SymmNeRF: Learning to Explore Symmetry Prior for Single-View View Synthesis

Xingyi Li[1], Chaoyi Hong[1], Yiran Wang[1], Zhiguo Cao[1], Ke Xian[2⋆], and Guosheng Lin[2]

[1] Key Laboratory of Image Processing and Intelligent Control, Ministry of Education
School of AIA, Huazhong University of Science and Technology, China
{xingyi_li,cyhong,wangyiran,zgcao}@hust.edu.cn
[2] S-lab, Nanyang Technological University
{ke.xian,gslin}@ntu.edu.sg

Abstract. We study the problem of novel view synthesis of objects from a single image. Existing methods have demonstrated the potential in single-view view synthesis. However, they still fail to recover the fine appearance details, especially in self-occluded areas. This is because a single view only provides limited information. We observe that man-made objects usually exhibit symmetric appearances, which introduce additional prior knowledge. Motivated by this, we investigate the potential performance gains of explicitly embedding symmetry into the scene representation. In this paper, we propose SymmNeRF, a neural radiance field (NeRF) based framework that combines local and global conditioning under the introduction of symmetry priors. In particular, SymmNeRF takes the pixel-aligned image features and the corresponding symmetric features as extra inputs to the NeRF, whose parameters are generated by a hypernetwork. As the parameters are conditioned on the image-encoded latent codes, SymmNeRF is thus scene-independent and can generalize to new scenes. Experiments on synthetic and real-world datasets show that SymmNeRF synthesizes novel views with more details regardless of the pose transformation, and demonstrates good generalization when applied to unseen objects. Code is available at: https://github.com/xingyi-li/SymmNeRF.

Keywords: Novel View Synthesis, NeRF, Symmetry, HyperNetwork

1 Introdution

Novel view synthesis is a long-standing problem in computer vision and graphics [4,9,20]. The task is to synthesize novel views from a set of input views or even a single input view, which is challenging as it requires comprehensive 3D understanding [36]. Prior works mainly focus on explicit geometric representations, such as voxel grids [14,21,32,42], point clouds [1,7], and triangle meshes [16,29,39]. However, these methods suffer from limited spatial resolution

⋆ Corresponding author

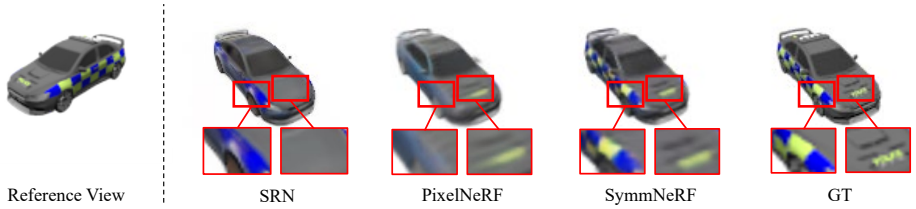

Fig. 1: Novel views from a single image synthesized by SRN [33], PixelNeRF [44] and our SymmNeRF. The competitive methods are prone to miss some texture details, especially when the pose difference between the reference view and target view is large. By contrast, SymmNeRF augmented with symmetry priors recovers more appearance details.

and representation capability because of the discrete properties. Recently, differentiable neural rendering methods [25,27,28,31,33,37] have shown great progress in synthesizing photo-realistic novel views. For example, neural radiance fields (NeRFs) [26], which implicitly encode volumetric density and color via multilayer perceptrons (MLPs), show an impressive level of fidelity on novel view synthesis. However, these methods usually require densely captured views as input and test-time optimization, leading to poor generalization across objects and scenes. To reduce the strong dependency on dense inputs and enable better generalization to unseen objects, in this paper, we explore novel view synthesis of object categories from only a single image.

Novel view synthesis from a single image is challenging, because a single view cannot provide sufficient information. Recent NeRF-based methods [13,44] learn scene priors for reconstruction by training on multiple scenes. Although they have shown the potential in single-view view synthesis, it is particularly challenging when the pose difference between the reference and target view is large (see Fig. 1). We observe that man-made objects in real world usually exhibit symmetric appearances. Based on this, a question arises: *can symmetry priors benefit single-view view synthesis?*

To answer this question, we explore how to take advantage of symmetry priors to introduce additional information for reconstruction. To this end, we present SymmNeRF, a NeRF-based framework that is augmented by symmetry priors. Specifically, we take the pixel-aligned image features and the corresponding symmetric features as extra inputs to NeRF. This allows reasonable recovery of occluded geometry and missing texture. During training, given a set of posed input images, SymmNeRF simultaneously optimizes a convolutional neural network (CNN) encoder and a hypernetwork. The former encodes image features, and generates latent codes which represent the coarse shape and appearance of unseen objects. The latter maps specific latent codes to the weights of the neural radiance fields. Therefore, SymmNeRF is scene-independent and can generalize to unseen objects. Unlike the original NeRF [26], for a single query point, we take as input its original and symmetric pixel-aligned image features besides its

3D location and viewing direction. At the inference stage, SymmNeRF generates novel views by feed-forward prediction without test-time optimization.

In the present paper, we investigate the potential performance gains by combining local and global conditioning under the introduction of the symmetry prior. To this end, we add the assumptions on the data distribution that objects are in a canonical coordinate frame. We demonstrate that such a symmetry prior can lead to significant performance gains. In summary, our main contributions are:

- We propose SymmNeRF, a NeRF-based framework for novel view synthesis from a single image. By introducing symmetry priors into NeRF, SymmNeRF can synthesize high-quality novel views with fine details regardless of pose transformation.
- We combine local features with global conditioning via hypernetworks and demonstrate significant performance gains. Note that we perform inference via a CNN instead of auto-decoding, *i.e.*, without test-time optimization, which is different from SRN [33].
- Given only a single input image, SymmNeRF demonstrates significant improvement over state-of-the-art methods on synthetic and real-world datasets.

