Pacific Graphics 2023
R. Chaine, Z. Deng, and M. H. Kim
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 7

CP-NeRF: Conditionally Parameterized Neural Radiance Fields for Cross-scene Novel View Synthesis

Hao He[†1,2], Yixun Liang[†1], Shishi Xiao[1], Jierun Chen[2], and Yingcong Chen[‡1,2]

[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]The Hong Kong University of Science and Technology

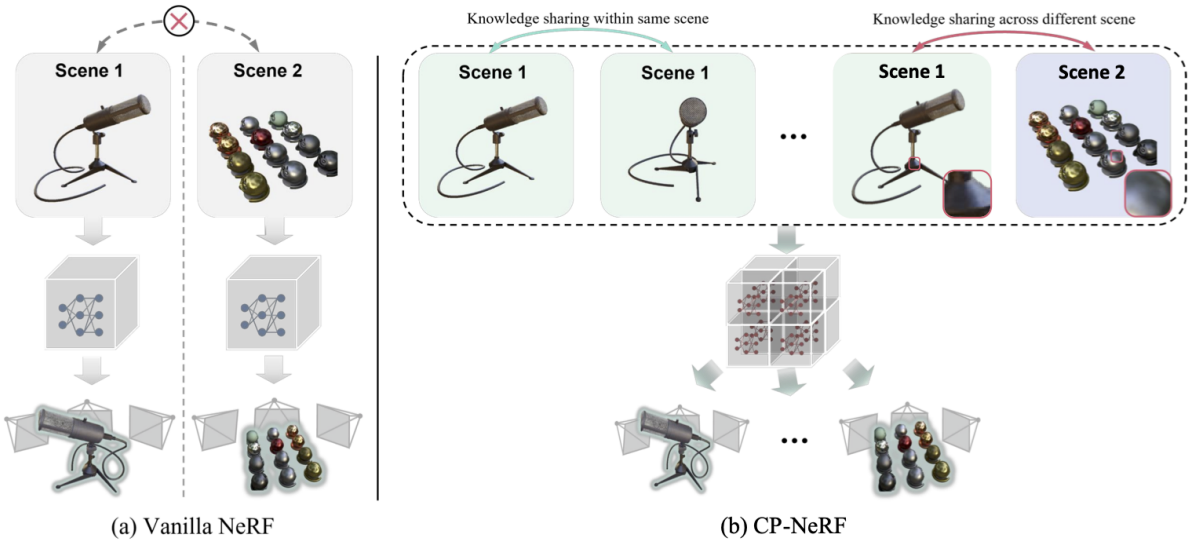

Figure 1: *Knowledge sharing between scenes: a) Vanilla NeRF overfits on a single scene and can not leverage shared information across different scenes. b) Our proposed CP-NeRF is able to use contextual information within and across scenes through a HyperNetwork.*

Abstract

Neural radiance fields (NeRF) have demonstrated a promising research direction for novel view synthesis. However, the existing approaches either require per-scene optimization that takes significant computation time or condition on local features which overlook the global context of images. To tackle this shortcoming, we propose the Conditionally Parameterized Neural Radiance Fields (CP-NeRF), a plug-in module that enables NeRF to leverage contextual information from different scales. Instead of optimizing the model parameters of NeRFs directly, we train a Feature Pyramid hyperNetwork (FPN) that extracts view-dependent global and local information from images within or across scenes to produce the model parameters. Our model can be trained end-to-end with standard photometric loss from NeRF. Extensive experiments demonstrate that our method can significantly boost the performance of NeRF, achieving state-of-the-art results in various benchmark datasets.

CCS Concepts

• ***Computing methodologies*** → ***Image-based rendering; 3D imaging;***

1. Introduction

Novel view synthesis aims to generate photo-realistic novel viewpoints of a scene from sparse 2D observations [TTM*22, LH96,

† Equal Contribution
‡ Corresponding Author

MST*21, MBRS*21]. Early works tackled this problem with either 3D structures inference [WSK*15, SHN*19, MON*19] or image-based rendering [SMKLM15, HPP*18]. However, image-based rendering may generate novel views with non-neglectable artifacts, while 3D structures inference is expensive in nature to scale to high-resolution images as it requires cubic (n^3) memory for computation. Recently, Neural Radiance Field (NeRF) [MST*21] has shown a promising direction, achieving high-quality results with affordable memory and computational costs. Using a differentiable volumetric render, NeRF enables photo-realistic novel view synthesis at high resolution with several multi-layer perception (MLP) networks [HTFF09].

Despite the appealing performance, NeRF requires a lengthy training process for every scene, which is costly when many scenes are optimized. Besides, as each scene is optimized individually, NeRF can not leverage shared information across different scenes, as shown in Fig. 1. Consequently, each scene requires many training views to guarantee a smooth transition. Several prior works [WWG*21, YYTK21, TY21] tried to resolve this shortcoming by disentangling the geometry and color information by conditioning on local features. However, these methods ignored the global context of images, which are proven to be useful in this paper. Moreover, some of these works relied on pre-trained models [YYTK21, WWG*21], which may not work well when query scene images lay outside the training space.

To address the aforementioned problems, we propose the Conditionally Parameterized Neural Radiance Fields (CP-NeRF), a plug-in module that enables training NeRF with multiple scenes. Instead of optimizing NeRF weights directly, we train a *HyperNetwork* [HDL16] that produces the weights based on training images. The benefits are manifolds. Firstly, our HyperNetwork takes a whole training image as input, and thus it can leverage global information of different locations. Besides, our HyperNetwork can learn shared knowledge of different scenes by jointly optimizing them. By fully using contextual information within and across scenes, our module significantly improves the performance of NeRF.

