
tokens and then learn a global feature representation. In the object detection task, DETR [28] presents a new framework combining a 2D CNN with a Transformer and predicts object detection in parallel as a sequence of output tokens. In image classification, ViT [29] demonstrates the impressive ability to detect in parallel as a sequence of output tokens. In image detection in parallel as a sequence of output tokens. In image classification, ViT [29] demonstrates the impressive ability to detect in parallel as a sequence of output tokens. In image classification, ViT [29] demonstrates the impressive ability to detect in parallel as a sequence of output tokens. In image classification, ViT [29] demonstrates the impressive ability to...
camera pose $\Theta$ is
\[ \mathcal{L} = \sum_{r \in \mathcal{R}(\Theta)} \| \tilde{C}(r) - C(r) \|^2, \]
where $\mathcal{R}(\Theta)$ is the set of all camera rays of the desired virtual camera with pose $\Theta$.

Given a set of observed source views, the rendering loss $\mathcal{L}$ between the observations and the predictions is minimized by optimizing the parameters of the neural radiance field $F_{\text{TransNeRF}}$. Here, our TransNeRF is fully differentiable and can be trained end-to-end only requiring source views. After being trained on a large dataset containing diverse scenes, our TransNeRF effectively generalizes to unseen scenes even without per-scene finetuning. Accordingly, given the camera pose of a query novel view of a scene and its multiple source views, our TransNeRF can render the query novel view from the target camera center towards the 3D space by casting rays from the target camera center towards the 3D space using volume rendering in Eq. (2).

\section*{A. Density Decoder in Surrounding-view Space}

We first present our density decoder in surrounding-view space (Density-ViewDecoder) fusion source views into the latent volume-density representation for a query 3D point.

For each source view image, we first extract its feature volume by a pre-trained view-shared U-Net. A query 3D point $(x, y, z)$ is then projected into each source-view image $\mathbf{f}_m^v$ by its camera projection matrix $\Theta^m$ to extract the corresponding RGB color $\mathbf{c}_m^v$ and feature vector $\mathbf{f}_m^v$ at the projected 2D pixel location through bilinear interpolation. In each source view, we also record its viewing direction $\mathbf{d}_m^v$ for the projected 2D pixel from the source camera pose. And then based on these information, we obtain the initial source-view embeddings $\{\mathbf{x}_m^v\}_{m=1}^M$ for source views, as in [2].

For the query 3D point, Density-ViewDecoder receives the initial source-view embeddings $\{\mathbf{x}_m^v\}_{m=1}^M$ and the learnable query density embedding $\mathbf{x}_0^v$ as inputs $\mathbf{X}_0$. The Density-ViewDecoder can be formulated as follows:
\begin{align}
\mathbf{X}_0 &= [\mathbf{x}_0^v; \mathbf{x}_1^v; \mathbf{x}_2^v; \cdots; \mathbf{x}_M^v] \tag{4} \\
\tilde{\mathbf{X}}_l &= \text{Norm}(\text{FFN}(\tilde{\mathbf{X}}_{l-1}) + \mathbf{X}_{l-1}) \tag{5} \\
\mathbf{X}_l &= \text{Norm}(\text{FFN}(\tilde{\mathbf{X}}_l) + \tilde{\mathbf{X}}_l), \tag{6}
\end{align}
where $l$ denotes the index of a basic block ($l = 1, \cdots, L$), "Norm" is the layer normalization function and "FFN" is a position-wise feed-forward network. At the L-th block, we can obtain $\mathbf{X}_L = [\mathbf{x}_0^v; \mathbf{x}_1^v; \mathbf{x}_2^v; \cdots; \mathbf{x}_M^v]$. In the Density-ViewDecoder, we concatenate the embedding $\mathbf{x}_0^v$ and its 3D coordinate location $(x, y, z)$ as the output density representation for the query 3D point.