2 Related Work

Novel View Synthesis. Novel view synthesis is the task of synthesizing novel camera perspectives of a scene, given source images and their camera poses. The key challenges are understanding the 3D structure of the scene and inpainting of invisible regions of the scene [11]. The research of novel view synthesis has a long history in computer vision and graphics [4,9,20]. Pioneer works typically synthesize novel views by warping, resampling, and blending reference views to target viewpoints, which can be classified as image-based rendering methods [4]. However, they require densely captured views of the scene. When only a few observations are available, ghosting-like artifacts and holes may appear [36]. With the advancement of deep learning, a few learning-based methods have been proposed, most of which focus on explicit geometric representations such as voxel grids [14,21,32,42], point clouds [1,7], triangle meshes [16,29,39], and multiplane images (MPIs) [8,38,46].

Recent works [3,25,27,28,37] show that neural networks can be used as an implicit representation for 3D shapes and appearances. DeepSDF [28] maps continuous spatial coordinates to signed distance and proves the superiority of neural implicit functions. SRN [33] proposes to represent 3D shapes and appearances implicitly as a continuous, differentiable function that maps a spatial coordinate to the local features of the scene properties at that point. Recently, NeRF [26] shows astonishing results for novel view synthesis, which is an implicit MLP-based model that maps 3D coordinates plus 2D viewing directions to opacity and color values. However, NeRF requires enormous posed images and must be independently optimized for every scene. PixelNeRF [44] tries to address this

issue by conditioning NeRF on image features, which are extracted by an image encoder. This enables its ability to render novel views from only a single image and its generalization to new scenes. Rematas *et al.* [30] propose ShaRF, a generative model aiming at estimating neural representation of objects from a single image, combining the benefits of explicit and implicit representations, which is capable of generalizing to unseen objects. CodeNeRF [13] learns to disentangle shape and texture by learning separate embeddings from a single image, allowing single view reconstruction and shape/texture editing. However, these methods usually struggle to synthesize reasonable novel views from a single image when self-occlusion occurs. In contrast, SymmNeRF first estimates coarse representations and then takes reflection symmetry as prior knowledge to inpaint invisible regions. This allows reasonable recovery of occluded geometry and missing texture.

HyperNetworks. A hypernetwork [10] refers to a small network that is trained to predict the weights of a large network, which has the potential to generalize previous knowledge and adapt quickly to novel environments. Various methods resort to hypernetworks in 3D vision. Littwin *et al.* [22] recover shape from a single image using hypernetworks in an end-to-end manner. SRN [33] utilizes hypernetworks for single-shot novel view synthesis with neural fields. In this work, we condition the parameters of NeRF on the image-encoded latent codes via the hypernetwork, which allows SymmNeRF to be scene-independent and generalize to new scenes.

Reflection Symmetry. Reflection symmetry plays a significant role in the human visual system and has already been exploited in the computer vision community. Wu *et al.* [41] infer 3D deformable objects given only a single image, using a symmetric structure to disentangle depth, albedo, viewpoint and illumination. Ladybird [43] assigns occluded points with features from their symmetric points based on the reflective symmetry of the object, allowing recovery of occluded geometry and texture. NeRD [47] learns a neural 3D reflection symmetry detector, which can estimate the normal vectors of objects' mirror planes. They focus on the task of detecting the 3D reflection symmetry of a symmetric object from a 2D image. In this work, we focus on exploring the advantages of explicitly embedding symmetry into the scene representation for single-view view synthesis.

3 SymmNeRF

3.1 Overview

Here we present an overview of our proposed method. We propose to firstly estimate holistic representations as well as symmetry planes, followed by fulfilling details, and to explicitly inject symmetry priors into single-view view synthesis of object categories. In particular, we design SymmNeRF to implement the ideas above. Fig. 2 shows the technical pipeline of SymmNeRF.

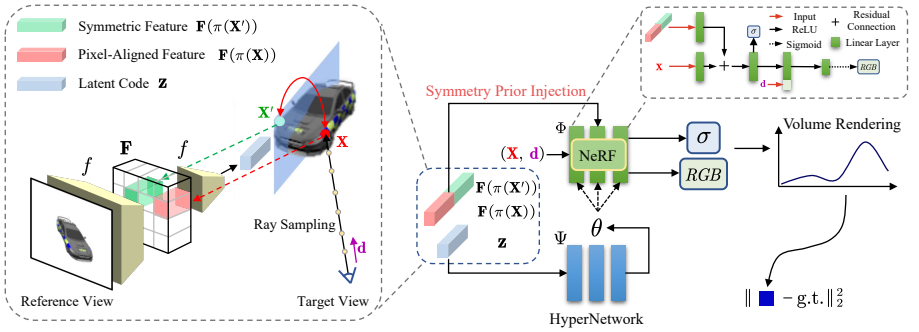

Fig. 2: An overview of our SymmNeRF. Given a reference view, we first encode holistic representations by estimating the latent code $\mathbf{z}$ through the image encoder network f. We then obtain the pixel-aligned feature $\mathbf{F}(\pi(\mathbf{X}))$ and the symmetric feature $\mathbf{F}(\pi(\mathbf{X}'))$, by projecting the query point $\mathbf{X}$ and the symmetric point $\mathbf{X}'$ to the 2D location on the image plane using camera parameters, followed by bilinear interpolation between the pixelwise features on the feature volume $\mathbf{F}$. The hypernetwork transforms the latent code $\mathbf{z}$ to the weights θ of the corresponding NeRF Φ. For a query point $\mathbf{X}$ along a target camera ray with viewing direction $\mathbf{d}$, NeRF takes the spatial coordinate $\mathbf{X}$, ray direction $\mathbf{d}$, pixel-aligned feature $\mathbf{F}(\pi(\mathbf{X}))$ and symmetric feature $\mathbf{F}(\pi(\mathbf{X}'))$ as input, and outputs the color and density.