Notably, bring the idea from [HDL16, LW19, SZW19, VOHSG19, GW20] that stacking a HyperNetwork on top of NeRF network can introduce inferior performance since HyperNetwork transforms input images from a high dimensional space to a low dimensional latent space. As a result, when we use prior knowledge to predict NeRF weights, fine-grained information might be lost during such transformation and degrades the quality of the generated novel view. In this paper, we make some key contributions to tailor HyperNetwork for CP-NeRF. Firstly, comparing with [HDL16, LW19, SWT*20, VOHSG19, SZW19, ATM*22] that directly generate weights with a few layers, we propose a Feature Pyramid hyperNetwork (FPN) that could jointly extract both global and local scene information to predict NeRF weights and refine local details of novel views. Note that without our feature pyramid mechanism, the HyperNetwork produces weights with a global feature with limited bandwidth and thus may fail to carry detailed information, resulting in blurry artifacts when rendering high-resolution images. Our proposed FPN leverages intermediate features to extract local information, significantly improving high-resolution images' fidelity. In addition, our HyperNetwork also leverages a weighted aggregation module to avoid misalignment in local features. Finally, we propose a Global View-aware Attention module to adaptively aggregate information on the training views. This avoids floating artifacts caused by non-consistency in far-away viewpoints. Our contribution can be summarized as follows:

- We propose a meta-learning-based one-for-all NeRF framework that takes multi-view inputs as a prior through a HyperNetwork. By conditioning on scene-specific global and local information, our network can generalize to many scene-specific models.
- Our proposed method can be easily plugged into other NeRF-like networks as a module, improving the quality of their generated novel views and enabling them to generalize to multiple diverse scenes.
- Extensive experiments show that our method surpasses existing single-scene baselines. In addition, we outperform other local conditioned methods.

2. Related Works

Novel View Synthesis. Novel view synthesis is a long-standing problem in computer vision, where the goal is to infer an unseen point of view from a sparse set of 2D observations [TTM*22]. Early works without knowing 3D structures do not reach the photorealism of a scene and require dense viewpoints to generate a novel view of a scene [LH96, DTM96, GGSC96]. The voxel-based method overcomes such an issue by explicitly defining a 3D structure but is limited by spatial resolution due to high memory consumption [BLRW16, LDG18, SG18, WWX*17]. Others investigate using implicit representation which maps *xyz* coordinates to a signed distance function [MON*19, PFS*19, NMOG20, CLI*20, SHN*19].

Recent works show a promising research direction by encoding a 3D scene as neural radiance fields [MST*21]. Such methods use a neural network to represent a scene's geometry and color over its coordinates; using a differentiable render can generate photo-realistic novel views with a sparse set of 2D observations. Since a scene is represented with a couple of multi-layer perceptron (MLP) networks, their model size is very compact compared to other traditional networks. However, one drawback of such an approach is that they must overfit each scene with a model, thus requiring a substantial amount of optimization time. To overcome this shortcoming, various techniques have been proposed in recent literature. Plenoxels [FKYT*22] utilizes spherical harmonics functions to expedite optimization time. InstanceNGP [MESK22] deploys compact multilayer perceptrons (MLPs) in tandem with hash-encoding to achieve more efficient training. Similarly, PERF [RSA22] employs Gauss-Newton approximations as second-order derivatives, providing an alternative to traditional optimizers like Adam. However, it should be emphasized that the scope of these methods is generally limited to optimization on single scenes.

Some prior works have addressed the generalization issue in NeRF. PixelNeRF [YYTK21] used a 2D CNN to extract features from different viewpoints on each ray point sampled. By conditioning 2D image features along the ray, it learns the prior over the space of the radiance field. IBRNet [WWG*21] leverages a similar idea by using a 2D UNET to extract image features from neighboring views and aggregating image features using a ray transformer

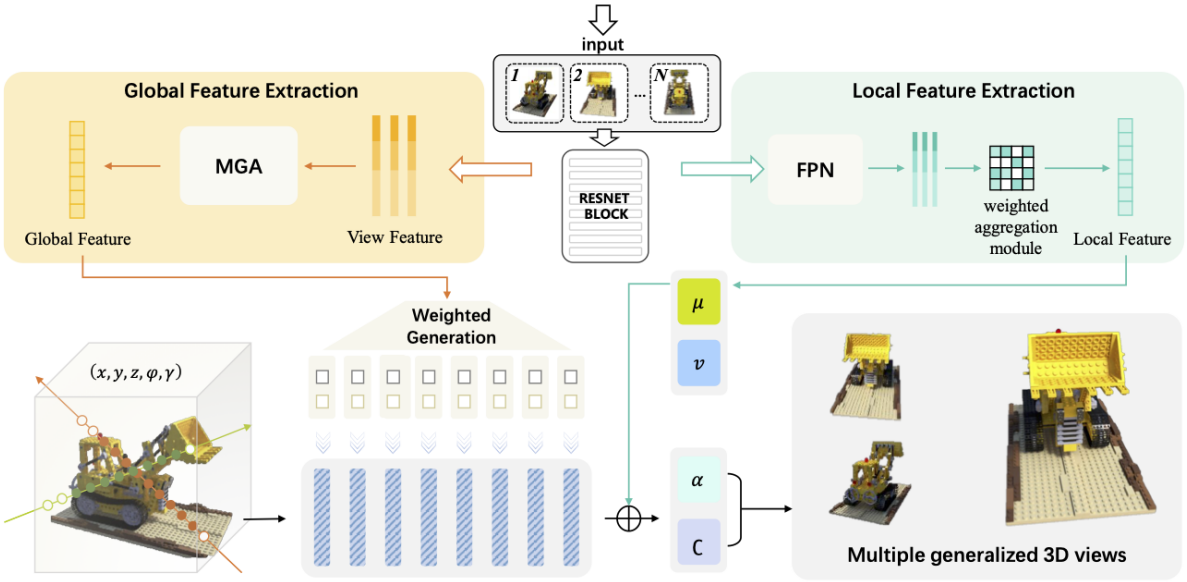

Figure 2: *The overview of CP-NeRF architecture. a) We first identify N closest neighboring source views (e.g., views labeled 1, 2,..., N). These views' features are extracted using an encoder. b) The extracted features are aggregated through a multi-head global view-aware attention (MGA) module, resulting in a fused, view-dependent global feature vector. c) Next, the view-dependent global feature is input into a HyperNetwork to generate weight and bias for coarse and fine networks. d) Finally, For each ray queried in the novel view, we compute the local features from the neighboring views using the FPN module and concatenate its corresponding information to generate a target view.*

to disentangle a scene's geometry and color. NeRFusion [ZBS*22] extends the idea by creating a 3D volumetric feature space by fusing the local features using a recurrent connection. However, these methods focus on conditioning the neural radiance field on local features and overlook the features from the global level. In contrast, our approach combines scene information from the global and local levels, which enhances the quality of the generated novel view.