Density-view attention layers (Density-ViAttn) can explore deep relationships among source views, defined as follows:
\[ \text{Density-ViAttn}(\mathbf{X}) = \text{MH-Attn}(\mathbf{X}, \mathbf{X}, \mathbf{X}), \tag{7} \]
where the multi-head attention function is defined as:
\[ \text{MH-Attn}(Q, K, V) = \text{Cat}(A_1, \cdots, A_H)W, \tag{8} \]
where $A_h = \text{Attention}(Q_h, K_h, V_h)$,
\[ Q_h = \mathbf{W}_Q^hK_h = \mathbf{W}_K^hV_h = \mathbf{W}_V^h \]
Here, $N_q$ queries are stacked in $Q \in \mathbb{R}^{N_q \times d_k}$, a set of $N_{kv}$ key-value pairs are stacked in $K \in \mathbb{R}^{N_{kv} \times d_k}$ and $V \in \mathbb{R}^{N_{kv} \times d_k}$. And $\mathbf{W}_Q^h, \mathbf{W}_K^h, \mathbf{W}_V^h \in \mathbb{R}^{d_k \times d_k}$ and $\mathbf{W} \in \mathbb{R}^{d_k \times d_k}$ are parameter matrices ($H \times d_k = d_k$ and $d_h$ is the feature dimension in each head). And the Attention function is computed by
\[ \text{Attention}(Q_h, K_h, V_h) = \text{softmax}(\frac{Q_hK_h^T}{\sqrt{d_k}})V_h, \tag{9} \]
where attention-score, stored in softmax$(\frac{Q_hK_h^T}{\sqrt{d_k}})$, of a specific value is obtained by the match between this query and the key paired with the target value. Our Density-ViewDecoder is invariant to permutations of source views and can receive an arbitrary number of source views.

\section*{B. Density Decoder in Ray-cast Space}

The density decoder in ray-cast space (Density-RayDecoder) decodes density information of the query 3D point by aggregating the density features of the neighboring 3D points along the target-view ray.

For the query point $q^t$ and neighboring $2n$ points $\{q_0^{-n}, \cdots, q_0^{1-n}, q_0^{-1}, \cdots, q_0^{+1}, \cdots, q_0^{+n}\}$ along the target-viewing ray, we denote $[\sigma_0^{-n}; \cdots; \sigma_0^{1-n}; \sigma_0^{0}; \sigma_0^{1-n}; \cdots; \sigma_0^{+n}]$ as their initial density representations at the input end of the Density-RayDecoder. Here, the initial density representation for each 3D point is computed by a linear function of the Density-RayDecoder output for the corresponding point $(\sigma_0 = FC(\mathbf{x}_0^v \odot (x, y, z))$, where $\odot$ is the concatenation operation). And then positional encodings $\mathbf{E}^{\text{pos}}$ are added to density representations of neighboring 3D points to keep their position information in the ray-cast space. Each positional encoding informs each point of its 3D spatial location, which is computed by utilizing sine and cosine functions of different frequencies in a similar way to [28]. The Density-RayDecoder is represented as:
\begin{align}
\mathbf{D}_0 &= [\sigma_0^{-n}; \cdots; \sigma_0^{1-n}; \sigma_0^{0}; \sigma_0^{1}; \cdots; \sigma_0^{+n}] + \mathbf{E}^{\text{pos}} \tag{10} \\
\tilde{\mathbf{D}}_l &= \text{Norm}(\text{Density-Attn}(\mathbf{D}_{l-1}) + \mathbf{D}_{l-1}) \tag{11} \\
\mathbf{D}_l &= \text{Norm}(\text{FFN}(\tilde{\mathbf{D}}_l) + \tilde{\mathbf{D}}_l), \tag{12}
\end{align}
where the density attention layer (Density-Attn) is computed as Density-Attn($\mathbf{D}$) = MH-Attn($\mathbf{D}, \mathbf{D}, \mathbf{D}$) fusing information of surrounding 3D points on the target-viewing ray. Finally, at the end block, the Density-RayDecoder outputs the density representation $\sigma_0^v$ of the query 3D point and then we use a linear function to project it to the density value for the query point.

\section*{C. Color Decoder in Surrounding-view Space}

The color decoder in surrounding-view space (Color-ViewDecoder) learns a query directional emitted color as
a function of viewing direction, source-view representations learned from Density-ViewDecoder and the latent density representation from Density-RayDecoder.