Given a set of M instances training datasets $\mathcal{D} = \{\mathcal{C}_j\}_{j=1}^{M}$, where $\mathcal{C}_j = \{(\mathcal{I}_i^j, \mathbf{E}_i^j, \mathbf{K}_i^j)\}_{i=1}^{N}$ is a dataset of an instance object, $\mathcal{I}_i^j \in \mathbb{R}^{H\times W\times 3}$ refers to an input image, $\mathbf{E}_i^j = [\mathbf{R}|\mathbf{t}] \in \mathbb{R}^{3\times 4}$ and $\mathbf{K}_i^j \in \mathbb{R}^{3\times 3}$ are the corresponding extrinsic and intrinsic camera matrix respectively, and N denotes the number of the input images, SymmNeRF first encodes a holistic representation and regresses the symmetry plane for the input view. We then extract symmetric features along with pixel-aligned features for the sake of preserving fine-grained details observed in the input view. Subsequently, we transform the holistic representation to the weights of the corresponding NeRF, and inject symmetry priors (*i.e.*, symmetric features as well as pixel-aligned features) to predict colors and densities. Finally, we adopt the classic volume rendering technique [15,26] to synthesize novel views.

3.2 Encoding Holistic Representations

Humans usually understand the 3D shapes and appearances by first generating a profile, then restoring details from observations. Emulating the human visual system, we implement coarse depictions of objects by an image encoder network.

The image encoder network f is responsible for mapping the input image $\mathcal{I}_i$ into the latent code $\mathbf{z}_i$ which characterizes holistic information of the object's shape and appearance:

$$f : \mathbb{R}^{H\times W\times 3} \to \mathbb{R}^k, \quad \mathcal{I}_i \mapsto f(\mathcal{I}_i) = \mathbf{z}_i, \tag{1}$$

where k is the dimension of $\mathbf{z}_i$, and the parameters of f are denoted by Ω. Here, we denote the feature volume extracted by f during the encoding of holistic representations as $\mathbf{F}$ (*i.e.*, the concatenation of upsampled feature maps outputted by ResNet blocks). The image encoder network contains four ResNet blocks of ResNet-34 [12], followed by an average pooling layer and a fully connected layer.

3.3 Extracting Symmetric Features

The holistic representations introduced in the previous section coarsely describe objects. To synthesize detailed novel views, we follow PixelNeRF [44] and adopt pixel-aligned image features to compensate for the details. However, simply using pixel-aligned image features ignores the underlying 3D structure. In contrast, humans can infer the 3D shape and appearance from a single image, despite the information loss and self-occlusion that occurs during the imagery capture. This can boil down to the fact that humans usually resort to prior knowledge, *e.g.*, symmetry. Motivated by the above observation, we propose to inpaint invisible regions and alleviate the ill-posedness of novel view synthesis from a single image via symmetry priors. In the following, we briefly introduce the properties of 3D reflection symmetry [47], followed by how symmetry priors are applied.

3D reflection symmetry. When two points on an object's surface are symmetric, they share identical surface properties of the object. Formally, we define the symmetry regrading a rigid transformation $\mathbf{M} \in \mathbb{R}^{4\times4}$ as

$$\forall \mathbf{X} \in \mathbb{S} : \begin{cases} \mathbf{MX} \in \mathbb{S}, \\ \mathcal{F}(\mathbf{X}) = \mathcal{F}(\mathbf{MX}), \end{cases} \tag{2}$$

where $\mathbf{X}$ is the homogeneous coordinate of a point on the object's surface, $\mathbb{S} \subset \mathbb{R}^4$ is the set of points that are on the surface of an object, $\mathbf{MX}$ is the symmetric point of $\mathbf{X}$, and $\mathcal{F}(\cdot)$ stands for the surface properties at a given point.

The 2D projections $\mathbf{x} = [x, y, 1, 1/d]^T$ and $\mathbf{x}' = [x', y', 1, 1/d']^T$ of two 3D points $\mathbf{X}, \mathbf{X}' \in \mathbb{S}$ satisfy

$$\begin{cases} \mathbf{x} = \mathbf{K}\mathbf{R}_\mathbf{t}\mathbf{X}/d, \\ \mathbf{x}' = \mathbf{K}\mathbf{R}_\mathbf{t}\mathbf{X}'/d', \end{cases} \tag{3}$$

where $\mathbf{K} \in \mathbb{R}^{4\times4}$ and $\mathbf{R}_\mathbf{t} \in \mathbb{R}^{4\times4}$ are respectively the camera intrinsic matrix and extrinsic matrix. The latter transforms the coordinate from the world coordinate system to the camera coordinate system. d, d' are the depth in the camera space. When these two points are symmetric w.r.t. a rigid transformation, *i.e.*, $\mathbf{X}' = \mathbf{MX}$, the following constraint can be derived:

$$\mathbf{x}' = \frac{d}{d'}\mathbf{K}\mathbf{R}_\mathbf{t}\mathbf{M}{\mathbf{R}_\mathbf{t}}^{-1}\mathbf{K}^{-1}\mathbf{x}, \tag{4}$$

where $\mathbf{x}$ and $\mathbf{x}'$ are 2D projections of these two 3D points. This suggests that given a 2D projection $\mathbf{x}$, we can obtain its symmetric counterpart $\mathbf{x}'$ if $\mathbf{M}$ and

camera parameters are known. In this paper, we focus on exploring the benefits of explicitly embedding symmetry into our representation. To this end, we add the assumptions on the data distribution that objects are in a canonical coordinate frame, and that their symmetry axis is known.

Applying Symmetry Prior. To inpaint invisible regions, we apply the symmetry property introduced above and extract symmetric features $\mathbf{F}(\pi(\mathbf{X}'))$. This can be achieved by projecting the symmetric point $\mathbf{X}'$ to the 2D location $\mathbf{x}'$ on the image plane using camera parameters, followed by bilinearly interpolating between the pixelwise features on the feature volume $\mathbf{F}$ extracted by the image encoder network f. In addition, we follow PixelNeRF [44] and adopt pixel-aligned features $\mathbf{F}(\pi(\mathbf{X}))$, which share the same acquisition approach with symmetric features. They are subsequently concatenated together to form the local image features corresponding to $\mathbf{X}$.

3.4 Injecting Symmetry Prior into NeRF

In this section, we inject symmetry priors into the neural radiance field for single-view view synthesis. Technically, the weights of the neural radiance field are conditioned on the latent code $\mathbf{z}_i$ introduced in Sec. 3.2, which represents a coarse but holistic depiction of the object. To preserve fine-grained details, during the color and density prediction, we also take the pixel-aligned image features and the corresponding symmetric features as extra inputs to fulfill details observed in the input view.