HyperNetworks. HyperNetwork is introduced in [HDL16], and has a wide range of applications in the meta-learning-based framework [NWH21, BLRW17]. The core idea is to use two networks f and g, where f is trained to output the weights of network g. Network g is used for a specific task without seeing the data in f. Thus, the scene information used in f is directly embedded in the optimization space of network g. In the 3D scene representation field, HyperCube [PMTS21] and HyperCloud [SWT*20] produce explicit 3D shape representations like voxel or mesh by HyperNetwork. Littwin et al. [LW19] leverage a HyperNetwork to produce point cloud estimation conditioned on single image inputs, where the network takes coordinate inputs and conditions on scene image. However, since discrete explicit representations limit spatial resolution and do not show scene surfaces smoothly, Sitzmann et al. [SZW19, SCT*20, SMB*20] introduced the implicit continuous 3D representation of a learned prior on SDF within a category. Once the geometry is reconstructed, a HyperNetwork renders category-specific color. However, these methods either overfit a single 2D image or require further optimization at inference time. Our method overcomes such issues by using multiple neighboring views with self-attention to fuse their global embeddings, which can be applied to a wider range of scenes and does not require test time optimization.

3. Methodology

At the core of our proposed system, we try to solve the issue that NeRF network only focuses on local patches of a scene and does not have the ability to learn shared knowledge within or across scenes. To overcome these issues, we present CP-NeRF, an end-to-end multi-scene aware neural radiance field network that leverages a HyperNetwork to inject scene-specific global and local information as prior knowledge to condition on the network's parameter space, as shown in Fig. 2. Our system can be divided into three key components: a) a backbone encoder extracts global features from a set of source views when given a target view in a scene, b) a weight generation network that injects scene-specific global knowledge into a network's optimization space, c) a weighted aggregation FPN module that extracts local features from multi-view inputs, and conditions on the NeRF representation. We will introduce them in detail in the following sections.

3.1. Hypernetwork and encoder

Hypernetwork is introduced in [HDL16,LW19,SZW19] that injects prior knowledge to produce learnable parameters for a target network. Therefore, we define the process as an encoder that takes a set of input images and maps the image representations into parameters of a target network. Let's denote E as the backbone encoder

that generates global prior knowledge and F as the NeRF network that produces neural radiance fields.

To learn the scene-specific global feature representation $\mathbf{Z}$, we first identify a set of viewpoints that are closer to the novel target view based on their camera projections. Let's denote $X = \{X_i\}^{i=1,\dots n}$ as a set of observations of a scene. To render a novel view X_j, we first select the top–k closest viewpoints and extract their features as view-dependent global feature representation, denoted as Z_k. The selection of k depends on the GPU memory and the distribution of the viewpoint locations. Our observation is that a source view that is occluded from the target view does not contribute to the view-dependent global features and introduces unwanted artifacts, and we empirically found that k between 2 to 4 works best for our proposed method.

Let $\mathbf{X_i} \in [0,1]^{H_i \times W_i \times 3}$ denote the i-th source view, and E denote the backbone encoder used to extract the global feature $Z_i \in \mathbb{R}^d$ from each image. Therefore,

$$Z_i = E(X_i), \text{ where } i = 1,\dots,k \text{ and } Z_i \in \mathbb{R}^d. \quad (1)$$

Therefore, $Z = (Z_1, Z_2, \dots, Z_k)$ represents the global features corresponding to the synthesis of the novel view X_j. **Global Feature Fusion.** We observe that naively interpolating global features Z_k acquired from Eq. (1) to synthesize novel view X_j will have floating artifacts around the edge of the scene. This is caused by non-consistent feature aggregation in 3D space, as source views that are close to the novel view's position should weigh more when fusing the global features. Therefore, instead of applying equal weight for source view features, we adopt a learnable Global View Aware Attention (GVA) [VSP*17] module to fuse the generated global features, as shown in Fig. 3. To generate the novel view X_j, we first determine that $P_j \in \mathbb{R}^{3\times4}$ and $P_i \in \mathbb{R}^{3\times4}$ are the camera projection matrices for the novel view at j and the set of source views from 1 to k. The global feature Z for the novel view X_j can be written as

$$Z := \mathbf{Att}(Z_i, P_j, P_i), \text{where } i = 1,\dots k. \quad (2)$$

The global view aware attention learns to predict weights for incoming neighboring view features and produce one final target view-dependent global feature $Z \in \mathbb{R}^d$ that will be used as a prior for the weight generation network, which will be described in §3.2.

3.2. Weight generation network

To incorporate the global scene-aware information, we use a weight generation network denoted as $\mathbb{H}$, a neural network used to inject prior knowledge into the learnable parameter space for the target network. For the fused global feature $Z \in \mathbb{R}^d$ from Eq. (2), $\mathbb{H}$ embeds such prior knowledge and parameterizes it to the optimization space of the NeRF network. Specifically, the parameters $\mathbf{W} \in \mathbb{R}^{d\times d}$ and $\mathbf{b} \in \mathbb{R}^d$ are generated by $\mathbb{H}$ conditioning on the global feature embedding Z. Therefore, it can be formulated as the following:

$$\mathbf{\Theta}(W_l, b_l) = \mathbb{H}(\mathbf{Z}) \quad (3)$$

where l is the number of layers of NeRF network. Since we are producing weights $\mathbf{W}$ and bias $\mathbf{b}$ of NeRF, the network $\mathbb{H}$ acts as a global scene operator and generates many network parameters