The Color-ViewDecoder can be formulated as follows:

\[
\hat{Y}_i = \text{Norm}(\text{Color-ViAttn}(Y_{i-1}, X_{src}, C_{src}) + Y_{i-1}) \quad (13)
\]

\[
Y_i = \text{Norm}(\text{FFN}(\hat{Y}_i) + \hat{Y}_i) . \quad (14)
\]

In the color-view layer (Color-ViAttn), the initial query directional color embedding is represented as \( Y_0 = FC(\sigma^l) \odot d_{tgt} \), where \( \sigma^l \) is the latent density representation from Density-RayDecoder and \( d_{tgt} \) is the target-viewing direction for the query 3D point. The Color-ViAttn layer is calculated as:

\[
\text{Color-ViAttn}(Y, X_{src}, C_{src}) = \text{MH-Attn}(Y, X_{src}, C_{src}) , \quad (15)
\]

where the value is \( C_{src} = [\gamma(e^1_{src}); \gamma(e^2_{src}); \cdots; \gamma(e^M_{src})] \) (\( \gamma(\cdot) \) is the embedding function) and the key is \( X_{src} = [FC(x^l_1) \odot d^1_{src}; FC(x^l_2) \odot d^2_{src}; \cdots; FC(x^l_M) \odot d^M_{src}] \) representing conditional source-view information. The output of Color-ViewDecoder \( y^l_i \) is the latent color representation for the query 3D point.

D. Color Decoder in Ray-cast Space

The color decoder in ray-cast space (Color-RayDecoder) learns a query directional emitted color by fusing color features of adjacent 3D points along the target ray in color attention layers (Color-Attn(\( Z \) = MH-Attn(\( Z, Z, Z \))). The Color-RayDecoder can be formulated as follows:

\[
\begin{align*}
Z_0 &= [z_0^{-n}; \cdots; z_0^{-1}; z_0^0; z_0^1; \cdots; z_0^{+n}] + E_{\text{pos}} \quad (16) \\
\hat{Z}_i &= \text{Norm}(\text{Color-Attn}(Z_{i-1}) + Z_{i-1}) \quad (17) \\
Z_i &= \text{Norm}(\text{FFN}(\hat{Z}_i) + \hat{Z}_i) , \quad (18)
\end{align*}
\]

where the latent color representation \( y^l_i \) for the query 3D point is assigned to the corresponding \( z^l_0 \) and likewise for latent color representations for adjacent \( 2n \) 3D-points in local ray-cast space.

Subsequently, after the Color-RayDecoder, we use a linear function to project the output color embedding \( z^l_i \) to its output predicted color value. Then the predicted volume density and directional color of a query point along a camera ray of the desired virtual camera are put forward to the classical volume rendering in Eq.\((19)\).

The implementation details of the network and training are described in the supplemental material.

IV. EXPERIMENTS

We evaluate our approach in the following experimental settings:

- Scene-agnostic setting: we train a single scene-agnostic model on a large training dataset, including various camera setups and scene types. We test its generalization ability to unseen-scene view synthesis on all test scenes from the evaluation dataset;
- Per-scene finetuning setting: our pre-trained scene-agnostic model can also be finetuned on each test scene. We then evaluate each scene-specific model on its corresponding scene separately.

The experiments are designed to examine whether our TransNeRF can efficiently learn a neural radiance field scene representation in scenarios where the difference between source views and the target rendering view varies.

A. Datasets

We train and evaluate our method on a collection of several multi-view datasets containing both synthetic data and real data, as in IBRNet [2].

- **Real-world datasets for training** include the Spaces dataset [34], RealEstate10K [35], and the handheld-cellphone-captured scene dataset. The Spaces dataset [34] has 100 scenes and each scene is collected with a 16 camera rig at 3 to 10 rig positions. RealEstate10K [35], a large indoor-scene dataset, is captured from around 80K video clips with camera poses. The cellphone-captured scene dataset contains 95 real scenes (36 from LLFF [33] and 59 from IBRNet [2]), where each scene consists of 20 to 60 forward-facing images.
- **Synthetic dataset for training** is generated by IBRNet [2] from Google Scanned Objects, which contains 1,030 models with a variety of view density rates.

Similar to the sampling strategy in [2], we randomly sample \( M \) source views for each target view from a pool of \( m \times M \) views where \( M \) is sampled uniformly at random from \( \{8, 12\} \) and \( m \) is from \( \{1, 5\} \).