Generating Neural Radiance Fields. We generate a specific neural radiance field by mapping a latent code $\mathbf{z}_i$ to the weights θ_i of the neural radiance field using the hypernetwork Ψ, which can be defined as follows:

$$\Psi : \mathbb{R}^k \to \mathbb{R}^l, \quad \mathbf{z}_i \mapsto \Psi(\mathbf{z}_i) = \theta_i, \tag{5}$$

where, l stands for the dimension of the parameter space of neural radiance fields. We parameterize Ψ as an MLP with parameters ψ. This can be interpreted as a simulation of the human visual system. Specifically, humans first estimate the holistic shape and appearance of the unseen object when given a single image, then formulate a sketch in their mind to represent the object. Similarly, SymmNeRF encodes overall information of the object as a latent code from a single image, followed by generating a corresponding neural radiance field to describe the object.

Color and Density Prediction. Given a reference image with known camera parameters, for a single query point location $\mathbf{X} \in \mathbb{R}^3$ on a ray $\mathbf{r} \in \mathbb{R}^3$ with unit-length viewing direction $\mathbf{d} \in \mathbb{R}^3$, SymmNeRF predicts the color and density at that point in 3D space, which is defined as:

$$\begin{aligned} &\Phi : \mathbb{R}^{m_{\mathbf{x}}} \times \mathbb{R}^{m_{\mathbf{d}}} \times \mathbb{R}^{2n} \to \mathbb{R}^3 \times \mathbb{R}, \\ &(\gamma_{\mathbf{x}}(\mathbf{X}), \gamma_{\mathbf{d}}(\mathbf{d}), \mathbf{F}(\pi(\mathbf{X})), \mathbf{F}(\pi(\mathbf{X}'))) \mapsto \\ &\Phi(\gamma_{\mathbf{x}}(\mathbf{X}), \gamma_{\mathbf{d}}(\mathbf{d}), \mathbf{F}(\pi(\mathbf{X})), \mathbf{F}(\pi(\mathbf{X}'))) = (\mathbf{c}, \sigma), \end{aligned} \tag{6}$$

where Φ represents a neural radiance field, an MLP network whose weights are given by the hypernetwork Ψ, $\mathbf{X}' \in \mathbb{R}^3$ is the corresponding symmetric 3D point of $\mathbf{X}$, $\gamma_{\mathbf{X}}(\cdot)$ and $\gamma_{\mathbf{d}}(\cdot)$ are positional encoding functions for spatial locations and viewing directions, n, $m_{\mathbf{X}}$ and $m_{\mathbf{d}}$ are respectively the dimensions of pixel-aligned features (symmetric features), $\gamma_{\mathbf{X}}(\mathbf{X})$ and $\gamma_{\mathbf{d}}(\mathbf{d})$, π denotes the process of projecting the 3D point onto the image plane using known intrinsics, and $\mathbf{F}$ is the feature volume extracted by the image encoder network f.

3.5 Volume Rendering

To render the color of a ray $\mathbf{r}$ passing through the scene, we first compute its camera ray $\mathbf{r}$ using the camera parameters and sample K points $\{\mathbf{X}_k\}_{k=1}^{K}$ along the camera ray $\mathbf{r}$ between near and far bounds, and then perform classical volume rendering [15,26]:

$$\tilde{C}(\mathbf{r}) = \sum_{k=1}^{K} T_k (1 - \exp(-\sigma_k \delta_k)) \mathbf{c}_k, \tag{7}$$

$$\text{where} \quad T_k = \exp(-\sum_{j=1}^{k-1} \sigma_j \delta_j), \tag{8}$$

where $\mathbf{c}_k$ and σ_k denote the color and density of the k-th sample on the ray, respectively, and $\delta_k = \|\mathbf{X}_{k+1} - \mathbf{X}_k\|_2$ is the interval between adjacent samples.

3.6 Training

To summarize, given a set of M instances training datasets $\mathcal{D} = \{\mathcal{C}_j\}_{j=1}^{M}$, where $\mathcal{C}_j = \{(\mathcal{I}_i^j, \mathbf{E}_i^j, \mathbf{K}_i^j)\}_{i=1}^{N}$ is a dataset of an instance object, we optimize SymmNeRF to minimize the rendering error of observed images:

$$\min_{\Omega,\psi} \sum_{j=1}^{M} \sum_{i=1}^{N} \mathcal{L}(\mathcal{I}_i^j, \mathbf{E}_i^j, \mathbf{K}_i^j; \Omega, \psi), \tag{9}$$

$$\mathcal{L} = \sum_{\mathbf{r} \in \mathcal{R}} \left\| \tilde{C}(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2, \tag{10}$$

where Ω and ψ are respectively the parameters of the image encoder network f and the hypernetwork Ψ, $\mathcal{R}$ is the set of camera rays passing through image pixels, and $C(\mathbf{r})$ denotes the ground truth pixel color.

4 Experiments

4.1 Datasets

Synthetic Renderings. We evaluate our approach on the synthetic ShapeNet benchmark [2] for single-shot reconstruction. 1) We mainly focus on the ShapeNet-SRN dataset, following the same protocol adopted in [33]. This dataset includes

Table 1: Quantitative comparisons against state-of-the-art methods on "Cars" and "Chairs" classes of the ShapeNet-SRN dataset. The best performance is in **bold**, and the second best is underlined.

Methods	Chairs		Cars		Average	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
GRF [37] (ICCV'21)	21.25	0.86	20.33	0.82	20.79	0.84
TCO [35] (ECCV'16)	21.27	0.88	-	-	-	-
dGQN [6] (Science'18)	21.59	0.87	-	-	-	-
ENR [5] (ICML'20)	22.83	-	22.26	-	22.55	-
SRN [33] (NeurIPS'19)	22.89	0.89	22.25	0.89	22.57	0.89
PixelNeRF [44] (CVPR'21)	23.72	0.91	23.17	0.90	23.45	0.91
ShaRF [30] (ICML'21)	23.37	**0.92**	22.53	0.90	22.90	0.91
CodeNeRF [13] (ICCV'21)	23.66	0.90	**23.80**	**0.91**	23.74	0.91
Ours	**24.32**	**0.92**	23.44	**0.91**	**23.88**	**0.92**

two object categories: 3,514 "Cars" and 6,591 "Chairs". The train/test split is predefined across object instances. There are 50 views per object instance in the training set. For testing, 251 novel views in an Archimedean spiral are used for each object instance in the test set. All images are at 128×128 pixels; 2) Similar to PixelNeRF [44], we also test our method on the ShapeNet-NMR dataset [17] under two settings: category-agnostic single-view reconstruction and generalization to unseen categories, following [17,23,27]. This dataset contains the 13 largest categories of ShapeNet and provides 24 fixed elevation views for each object instance. All images are of 64×64 resolution.