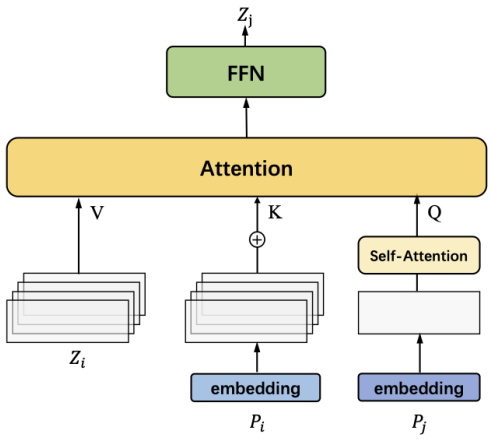

Figure 3: *Multi-head Global view aware Attention. The multi-head global view attention module aggregates view-dependent global feature vector Z_j of input neighboring view features Z_i given the input views' position and target view's position. For each input neighboring view, Z_i is calculated from encoder E. The generated global feature is then fed to the weight generation network to generate weights and bias.*

based on the global features. Therefore, instead of directly optimizing a neural radiance field of the 3D scene via NeRF, we dynamically adjust the NeRF parameters based on different scene inputs. Thus, we successfully disentangle the 3D space coordinates with their scene-specific information, as shown in Fig. 4.

Figure 4: *A t-SNE visualization of global embedding from the 8 scenes of the Real Forward Facing test set.*

3.3. Local Feature aggregation

Although the scene-specific global feature provides prior knowledge for modeling the 3D space, it transports source view image

data to a lower-dimensional latent space, which overlooks fine-grained details at the local level. Therefore, to incorporate local feature awareness, we add a feature pyramid network at blocks $2,3,4$ from our encoder E to generate a local feature map f with dimension $\mathbb{R}^{H_l \times W_l \times d}$. When querying a ray at a specific location (x,y,z) of the target view at j, we retrieve the local features $\zeta_i \in \mathbb{R}^d$, where $i = 1,\ldots,k$ via projecting the location from the target domain to the source domain and extract local features from the i-th feature maps by bilinear interpolation. Please see appendix for more details.

$$\zeta_i = f((x,y,z)_{j\to i}), \text{where } i = 1,...k. \tag{4}$$

Therefore, $\zeta = (\zeta_1, \zeta_2, ..., \zeta_k)$ denotes the local features from each source view when querying a specific location (x,y,z) at target view position j. To fuse the features at the local level while maintaining the 3D consistency, we aggregate each local feature with a weighted factor γ. The γ is calculated based on the relative distance between the target view and the source view camera position.

$$\gamma_i = Dist(\mathbf{P}_j, \mathbf{P}_i) = Dist(\frac{\mathbf{P}_j}{\|\mathbf{P}_j\|_2}, \frac{\mathbf{P}_i}{\|\mathbf{P}_i\|_2}), i \in \{1,...,k\}. \tag{5}$$

Specifically, we follow PointNet [QSMG17] design and extract mean and variance from the weighted local features obtained from Eq. (4) and Eq. (5), and pool them to generate weighted mean μ_w and σ_w^2. Finally, local features are integrated into the last two output layers in conjunction with the viewing parameter. The process can be formulated as the following:

$$(\mu_w, \sigma_w^2) = pool(\gamma_i \cdot \zeta_i), \text{where } i = 1,...k. \tag{6}$$

Therefore, the color and α at the specific 3D space can be represented as

$$(c, \alpha) = F(x,y,z,\mu_w,\sigma_w^2,\theta,\phi|\Theta). \tag{7}$$

Where F represents the NeRF network whose parameters are generated from Eq. (3) and view-dependent local features are conditioned from Eq. (6).

3.4. Rendering and Loss

We use the same volumetric rendering function as vanilla NeRF that computes color and density at the continuous 5D location. To render a color of a ray in a scene, we first query M samples along the ray and accumulate colors with its densities at the given location:

$$\tilde{C}(\mathbf{r}) = \sum_{k=1}^{M} T_k(1 - exp(-\sigma_k))\mathbf{c}_k. \tag{8}$$

$$\text{where } T_k = exp(-\sum_{j=1}^{k-1} \sigma_j). \tag{9}$$

We also use a hierarchical sampling strategy to predict the final RGB color. The hierarchical sampling first uniformly samples M_c points along the ray and renders through the coarse network. Given the coarse network's prediction, we then sample M_f points along the ray and render through the fine network. Therefore, total $M_c + M_f$ points are accumulated along each ray. Since both of the coarse and fine networks are optimizing the same ray in the same scene, the parameters of both networks are generated using the same HyperNetwork.

The final loss function can be formalized as following:

$$\mathscr{L} = \sum_{\mathbf{r}\in\mathbb{R}} \left[\|\tilde{C}_c(\mathbf{r}) - C(\mathbf{r})\|_2^2 + \|\tilde{C}_f(\mathbf{r}) - C(\mathbf{r})\|_2^2 \right]. \tag{10}$$

3.5. Implementation details.

We use ResNet-34 [HZRS16] as our backbone encoder for global feature extraction. However, the backbone encoder can be any other genetic network. We implement a feature pyramid network for local features with two additional up-sampling layers. We fuse the outputs from layer 4 to layer 2 of the FPN and produce one final feature map whose dimension is $H/4, W/4, 64$. The N views global information is aggregated using a global view aware attention module to produce one global feature whose dimension is $d = 512$, and local features are queried via the ray coordinates projected to the N views' local feature map. To render a novel view, we first query its N nearest neighboring views via camera pose information. We create the training pairs with (target view and reference views). The reference views are then sent to the backbone encoder to produce global features and generate NeRF network weights and bias. The target view's coordinates are then sent to the generated NeRF network and the local features queried from the local feature map for final prediction. We train our framework end to end with Adam optimizer [KB14]. The base learning rate is 5×10^{-4}, decaying exponentially along with the optimization steps. Our model is trained on eight NVIDIA RTX 3090Ti GPUs with a batch size of 2000 to 5120 rays, depending on image resolution. We train our network for 400k iterations, which takes about a day to finish. We provide more implementation details in supplementary materials.