- **Real-world dataset for evaluation** collects 8 complex real-world scenes captured with a handheld cellphone (5 from LLFF [33] and 3 from NeRF [1]). Each scene consists of 20 to 62 forward-facing images with 1008 \( \times \) 756 pixels, and 1/8th of these is held out as the test set (7/8 for per-scene finetuning).
- **Synthetic dataset for evaluation**, adopted from NeRF [1], includes 8 objects with complicated geometry and realistic non-Lambertian materials which are rendered at 800 \( \times \) 800 pixels from viewpoints sampled either on the upper hemisphere or full sphere (100 views for per-scene finetuning and 200 views for testing).

**Evaluation dependence on the difference between source views and the target view.** For the evaluation of a specific scene, target rendering views are from the testing set of evaluation datasets, and the corresponding surrounding-view set is contained in the training set of evaluation datasets. In this evaluation, we sampled \( M=10 \) views from the surrounding-view set as source views to render a target view. And the sampling procedure of source views is defined as: for each target rendering view, we first compute the differences between the target-view camera pose and the surrounding-view camera poses using Euclidean transformation matrices of camera poses [11, 2], and then rank the surrounding views according to how different they are, from small to large; by this difference ranking, we construct \( N_S \) source-view sets \( \{S_i\}_{i=1}^{N_S} \) of 10 views from the surrounding-view set to render each test view. For the real-world evaluation dataset, there are
### Table I: Quantitative Comparisons of Methods (PixelNeRF [22], MVSNeRF [23], IBRNet [2] and the proposed TransNeRF) for the Scene-agnostic Setting on the Realistic Synthetic Dataset [1]

| Scene | S1 | S2 | S3 | S4 | S5 | S6 |
|-------|----|----|----|----|----|----|
| Chair | 21.21 | 23.50 | 28.55 | 29.06 | 0.890 | 0.910 | 0.942 | 0.954 | 0.135 | 0.108 | 0.066 | 0.055 |
| Lego | 19.51 | 21.13 | 24.64 | 26.13 | 0.834 | 0.862 | 0.917 | 0.923 | 0.174 | 0.148 | 0.106 | 0.090 |
| Ship | 21.31 | 21.79 | 22.92 | 24.57 | 0.803 | 0.808 | 0.825 | 0.836 | 0.267 | 0.256 | 0.227 | 0.211 |
| Drums | 17.00 | 18.74 | 21.28 | 21.99 | 0.771 | 0.807 | 0.893 | 0.902 | 0.184 | 0.164 | 0.113 | 0.105 |
| Mic | 20.53 | 21.61 | 24.15 | 25.65 | 0.852 | 0.864 | 0.891 | 0.914 | 0.165 | 0.165 | 0.135 | 0.135 |
| Ficus | 21.67 | 22.41 | 25.80 | 26.99 | 0.799 | 0.797 | 0.835 | 0.881 | 0.226 | 0.207 | 0.169 | 0.164 |
| Materials | 19.47 | 20.43 | 23.74 | 25.14 | 0.827 | 0.847 | 0.895 | 0.903 | 0.195 | 0.174 | 0.124 | 0.110 |
| Hotsdog | 22.14 | 24.70 | 30.45 | 32.70 | 0.902 | 0.919 | 0.958 | 0.968 | 0.135 | 0.115 | 0.066 | 0.054 |
| Ave | 21.20 | 22.47 | 25.31 | 26.45 | 0.857 | 0.874 | 0.913 | 0.922 | 0.161 | 0.143 | 0.104 | 0.092 |

### Table II: Quantitative Comparisons of Methods (PixelNeRF [22], MVSNeRF [23], IBRNet [2] and the proposed TransNeRF) for the Scene-agnostic Setting on the Realistic Front-facing Dataset [33]