Real-World Renderings. We also generalize our model, trained only on the ShapeNet-SRN dataset, directly to two complex real-world datasets. One is the Pix3D [34] dataset containing various image-shape pairs with 2D-3D alignment. The other is the Stanford Cars [19] dataset which contains various real images of 196 classes of cars. All images of the two datasets are cropped and resized to 128×128 pixels during testing.

4.2 Implementation Details

SymmNeRF is trained using the AdamW optimizer [18,24]. The learning rate follows the warmup [12] strategy: linearly growing from 0 to 1×10^{-4} during the first 2k iterations and then decaying exponentially close to 0 over the optimization. The network parameters are updated with around 400-500k iterations. We use a batch size of 4 objects and a ray batch size of 256, each queried at 64 samples. Experiments are conducted on 2 NVIDIA GeForce RTX 3090 GPUs.

4.3 Comparisions

Here we compare SymmNeRF against the existing state-of-the-art methods, among which CodeNeRF, SRN and ShaRF require test-time optimization at inference, and ShaRF entails 3D ground truth voxel grids besides 2D supervision. To evaluate the quality of renderings, we adopt two standard image quality metrics: the Peak Signal-to-Noise Ratio (PSNR) and the Structure Similarity

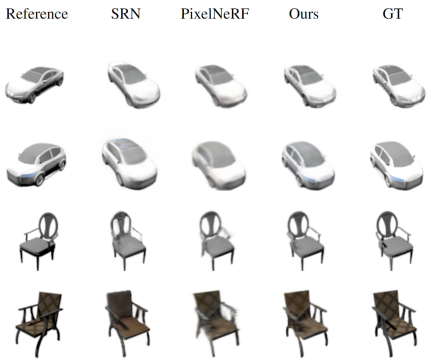

Fig. 3: Qualitative comparisons on "Cars" and "Chairs" classes. SymmNeRF can produce high-quality renderings with fine-grained details, proper geometry and reasonable appearance.

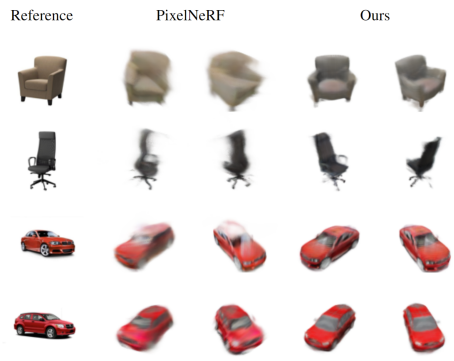

Fig. 4: Qualitative comparisons with PixelNeRF [44] on real-world Pix3D [34] and Stanford Cars [19] datasets. Compared with PixelNeRF, SymmNeRF yields better generalization.

Index Measure (SSIM) [40]. We also include LPIPS [45] in all experiments except the ShapeNet-SRN dataset. The better approach favors the higher PSNR and SSIM, and the lower LPIPS. Please refer to the supplementary material for more visualization.

Evaluations on the ShapeNet-SRN Dataset. In general, as shown in Table 1, our method outperforms or at least is on par with state-of-the-art methods. For the "Cars" category, SymmNeRF outperforms its competitors including PixelNeRF, SRN and ShaRF, and achieves comparable performance with CodeNeRF. Note that *our SymmNeRF solves a much harder problem than SRN and CodeNeRF*. In particular, SymmNeRF directly infers the unseen object representation in a single forward pass, while SRN and CodeNeRF need to be retrained on all new objects to optimize the latent codes. In addition, we observe that most cars from the "Cars" category share similar 3D shapes and simple textures. As a result, the experiment on the "Cars" category is in favor of CodeNeRF. In contrast, for the "Chairs" category, SymmNeRF significantly outperforms all baselines across all metrics by a large margin. This result implies that our model can generalize well on new objects, as the shapes and textures of chairs in the "Chairs" category vary considerably. This implies that SymmNeRF indeed captures the underlying 3D structure of objects with the help of symmetry priors and the hypernetwork, rather than simply exploiting data biases.

Here we compare SymmNeRF qualitatively with SRN and PixelNeRF in Fig. 3. One can observe that: i) SRN is prone to generate overly smooth renderings and is unable to capture the accurate geometry, leading to some distortions; ii) PixelNeRF performs well when the query view is close to the reference one, but fails to recover the details invisible in the reference, especially when the rendered view is far from the reference; iii) SymmNeRF, by contrast, can synthesize

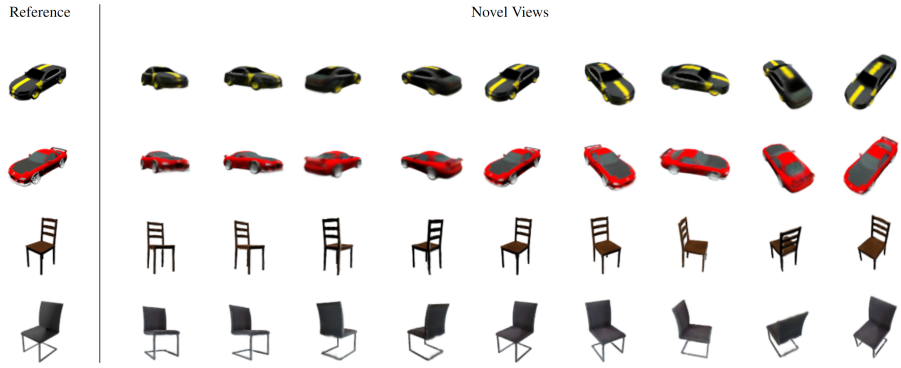

Fig. 5: Novel view synthesis on "Cars" and "Chairs" of ShapeNet-SRN dataset.

photo-realistic, reasonable novel views with fine-grained details close to ground truths.