4. Experiments

In this section, we conduct experiments to evaluate the effectiveness of our proposed framework. First, we introduce the benchmarks and evaluation metrics. Next, we compare our model to other baselines. Finally, we give a detailed analysis of our design choice and the effectiveness of our method by ablation studies. More analysis and visualizations are provided in supplementary pages.

Datasets and metrics. We evaluate our method on two different datasets from NeRF *Realistic synthetic 360°* and *Real forward-facing* [MST*21]. For *Realistic synthetic 360°* dataset, it has eight different scenes and each of which consists of complicated geometry and non-Lambertian materials. Objects in the synthetic dataset have 100 training views and 200 test views; each views have 800 $\times$ 800 image resolution rendered 360°from either the upper hemisphere or the full sphere. For the *Real forward-facing* dataset, we use the LIFF [MSOC*19] benchmark that has 35 real scenes from cellphone capture for pretraining, then finetuning on NeRF *real forward facing* dataset. The latter one contains eight complex scenes, each captured with 20 to 62 images at image resolution 1008 $\times$ 756. We also follow the same setup as NeRF that $1/8$th of the images are held out as testset. Such a setup is to show that our method

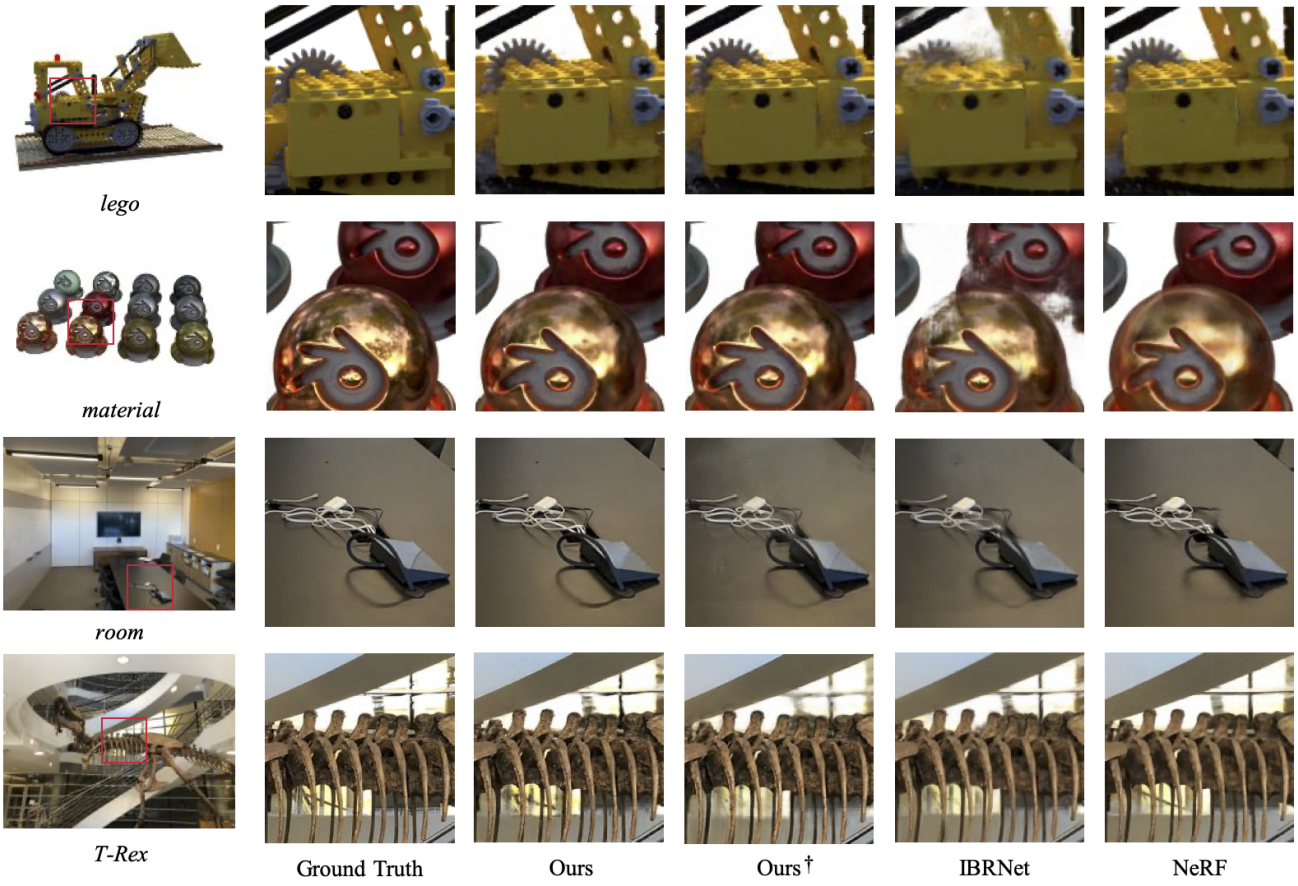

Figure 5: *Qualitative comparison on Realistic Synthetic 360°and Real Forward-Facing [MST*21] dataset. Visualization of generated novel views. IBRNet [WWG*21] has difficulties when input views are sparse, resulting in floating artifacts around object boundaries. NeRF [MST*21] rendered objects are missing texture information, and some fine details are not well recovered (see T-Rex). Our method can leverage global and local information. Thus, it produces high-quality novel views. † denotes the method with pre-trained model.*

can learn shared knowledge cross various complex scenes. To evaluate the visual quality of generated novel views, the mean value of the Peak-Signal-to-Noise Ratio (PSNR), Structural Similarity in Images (SSIM) [WBSS04] and LPIPS [ZIE*18] perceptual metric are reported across all scenes in each of the two datasets.

Baselines and settings. We compare our method with two different baselines, vanilla and local condition-based neural radiance field networks. For the vanilla neural radiance field network, we compare ours with NeRF [MST*21], which requires per-scene optimization for separate networks, and it does not generalize at all. We also compare ours with the local conditioned neural radiance field network IBRNet [WWG*21], which uses local patches to blend neighboring views into novel viewpoint. Both ours and IBRNet are designed to optimize multiple scenes together. However, the major difference between ours and IBRNet is we inject prior knowledge into the optimization space instead of feeding it directly as input.