| Scene | S1 | S2 | S3 | S4 | S5 | S6 |
|-------|----|----|----|----|----|----|
| Fern | 20.65 | 21.12 | 23.69 | 23.70 | 0.671 | 0.699 | 0.767 | 0.771 | 0.355 | 0.322 | 0.250 | 0.247 |
| Trex | 18.63 | 19.24 | 21.83 | 23.84 | 0.705 | 0.722 | 0.849 | 0.850 | 0.392 | 0.377 | 0.239 | 0.237 |
| Homs | 16.03 | 16.88 | 21.68 | 21.85 | 0.633 | 0.660 | 0.788 | 0.797 | 0.467 | 0.448 | 0.299 | 0.287 |
| Fortress | 21.57 | 24.21 | 29.97 | 29.98 | 0.719 | 0.732 | 0.879 | 0.880 | 0.321 | 0.287 | 0.155 | 0.152 |
| Leaves | 15.56 | 16.37 | 20.30 | 20.31 | 0.513 | 0.561 | 0.722 | 0.724 | 0.418 | 0.378 | 0.228 | 0.226 |
| Orchids | 13.86 | 14.29 | 18.72 | 18.54 | 0.895 | 0.984 | 0.915 | 0.923 | 0.529 | 0.484 | 0.328 | 0.329 |
| Room | 21.52 | 22.74 | 29.70 | 29.71 | 0.820 | 0.858 | 0.941 | 0.944 | 0.318 | 0.293 | 0.153 | 0.153 |
| Flower | 17.84 | 19.46 | 26.61 | 26.83 | 0.614 | 0.663 | 0.884 | 0.886 | 0.414 | 0.376 | 0.103 | 0.103 |
| Ave | 16.30 | 17.68 | 26.69 | 22.94 | 0.576 | 0.614 | 0.749 | 0.760 | 0.459 | 0.422 | 0.273 | 0.260 |

**PSNR** stands for Peak Signal-to-Noise Ratio, **SSIM** for Structural Similarity Index, and **LPIPS** for Learned Perceptual Image Patch Similarity.
TABLE III: Quantitative comparisons of methods (PixelNeRF [22], MVSNeRF [23], IBRNet [2] and the proposed TransNeRF) for the per-scene fine-tuning setting on the realistic synthetic dataset [1].

| Scene | PixelNeRF | MVSNeRF | IBRNet | TransNeRF | PixelNeRF | MVSNeRF | IBRNet | TransNeRF | PixelNeRF | MVSNeRF | IBRNet | TransNeRF |
|-------|-----------|---------|--------|----------|-----------|---------|--------|----------|-----------|---------|--------|----------|----------|
| S1    | 32.08     | 29.43   | 28.04  | 32.64    | 0.721     | 0.589   | 0.632  | 0.776    | 0.300     | 0.262   | 0.303  | 0.381    |
| S2    | 20.52     | 21.46   | 21.05  | 22.36    | 0.774     | 0.709   | 0.758  | 0.843    | 0.394     | 0.323   | 0.347  | 0.408    |
| S3    | 19.77     | 20.53   | 20.34  | 21.85    | 0.791     | 0.826   | 0.867  | 0.910    | 0.264     | 0.220   | 0.277  | 0.312    |
| S4    | 18.49     | 24.30   | 28.79  | 31.04    | 0.761     | 0.822   | 0.894  | 0.901    | 0.272     | 0.216   | 0.256  | 0.339    |

TABLE IV: Quantitative comparisons of methods (PixelNeRF [22], MVSNeRF [23], IBRNet [2] and the proposed TransNeRF) for the per-scene fine-tuning setting on the real forward-facing dataset [1].

| Scene | PixelNeRF | MVSNeRF | IBRNet | TransNeRF | PixelNeRF | MVSNeRF | IBRNet | TransNeRF | PixelNeRF | MVSNeRF | IBRNet | TransNeRF |
|-------|-----------|---------|--------|----------|-----------|---------|--------|----------|-----------|---------|--------|----------|----------|
| S1    | 32.55     | 29.43   | 28.04  | 32.64    | 0.721     | 0.589   | 0.632  | 0.776    | 0.300     | 0.262   | 0.303  | 0.381    |
| S2    | 20.52     | 21.46   | 21.05  | 22.36    | 0.774     | 0.709   | 0.758  | 0.843    | 0.394     | 0.323   | 0.347  | 0.408    |
| S3    | 19.77     | 20.53   | 20.34  | 21.85    | 0.791     | 0.826   | 0.867  | 0.910    | 0.264     | 0.220   | 0.277  | 0.312    |
| S4    | 18.49     | 24.30   | 28.79  | 31.04    | 0.761     | 0.822   | 0.894  | 0.901    | 0.272     | 0.216   | 0.256  | 0.339    |
$N_s = 3$ sets, the top-10 views ($S_1$), the middle-10 ($S_2$), and the bottom-10 ($S_3$) respectively; for the synthetic evaluation dataset, $N_s = 4$, the top-10 ($S_1$), the middle-10 ($S_2$), the 3/4th-10 ($S_3$), and the bottom-10 ($S_4$). This sampling strategy can be used to compare methods in scenarios where the camera poses of source views have diverse degrees of difference, from small to large, relative to the camera pose of the target view.