We further demonstrate the high-quality results of SymmNeRF by providing more novel view synthesis visualization in Fig. 5. As can be seen, SymmNeRF can always synthesize photo-realistic and reasonable novel renderings from totally different viewpoints.

Generalization on Real-World Datasets. To further investigate the generalization of SymmNeRF, we evaluate SymmNeRF on two real-world datasets, *i.e.*, the Pix3D [34] and the Stanford Cars [19]. Note that for the lack of ground truth, we only show the qualitative results on the two datasets. Here we apply SymmNeRF trained on the synthetic chairs and cars directly on the real-world images without any finetuning. As shown in Fig. 4, PixelNeRF [44] fails to synthesize reasonable novel views, because it only notices the use of pixel-aligned image features, ignoring the underlying 3D structure the reference view provides. Compared with PixelNeRF [44], SymmNeRF can effectively infer the geometry and appearance of real-world chairs and cars. Please also note that there are no camera poses for real-world objects from Pix3D and Stanford Cars. Our model assumes that objects are at the center of the canonical space and once trained, can estimate camera poses for each reference view similar to CodeNeRF [13].

Evaluations on the ShapeNet-NMR Dataset. Although the experimental results of two common categories have demonstrated that including symmetry is simple yet effective, we further explore our approach on the ShapeNet-NMR dataset under two settings: category-agnostic single-view reconstruction and generalization to unseen categories. 1) Category-agnostic single-view reconstruction: only a single model is trained across the 13 largest categories of ShapeNet. We show in Table 2 and Fig. 6 that SymmNeRF outperforms other state-of-the-art methods [23,27,33,44]. This also implies that *symmetry priors benefit the reconstruction of almost all symmetric objects*; 2) Generalization to unseen categories: we reconstruct ShapeNet categories which are not involved in training.

Table 2: Quantitative comparisons against state-of-the-art methods on the 13 largest categories of the ShapeNet-NMR dataset.

	Methods	plane	bench	cbnt.	car	chair	disp.	lamp	spkr.	rifle	sofa	table	phone	boat	mean
PSNR↑	DVR	25.29	22.64	24.47	23.95	19.91	20.86	23.27	20.78	23.44	22.35	21.53	24.18	25.09	22.70
	SRN	26.62	22.20	23.42	24.40	21.85	19.07	22.17	21.04	24.95	23.65	22.45	20.87	25.86	23.28
	PixelNeRF	29.76	26.35	27.72	27.58	23.84	24.22	28.58	24.44	30.60	26.94	25.59	27.13	29.18	26.80
	Ours	**30.57**	**27.44**	**29.34**	**27.87**	**24.29**	**24.90**	**28.98**	**25.14**	**30.64**	**27.70**	**27.16**	**28.27**	**29.71**	**27.57**
SSIM↑	DVR	0.905	0.866	0.877	0.909	0.787	0.814	0.849	0.798	0.916	0.868	0.840	0.892	0.902	0.860
	SRN	0.901	0.837	0.831	0.897	0.814	0.744	0.801	0.779	0.913	0.851	0.828	0.811	0.898	0.849
	PixelNeRF	0.947	0.911	0.910	0.942	0.858	0.867	0.913	0.855	0.968	0.908	0.898	0.922	0.939	0.910
	Ours	**0.955**	**0.925**	**0.922**	**0.945**	**0.865**	**0.875**	**0.917**	**0.862**	**0.970**	**0.915**	**0.917**	**0.929**	**0.943**	**0.919**
LPIPS↓	DVR	0.095	0.129	0.125	0.098	0.173	0.150	0.172	0.170	0.094	0.119	0.139	0.110	0.116	0.130
	SRN	0.111	0.150	0.147	0.115	0.152	0.197	0.210	0.178	0.111	0.129	0.135	0.165	0.134	0.139
	PixelNeRF	0.084	0.116	0.105	0.095	0.146	0.129	0.114	0.141	0.066	0.116	0.098	0.097	0.111	0.108
	Ours	**0.062**	**0.085**	**0.068**	**0.082**	**0.120**	**0.104**	**0.096**	**0.108**	**0.054**	**0.086**	**0.067**	**0.068**	**0.089**	**0.084**

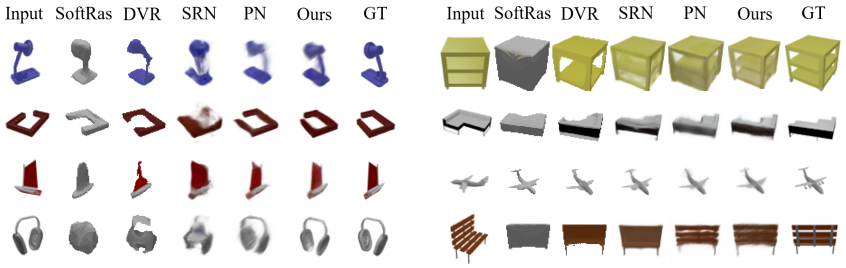

Fig. 6: Qualitative comparisons on the ShapeNet-NMR dataset under the category-agnostic single-view reconstruction setting.

The results in Table 3 and Fig. 7 suggest that our method performs comparably to PixelNeRF. This means that *our method can also handle out-of-distribution categories with the help of symmetry priors.*

Asymmetric Objects. As shown in Fig. 6 (Row 2), *our method can also deal with objects that are not perfectly symmetric.* This is because a few asymmetric objects are also included in the training dataset. Our model can perceive and recognize asymmetry thanks to the global latent code and hypernetwork. SymmNeRF therefore adaptively chooses to utilize local features to reconstruct asymmetric objects.

4.4 Ablation Study

To validate the design choice of SymmNeRF, we conduct ablation studies on the synthetic "Chairs" and "Cars" from the ShapeNet-SRN dataset. Table 4 shows the results corresponding to the effectiveness of the pixel-aligned, symmetric features and the hypernetwork. One can observe: i) *The symmetry priors injection benefits novel view synthesis.* Compared with (b) in average performance, our full model (c) with the symmetric priors injection yields a relative improvement of 6.0% PSNR and 2.3% SSIM. This finding highlights the importance of the

Table 3: Quantitative comparisons against state-of-the-art methods on 10 unseen categories of ShapeNet-NMR dataset. The models are trained on only planes, cars and chairs.