We evaluate our methods in two ways: a) we directly train all the scenes together without finetuning specific scenes. Note that NeRF has to optimize on a single scene. In addition, IBRNet requires a large dataset for pretraining. b) We then finetune our model on each specific scene for a fair comparison between our proposed method and baselines. Results are shown in Tab. 1. In addition, we also show that when our method is plugged in as a module to the existing baselines, such as NeRF and mip-NeRF [BMT*21], it will also improve the quality of their generated novel views and enable them to optimize multiple diverse scenes with a single model. We show the results in Results Tab. 2.

4.1. Results

We sample 3 nearest source views from the training dataset to render a novel view from the evaluation dataset. We report our result in Tab. 1.

Quantitative comparisons. Tab. 1 shows that when all scenes are trained together, our model consistently outperforms IBRNet on

Method	Settings	Realistic Synthetic 360°			Real Forward-Facing		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
PixelNeRF [YYTK21]	No Per-Scene Optimization	22.65	0.808	0.202	18.66	0.588	0.463
IBRNet [WWG*21]		25.49	0.916	0.100	25.13	**0.817**	0.205
MVSNeRF [CXZ*21]		23.62	0.897	0.176	21.93	0.795	0.252
Ours		**29.54**	**0.921**	**0.092**	**25.41**	0.766	**0.199**
NeRF [MST*21]	Per-Scene Optimization	31.01	0.947	0.081	26.50	0.811	0.250
IBRNet [WWG*21]		28.14	0.942	0.072	26.73	**0.851**	0.175
MVSNeRF [CXZ*21]		27.07	0.931	0.168	25.45	0.877	0.192
Ours_ft		**31.77**	**0.949**	**0.063**	**27.23**	0.812	**0.136**

Table 1: *Quantitative comparison: Realistic Synthetic 360°and Real Forward-Facing [MST*21] with baselines.*

Method	Lego			Materials		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRF [MST*21]	32.54	0.961	0.050	29.62	**0.949**	0.063
NeRF + Ours	**33.35(+0.81)**	**0.966**	**0.016**	**30.02(+0.4)**	0.945	**0.026**
mip-NeRF* [BMT*21]	33.72	0.969	0.035	29.86	**0.954**	0.046
mip-NeRF* + Ours	**34.48(+0.76)**	**0.969**	**0.014**	**30.68(+0.82)**	0.953	**0.025**

Table 2: *Quantitative comparison: When our method is plugged in as a module to NeRF [MST*21] and mip-NeRF [BMT*21], we show results on lego and materials from Real synthetic 360° [MST*21] dataset. * denotes stopped early from [BMT*21].*

both synthetic and real forward-facing datasets. After finetuning on each specific scene, our method also outperforms the per-scene optimized neural radiance field method and achieves better PSNR, SSIM, and LPIPS. Such results show that our model is robust and consistently generates photo-realistic images on synthetic and real datasets. Note that in the *Realistic synthetic 360°* dataset, training views are sparsely captured from either the upper hemisphere or the full sphere. Therefore, methods like IBRNet suffers under such condition simply because local information is insufficient for generating a novel view, as shown in Fig. 5. However, ours can leverage shared knowledge across different viewpoints and outperform per-scene-optimized methods.

In addition to the *Real forward-facing* dataset, some of the scenes only have a limited number of training views. Our method can leverage shared knowledge across different scenes and produce high-quality novel views compared to the one that does not explore global information at all or the local conditioned method that only sees patches of the image (see Fig. 6).

Qualitative comparisons. As shown in Fig. 5, our method can leverage contextual information within or across scenes, producing high-fidelity novel view images compared with other methods. As we can see, local condition methods such as IBRNet have difficulty generating novel views from sparse inputs, resulting in floating artifacts at the boundary of objects. A per-scene-optimized method such as NeRF requires a dense training point. Therefore, they produce unrealistic noise when the training set is limited. In contrast, our method can recover more information from either sparse inputs or limited training sets.

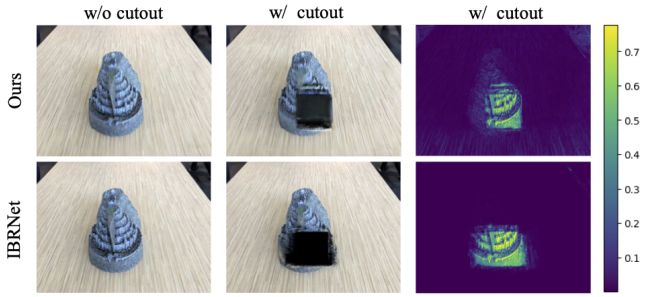

Figure 6: *Global features. When input source views are masked out at a specific region, the global feature is affected by the missing information, which results in blurring or missing regions for the generated novel view. Error map shows that mask is affecting the whole object as our method explores global contextual information, where IBRNet only affects mask region as they only condition on local patches.*

4.2. Ablation study

In this section, we first conduct experiments to validate our design choice of each individual module on *Real synthetic 360°* dataset [MST*21]. We first remove our FPN module that aggregates local features when queried on specific coordinates. Therefore, the model only conditions the global features of neighboring views. Removing the FPN module hinders the overall quality of generated novel views. We then remove the GVA module and put equal weights on each neighboring view. Averaging the global features causes the generated novel view to have blurring artifacts on the edge of the object, which is caused by non-consistency in far away

viewpoints (please see supplementary material for more details). Note that if we remove the HyperNetwork, our method degrades to vanilla NeRF and does not generalize across different scenes. We report the result on Tab. 3.

Model	Realistic Synthetic 360°		
	PSNR↑	SSIM↑	LPIPS↓
w/o Local	31.24	0.941	0.069
w/o GVA	30.92	0.936	0.073
Ours	**31.77**	**0.949**	**0.063**

Table 3: *Ablation study: Real synthetic 360° [MST*21] data. The results are from the per-scene optimized model by taking an average of 8 different scenes.*

Sensitivity to global features. We investigate how the global feature is affecting our overall performance in *Real forward-facing* [MST*21] dataset. We show in Fig. 6 that if N source views are masked out at a specific region, the generated novel view will have a blurring patch around the masked region. Notice that when all input source views are masked out at a specific region, it affects the entire object in the generated novel views in our method. Whereas local condition methods such as IBRNet only affect the masked region. This demonstrates that our method can explore global contextual information for novel view synthesis.