\section*{B. Metrics}

For the task of novel view synthesis, we quantitatively evaluate the rendered image quality based on PSNR, SSIM [37] (higher is better), and LPIPS [38] (lower is better) by comparing with the ground truth.

\section*{C. Baselines}

We compare the proposed method quantitatively and qualitatively with top-performing generic NeRF approaches (PixelNeRF [22], MVSNeRF [23], IBRNet [2] and our TransNeRF) that are also
designed to generalize the scene-specific NeRF representation to a scene-agnostic one by taking multi-view features as conditional inputs. In the evaluation, each comparing method trains a single set of model parameters on the same training dataset and tests their generalizability for novel scenes on the same evaluation dataset described in Sec. IV-A as do our model. Furthermore, for a specific scene, we also evaluate comparing methods in the per-scene finetuning experimental setting. All comparing approaches are implemented based on their released codes in the same training setting.

D. Results

In both the scene-agnostic and per-scene finetuning settings, we evaluate the performance of different methods in scenarios where the source views belong to the different source view sets defined in \( \{ S_i \}_{i=1}^{N_s} \) in Sec. IV-A. To render a testing view, each compared approach receives as input the same source-view set. We present both qualitative and quantitative results for evaluation. PSNR/SSIM (higher is better) and LPIPS (lower is better) are used as quantitative metrics. And for quantitative results, the best score for each category is in
bold. Tab. I and II show quantitative results for the synthetic and real data in the scene-agnostic setting. The quantitative results of the per-scene finetuning setting for the synthetic and real datasets are displayed in Tab. III and IV respectively. Qualitative results are shown in Fig. 3 and 5 for the scene-agnostic and per-scene finetuning settings, where the i-th column exhibits the rendering results for different methods based on the source view set $S_i$.

1) Scene-agnostic Experiments: In scene-agnostic experiments, we compare our TransNeRF with PixelNeRF [22], MVSNeRF [23] and IBRNet [2] on the real forward-facing dataset [33] and the realistic synthetic dataset [2].

Tab. I and II show that the proposed TransNeRF outperforms comparing methods in the scene-agnostic scenario on all test scenes from synthetic and real datasets. It shows that our method has better generalization ability to novel scenes though it is trained on datasets with noticeably different scenes and view distributions. The superior generalization ability of our method are also reflected in qualitative results. As shown in Fig. 4, we examine the performance of the comparing methods on rendering the same randomly-selected test view when receiving different source-view sets, and the ground-truth rendering view is displayed in Fig. 2. In Fig. 4 the results of comparing approaches contain more obvious artifacts than our approach and even completely fail in the $S_3$ scenario, where the difference between the given source views and the target rendering view is more considerable than that in $S_1$ and $S_2$ scenarios. As depicted in colored boxes, the comparing methods cannot synthesize clean boundary of fronds and recover thin structures (branches and leaves besides fern). The artifacts become worse with the increment of the difference between the given source views and the target rendering view.

From the above qualitative results, we observe that there exists a gradual decline in the quality of the synthesized view when the difference between source views and the target rendering view increases from $S_1$ to $S_3$. Similarly, in quantitative results from $S_1$ to $S_3$, PSNR and SSIM values both decrease while LPIPS increases for all comparing methods. It reveals that the more dissimilar the source views are with respect to the target rendering view, the more difficult novel view synthesis becomes. The tables also indicate that the advantage of TransNeRF becomes more significant than the comparing methods with the increase in the difference between source views and the target rendering view. It demonstrates that TransNeRF has a stronger capability to explore complicated relationships between source views and the target view and learn a better scene representation in challenging scenarios. Please see the supplemental material for more results.

2) Per-scene Finetuning Experiments: In the per-scene finetuning scenario, each pre-trained model of comparing methods (PixelNeRF [22], MVSNeRF [23], IBRNet [2]) and our TransNeRF is finetuned for each scene in the evaluation dataset [33].