	Methods	bench	cbnt.	disp.	lamp	spkr.	rifle	sofa	table	phone	boat	mean
PSNR↑	DVR	18.37	17.19	14.33	18.48	16.09	20.28	18.62	16.20	16.84	22.43	17.72
	SRN	18.71	17.04	15.06	19.26	17.06	23.12	18.76	17.35	15.66	24.97	18.71
	PixelNeRF	23.79	**22.85**	**18.09**	**22.76**	**21.22**	23.68	24.62	**21.65**	**21.05**	26.55	**22.71**
	Ours	**23.87**	21.36	16.83	22.68	19.98	**23.77**	**25.10**	21.10	20.48	**26.80**	22.36
SSIM↑	DVR	0.754	0.686	0.601	0.749	0.657	0.858	0.755	0.644	0.731	0.857	0.716
	SRN	0.702	0.626	0.577	0.685	0.633	0.875	0.702	0.617	0.635	0.875	0.684
	PixelNeRF	0.863	**0.814**	**0.687**	0.818	**0.778**	0.899	0.866	**0.798**	0.801	0.896	**0.825**
	Ours	**0.873**	0.780	0.663	**0.824**	0.751	**0.902**	**0.881**	0.792	**0.802**	**0.909**	0.823
LPIPS↓	DVR	0.219	0.257	0.306	0.259	0.266	0.158	0.196	0.280	0.245	0.152	0.240
	SRN	0.282	0.314	0.333	0.321	0.289	0.175	0.248	0.315	0.324	0.163	0.280
	PixelNeRF	0.164	0.186	0.271	0.208	0.203	0.141	0.157	0.188	0.207	0.148	0.182
	Ours	**0.126**	**0.174**	**0.251**	**0.184**	**0.185**	**0.121**	**0.115**	**0.163**	**0.178**	**0.111**	**0.155**

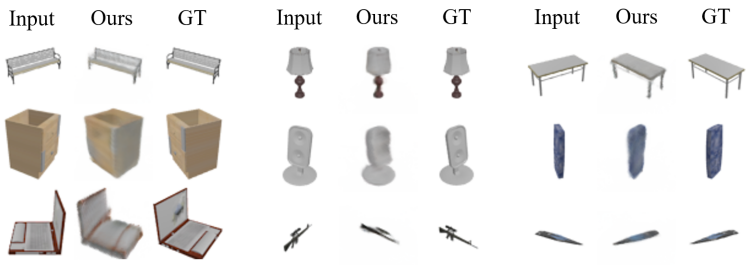

Fig. 7: Qualitative visualization on the ShapeNet-NMR dataset under the generalization to unseen categories setting.

symmetry priors on novel view synthesis when only a single image is provided; ii) *The hypernetwork matters.* Compared with our full model (c), the rendering quality of (d) deteriorates if we do not adopt the hypernetwork. This may lie in the fact that simply conditioning on local features ignores the underlying 3D structure of objects. In contrast, combining local and global conditioning via the hypernetwork module not only enables recovery of rendering details, but also improves generalization to unseen objects in a coarse-to-fine manner. We also visualize the comparative results in Fig. 8. The baseline model (a) tends to render smoothly. Simply using pixel-aligned image features (b) still fails to fully understand 3D structure. In contrast, our full model (c) reproduces photo-realistic details from most viewpoints. The rendering quality of (d) deteriorates as the hypernetwork is not adopted. We have to emphasize that, *only including both the symmetry priors and the hypernetwork can accurately recovers the geometry information and texture details despite the occlusions.*

Table 4: Ablation study on each component of SymmNeRF.

		Image encoder network	Hypernetwork	Local features	Symm features	PSNR(Δ)↑	SSIM(Δ)↑
Chairs	(a)	✓	✓			21.26 (-)	0.87 (-)
	(b)	✓	✓	✓		23.09 (8.6%)	0.91 (4.6%)
	(c)	✓	✓	✓	✓	**24.32 (14.4%)**	**0.92 (5.7%)**
	(d)	✓		✓	✓	19.76 (-7.1%)	0.85 (-2.3%)
Cars	(a)	✓	✓			20.65 (-)	0.87 (-)
	(b)	✓	✓	✓		22.15 (7.3%)	0.89 (2.3%)
	(c)	✓	✓	✓	✓	**23.44 (13.5%)**	**0.91 (4.6%)**
	(d)	✓		✓	✓	21.15 (2.4%)	0.86 (-1.2%)
Average	(a)	✓	✓			20.96 (-)	0.87 (-)
	(b)	✓	✓	✓		22.62 (7.9%)	0.90 (3.4%)
	(c)	✓	✓	✓	✓	**23.88 (13.9%)**	**0.92 (5.7%)**
	(d)	✓		✓	✓	20.46 (-2.4%)	0.86 (-1.1%)

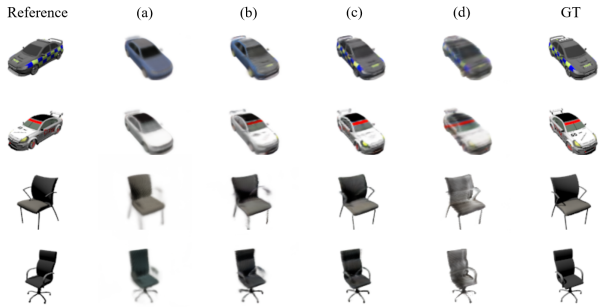

Fig. 8: Qualitative evaluation of different configurations on ShapeNet-SRN.

5 Conclusion

Existing methods [13,44] fail to synthesize fine appearance details of objects, especially when the target view is far away from the reference view. They focus on learning scene priors, but ignore fully exploring the attributes of objects, *e.g.*, symmetry. In this paper, we investigate the potential performance gains of explicitly injecting symmetry priors into the scene representation. In particular, we combine hypernetworks [33] with local conditioning [31,37,44], embedded with the symmetry prior. Experimental results demonstrate that such a symmetry prior can boost our model to synthesize novel views with more details regardless of the pose transformation, and show good generalization when applied to unseen objects.