Plugging in as a Module. We show in Tab. 2 that when our method is plugged in as a module to the per-scene optimized methods, we can enable them to learn shared knowledge across scenes and enhance their overall performance. As a result, those per-scene optimized methods can now train only one model for all scenes. Such a result demonstrates the effectiveness of our proposed method and its potential to be practically used.

Sensitivity to the number of neighboring views. We conduct experiments to investigate how the number of source views affects the quality of generated novel views. The result is shown in Fig. 7 that simply blending more source views does not enhance the quality of the generated novel view. This result also demonstrates that if a source view is far from the novel viewpoint, it does not contribute much globally and locally.

Computation Metrics. We comprehensively examined the computational cost of our proposed method and compared it with baseline approaches, as delineated in Tab. 4. Remarkably, our method demands significantly fewer Floating Point Operations Per Second (FLOPs) than the baseline methods, reducing computational requirements by factors of 6, 10, and 12 compared to IBRNet, NeRF, and PixelNeRF, respectively. This efficiency gain is primarily attributed to our weight generation module, which leverages shared knowledge across scenes. Specifically, for a set of N query points in a given scene, the forward pass through the weight generation module occurs only once to produce the global embedding. Subsequently, the predicted NeRF network is executed for each query point individually. Additionally, our methodology stands apart by its capacity for a singular model to generate representations for N distinct scenes. This starkly contrasts NeRF, which necessitates N distinct models for N unique scenes. Consequently, after the initial training phase, which is required only once, our approach necessitates only a limited number of iterations for finetuning each specific scene.

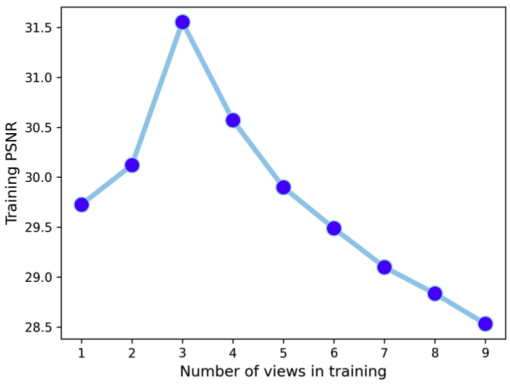

Figure 7: *Selection of the number of neighboring views. We select $k = 1,...9$) during training on Real forward-facing dataset and report their training PSNR.*

Model	GFLOPs↓	Finetune Iteration
NeRF	121.6	N/A
PixelNeRF	146.6	N/A
IBRNet_ft	77.85	500
Ours_ft	**11.90**	500

Table 4: *Computation Metrics: All FLOPs are calculated for rendering a 400x400 novel view. We use 5 neighbor views to render one novel view and report its GFLOPs [Sov23]. Finetune Iteration denotes per-scene finetuning from the pretraining.*

5. Conclusion

In this paper, we propose the CP-NeRF, a conditionally parameterized neural radiance field that learns global and local features via a HyperNetwork. Our major advantage is that we can fully leverage contextual information across different scales. We also show that when our method is plugged in as a module to NeRF, we enable them to generalize on multiple diverse scenes and enhance its overall performance. Experimental results show that our proposed method outperforms baselines and produces state-of-the-art rendering quality on novel view synthesis for both real and synthetic data.

6. Acknowledgment

We would like to thank the Turing AI Computing Cloud (TACC) [XWW*21] and HKUST iSING Lab for providing us with computation resources on their platform.

References

[ATM*22] ALALUF Y., TOV O., MOKADY R., GAL R., BERMANO A.: Hyperstyle: Stylegan inversion with hypernetworks for real image edit-

ing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 18511–18521. 2

[BLRW16] BROCK A., LIM T., RITCHIE J. M., WESTON N.: Generative and discriminative voxel modeling with convolutional neural networks. *arXiv preprint arXiv:1608.04236* (2016). 2

[BLRW17] BROCK A., LIM T., RITCHIE J. M., WESTON N.: Smash: one-shot model architecture search through hypernetworks. *arXiv preprint arXiv:1708.05344* (2017). 3

[BMT*21] BARRON J. T., MILDENHALL B., TANCIK M., HEDMAN P., MARTIN-BRUALLA R., SRINIVASAN P. P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 5855–5864. 6, 7

[CLI*20] CHABRA R., LENSSEN J. E., ILG E., SCHMIDT T., STRAUB J., LOVEGROVE S., NEWCOMBE R.: Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *European Conference on Computer Vision* (2020), Springer, pp. 608–625. 2

[CXZ*21] CHEN A., XU Z., ZHAO F., ZHANG X., XIANG F., YU J., SU H.: Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 14124–14133. 7

[DTM96] DEBEVEC P. E., TAYLOR C. J., MALIK J.: Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), pp. 11–20. 2

[FKYT*22] FRIDOVICH-KEIL S., YU A., TANCIK M., CHEN Q., RECHT B., KANAZAWA A.: Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 5501–5510. 2

[GGSC96] GORTLER S. J., GRZESZCZUK R., SZELISKI R., COHEN M. F.: The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), pp. 43–54. 2

[GW20] GALANTI T., WOLF L.: On the modularity of hypernetworks. *Advances in Neural Information Processing Systems 33* (2020), 10409–10419. 2

[HDL16] HA D., DAI A., LE Q. V.: Hypernetworks. *arXiv preprint arXiv:1609.09106* (2016). 2, 3

[HPP*18] HEDMAN P., PHILIP J., PRICE T., FRAHM J.-M., DRETTAKIS G., BROSTOW G.: Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics 37*, 6 (2018), 1–15. 2