As shown in Tab. III and IV TransNeRF outperforms other comparing methods after per-scene finetuning. Similar to scene-agnostic results, per-scene finetuning results further validate that TransNeRF can learn more satisfactory novel view rendering than the comparing methods in the difference source-view scenarios. Meanwhile, the performance gap between TransNeRF and comparing methods becomes more extensive in contrast with that in the scene-agnostic setting, indicating that per-scene finetuning can further fulfill the potential of our TransNeRF. Similar to quantitative results, Fig. 5 illustrates that TransNeRF can achieve more realistic view synthesis results with fewer artifacts in comparison with baseline approaches.

V. Conclusion

This paper proposes a unified Transformer-based framework to learn a general radiance field for novel view synthesis, which achieves state-of-the-art accuracy on both scene-agnostic and per-scene finetuning experimental settings for real and synthetic datasets. The proposed framework naturally utilizes the attention mechanism to integrate information of the projected 2D pixels in the surrounding source views and the neighboring 3D points along the query ray. Meanwhile our framework explores and fuses deep and complicated relationships between observed views and the rendering view which is ignored in the previous MLP-based NeRFs.

References
D. Wang, X. Cui, X. Chen, Z. Zou, T. Shi, S. Salcudean, Z. J. Wang, and R. Ward, “Multi-view 3d reconstruction with transformers,” in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2021, pp. 5722–5731.

L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, “Occupancy networks: Learning 3d reconstruction in function space,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4460–4470.

K. Genova, F. Cole, A. Sud, A. Sarna, and T. Funkhouser, “Local deep implicit functions for 3d shape,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 4857–4866.

J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, “Deepodfs: Learning continuous signed distance functions for shape representation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 165–174.

C. Jiang, A. Sud, A. Makadia, J. Huang, M. Nießner, T. Funkhouser et al., “Local implicit grid representations for 3d scenes,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6001–6010.

E. Insafutdinov and A. Dosovitskiy, “Unsupervised learning of shape and pose with differentiable point clouds,” Advances in neural information processing systems, vol. 31, 2018.

S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik, “Multi-view supervision for single-view reconstruction via differentiable ray consistency,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2626–2634.

V. Sitzmann, M. Zollhöfer, and G. Wetzstein, “Scene representation networks: Continuous 3d-structure-aware neural scene representations,” Advances in Neural Information Processing Systems, vol. 32, 2019.

M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, “Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 3504–3515.

A. Yu, V. Ye, M. Tancik, and A. Kanazawa, “pixelnerf: Neural radiance fields from one or few images,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4578–4587.

A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, and H. Su, “Msvnerf: Fast generalizable radiance field reconstruction from multi-view stereo,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 14124–14133.

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, 2021, pp. 15182–15192.

M. Marti-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 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 7210–7219.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” Advances in neural information processing systems, vol. 27, 2014.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell et al., “Language models are few-shot learners,” Advances in neural information processing systems, vol. 33, pp. 1877–1901, 2020.

N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in European Conference on Computer Vision. Springer, 2020, pp. 213–229.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” in International Conference on Learning Representations (ICLR), 2021.

M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever, “Generative pretraining from pixels,” in International Conference on Machine Learning. PMLR, 2020, pp. 1691–1703.

D. Wang, X. Cui, X. Chen, R. Ward, and Z. J. Wang, “Interpreting bottom-up decision-making of cnns via hierarchical inference,” IEEE Transactions on Image Processing, vol. 30, pp. 6701–6714, 2021.

T. Kajiya and B. P. von Herzen, “Ray tracing volume densities,” ACM SIGGRAPH computer graphics, vol. 18, no. 3, pp. 165–174, 1984.

B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, “Local light field fusion: Practical view synthesis with prescriptive sampling guidelines,” ACM Transactions on Graphics (TOG), 2019.

J. Flynn, M. Broxton, P. E. Debevec, M. DuVall, G. Fyffe, R. S. Overbeck, N. Snavely, and R. Tucker, “Deepview: View synthesis with learned gradient descent.” in CVPR, 2019.

T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, “Stereo magnification: learning view synthesis using multipleplane images,” ACM Transactions on Graphics (TOG), vol. 37, no. 4, pp. 1–12, 2018.

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

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

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