Acknowledgement. This work is supported in part by the National Natural Science Foundation of China (Grant No. U1913602). This study is also supported under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds. In: International conference on machine learning. pp. 40–49. PMLR (2018) 1, 3
2. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015) 8
3. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5939–5948 (2019) 3
4. 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. pp. 11–20 (1996) 1, 3
5. Dupont, E., Martin, M.B., Colburn, A., Sankar, A., Susskind, J., Shan, Q.: Equivariant neural rendering. In: International Conference on Machine Learning. pp. 2761–2770. PMLR (2020) 9
6. Eslami, S.A., Rezende, D.J., Besse, F., Viola, F., Morcos, A.S., Garnelo, M., Ruderman, A., Rusu, A.A., Danihelka, I., Gregor, K., et al.: Neural scene representation and rendering. Science **360**(6394), 1204–1210 (2018) 9
7. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 605–613 (2017) 1, 3
8. Flynn, J., Broxton, M., Debevec, P., DuVall, M., Fyffe, G., Overbeck, R., Snavely, N., Tucker, R.: Deepview: View synthesis with learned gradient descent. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2367–2376 (2019) 3
9. 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. pp. 43–54 (1996) 1, 3
10. Ha, D., Dai, A., Le, Q.V.: Hypernetworks. arXiv preprint arXiv:1609.09106 (2016) 4
11. Häni, N., Engin, S., Chao, J.J., Isler, V.: Continuous object representation networks: novel view synthesis without target view supervision. arXiv preprint arXiv:2007.15627 (2020) 3
12. 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. pp. 770–778 (2016) 6, 9
13. Jang, W., Agapito, L.: Codenerf: Disentangled neural radiance fields for object categories. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12949–12958 (2021) 2, 4, 9, 11, 14
14. Jimenez Rezende, D., Eslami, S., Mohamed, S., Battaglia, P., Jaderberg, M., Heess, N.: Unsupervised learning of 3d structure from images. Advances in neural information processing systems **29**, 4996–5004 (2016) 1, 3
15. Kajiya, J.T., Von Herzen, B.P.: Ray tracing volume densities. ACM SIGGRAPH computer graphics **18**(3), 165–174 (1984) 5, 8
16. Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 371–386 (2018) 1, 3

17. Kato, H., Ushiku, Y., Harada, T.: Neural 3d mesh renderer. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 9
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 9
19. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: Proceedings of the IEEE international conference on computer vision workshops. pp. 554–561 (2013) 9, 10, 11
20. Levoy, M., Hanrahan, P.: Light field rendering. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. pp. 31–42 (1996) 1, 3
21. 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. pp. 2916–2925 (2018) 1, 3
22. Littwin, G., Wolf, L.: Deep meta functionals for shape representation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1824–1833 (2019) 4
23. Liu, S., Li, T., Chen, W., Li, H.: Soft rasterizer: A differentiable renderer for image-based 3d reasoning. The IEEE International Conference on Computer Vision (ICCV) (Oct 2019) 9, 11
24. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2018) 9
25. 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. pp. 4460–4470 (2019) 2, 3
26. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European conference on computer vision. pp. 405–421. Springer (2020) 2, 3, 5, 8
27. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3504–3515 (2020) 2, 3, 9, 11
28. 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. pp. 165–174 (2019) 2, 3
29. Ranjan, A., Bolkart, T., Sanyal, S., Black, M.J.: Generating 3d faces using convolutional mesh autoencoders. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 704–720 (2018) 1, 3
30. Rematas, K., Martin-Brualla, R., Ferrari, V.: Sharf: Shape-conditioned radiance fields from a single view. In: ICML (2021) 4, 9
31. 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. pp. 2304–2314 (2019) 2, 14
32. Sitzmann, V., Thies, J., Heide, F., Nießner, M., Wetzstein, G., Zollhofer, M.: Deepvoxels: Learning persistent 3d feature embeddings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2437–2446 (2019) 1, 3

33. Sitzmann, V., Zollhoefer, M., Wetzstein, G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. Advances in Neural Information Processing Systems **32**, 1121–1132 (2019) 2, 3, 4, 8, 9, 11, 14
34. Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J.B., Freeman, W.T.: Pix3d: Dataset and methods for single-image 3d shape modeling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2974–2983 (2018) 9, 10, 11
35. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Multi-view 3d models from single images with a convolutional network. In: European Conference on Computer Vision. pp. 322–337. Springer (2016) 9
36. Tewari, A., Fried, O., Thies, J., Sitzmann, V., Lombardi, S., Sunkavalli, K., Martin-Brualla, R., Simon, T., Saragih, J., Nießner, M., et al.: State of the art on neural rendering. In: Computer Graphics Forum. vol. 39, pp. 701–727. Wiley Online Library (2020) 1, 3
37. 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 (ICCV). pp. 15182–15192 (October 2021) 2, 3, 9, 14
38. Tucker, R., Snavely, N.: Single-view view synthesis with multiplane images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 551–560 (2020) 3
39. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 52–67 (2018) 1, 3
40. 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), 600–612 (2004) 10
41. Wu, S., Rupprecht, C., Vedaldi, A.: Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1–10 (2020) 4
42. Xie, H., Yao, H., Sun, X., Zhou, S., Zhang, S.: Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2690–2698 (2019) 1, 3
43. Xu, Y., Fan, T., Yuan, Y., Singh, G.: Ladybird: Quasi-monte carlo sampling for deep implicit field based 3d reconstruction with symmetry. In: European Conference on Computer Vision. pp. 248–263. Springer (2020) 4
44. 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. pp. 4578–4587 (2021) 2, 3, 6, 7, 9, 10, 11, 14
45. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018) 10
46. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: learning view synthesis using multiplane images. ACM Transactions on Graphics (TOG) **37**(4), 1–12 (2018) 3
47. Zhou, Y., Liu, S., Ma, Y.: Nerd: Neural 3d reflection symmetry detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15940–15949 (2021) 4, 6