[HTFF09] HASTIE T., TIBSHIRANI R., FRIEDMAN J. H., FRIEDMAN J. H.: *The elements of statistical learning: data mining, inference, and prediction*, vol. 2. Springer, 2009. 2

[HZRS16] HE K., ZHANG X., REN S., SUN J.: Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 770–778. 5

[KB14] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014). 5

[LDG18] LIAO Y., DONNE S., GEIGER A.: Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 2916–2925. 2

[LH96] LEVOY M., HANRAHAN P.: Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), pp. 31–42. 1, 2

[LW19] LITTWIN G., WOLF L.: Deep meta functionals for shape representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 1824–1833. 2, 3

[MBRS*21] MARTIN-BRUALLA R., RADWAN N., SAJJADI M. S., BARRON J. T., DOSOVITSKIY A., DUCKWORTH D.: 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. 1

[MESK22] MÜLLER T., EVANS A., SCHIED C., KELLER A.: Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG) 41*, 4 (2022), 1–15. 2

[MON*19] MESCHEDER L., OECHSLE M., NIEMEYER M., NOWOZIN S., GEIGER A.: 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. 2

[MSOC*19] MILDENHALL B., SRINIVASAN P. P., ORTIZ-CAYON R., KALANTARI N. K., RAMAMOORTHI R., NG R., KAR A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG) 38*, 4 (2019), 1–14. 5

[MST*21] MILDENHALL B., SRINIVASAN P. P., TANCIK M., BARRON J. T., RAMAMOORTHI R., NG R.: Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM 65*, 1 (2021), 99–106. 1, 2, 5, 6, 7, 8

[NMOG20] NIEMEYER M., MESCHEDER L., OECHSLE M., GEIGER A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2020). 2

[NWH21] NIRKIN Y., WOLF L., HASSNER T.: Hyperseg: Patch-wise hypernetwork for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 4061–4070. 3

[PFS*19] PARK J. J., FLORENCE P., STRAUB J., NEWCOMBE R., LOVEGROVE S.: Deepsdf: 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. 2

[PMTS21] PROSZEWSKA M., MAZUR M., TRZCIŃSKI T., SPUREK P.: Hypercube: Implicit field representations of voxelized 3d models. *arXiv preprint arXiv:2110.05770* (2021). 3

[QSMG17] QI C. R., SU H., MO K., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 652–660. 5

[RSA22] RASMUSON S., SINTORN E., ASSARSSON U.: Perf: performant, explicit radiance fields. *Frontiers in Computer Science 4* (2022), 871808. 2

[SCT*20] SITZMANN V., CHAN E., TUCKER R., SNAVELY N., WETZSTEIN G.: Metasdf: Meta-learning signed distance functions. *Advances in Neural Information Processing Systems 33* (2020), 10136–10147. 3

[SG18] STUTZ D., GEIGER A.: Learning 3d shape completion from laser scan data with weak supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 1955–1964. 2

[SHN*19] SAITO S., HUANG Z., NATSUME R., MORISHIMA S., KANAZAWA A., LI H.: Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 2304–2314. 2

[SMB*20] SITZMANN V., MARTEL J., BERGMAN A., LINDELL D., WETZSTEIN G.: Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems 33* (2020), 7462–7473. 3

[SMKLM15] SU H., MAJI S., KALOGERAKIS E., LEARNED-MILLER E.: Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision* (2015), pp. 945–953. 2

[Sov23] SOVRASOV V.: ptflops: a flops counting tool for neural networks in pytorch framework, 2018-2023. 8

[SWT*20] SPUREK P., WINCZOWSKI S., TABOR J., ZAMORSKI M., ZIĘBA M., TRZCIŃSKI T.: Hypernetwork approach to generating point clouds. *arXiv preprint arXiv:2003.00802* (2020). 2, 3

[SZW19] SITZMANN V., ZOLLHÖFER M., WETZSTEIN G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems 32* (2019). 2, 3

[TTM*22] TEWARI A., THIES J., MILDENHALL B., SRINIVASAN P., TRETSCHK E., YIFAN W., LASSNER C., SITZMANN V., MARTIN-BRUALLA R., LOMBARDI S., ET AL.: Advances in neural rendering. In *Computer Graphics Forum* (2022), vol. 41, Wiley Online Library, pp. 703–735. 1, 2

[TY21] TREVITHICK A., YANG B.: 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. 2

[VOHSG19] VON OSWALD J., HENNING C., SACRAMENTO J., GREWE B. F.: Continual learning with hypernetworks. *arXiv preprint arXiv:1906.00695* (2019). 2

[VSP*17] VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł., POLOSUKHIN I.: Attention is all you need. *Advances in neural information processing systems 30* (2017). 4

[WBSS04] WANG Z., BOVIK A. C., SHEIKH H. R., SIMONCELLI E. P.: Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing 13*, 4 (2004), 600–612. 6

[WSK*15] WU Z., SONG S., KHOSLA A., YU F., ZHANG L., TANG X., XIAO J.: 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 1912–1920. 2

[WWG*21] WANG Q., WANG Z., GENOVA K., SRINIVASAN P. P., ZHOU H., BARRON J. T., MARTIN-BRUALLA R., SNAVELY N., FUNKHOUSER T.: Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 4690–4699. 2, 6, 7

[WWX*17] WU J., WANG Y., XUE T., SUN X., FREEMAN B., TENENBAUM J.: Marrnet: 3d shape reconstruction via 2.5 d sketches. *Advances in neural information processing systems 30* (2017). 2

[XWW*21] XU K., WAN X., WANG H., REN Z., LIAO X., SUN D., ZENG C., CHEN K.: Tacc: A full-stack cloud computing infrastructure for machine learning tasks. *arXiv preprint arXiv:2110.01556* (2021). 8

[YYTK21] YU A., YE V., TANCIK M., KANAZAWA A.: 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. 2, 7

[ZBS*22] ZHANG X., BI S., SUNKAVALLI K., SU H., XU Z.: Nerfusion: Fusing radiance fields for large-scale scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 5449–5458. 3

[ZIE*18] ZHANG R., ISOLA P., EFROS A. A., SHECHTMAN E., WANG O.: 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. 6