MS-NeRF: Multi-Space Neural Radiance Fields

Ze-Xin Yin, Peng-Yi Jiao, Jiaxiong Qiu, Ming-Ming Cheng, and Bo Ren

Abstract—Existing Neural Radiance Fields (NeRF) methods suffer from the existence of reflective objects, often resulting in blurry or distorted rendering. Instead of calculating a single radiance field, we propose a multi-space neural radiance field (MS-NeRF) that represents the scene using a group of feature fields in parallel sub-spaces, which leads to a better understanding of the neural network toward the existence of reflective and refractive objects. Our multi-space scheme works as an enhancement to existing NeRF methods, with only small computational overheads needed for training and inferring the extra-space outputs. We design different multi-space modules for representative MLP-based and grid-based NeRF methods, which improve Mip-NeRF 360 by 4.15 dB in PSNR with 0.5% extra parameters and further improve TensoRF by 2.71 dB with 0.046% extra parameters on reflective regions without degrading the rendering quality on other regions. We further construct a novel dataset consisting of 33 synthetic scenes and 7 real captured scenes with complex reflection and refraction, where we design complex camera paths to fully benchmark the robustness of NeRF-based methods. Extensive experiments show that our approach significantly outperforms the existing single-space NeRF methods for rendering high-quality scenes concerned with complex light paths through mirror-like objects. The source code, dataset, and results are available via our project page: https://zx-yin.github.io/msnerf/.

Index Terms—Neural Radiance Fields, Multi-Space NeRF, dataset.

1 Introduction

Neural Radiance Fields (NeRF) [2] and its variants are refreshing the community of neural rendering and 3D reconstruction, and the potential for more promising applications is still under exploration. NeRF represents scenes as continuous radiance fields stored by simple Multi-layer Perceptrons (MLPs) and renders novel views by integrating the densities and radiance, which are queried from the MLPs by points sampled along the ray from the camera to the image plane. Since its first presentation [2], many efforts have been investigated to enhance the method, such as extending to unbounded scenes [3], [4], handling moving objects [5]–[7], or reconstructing from pictures in the wild [8]–[11].

However, rendering scenes with mirrors is still a challenging task for state-of-the-art NeRF-like methods. One of the principle assumptions for the NeRF method is the multi-view consistency property of the target scenes [12]–[15]. When there are mirrors in the space, if one allows the viewpoints to move 360-degree around the scene, there is no consistency between the front and back views of a mirror, since the mirror surface and its reflected virtual image are only visible from a small range of views. As a result, it is often required to manually label the reflective surfaces in order to avoid falling into sub-optimal convergences [16].

In this paper, we propose a novel multi-space NeRF method to allow the automatic handling of mirror-like objects in the 360-degree high-fidelity rendering of scenes without any manual labeling. Instead of regarding the Euclidean scene space as a single space, we treat it as composed of multiple virtual sub-spaces, whose composition changes according to location and view direction. We show that our approach using multi-space decomposition leads to successful handlements of complex reflections and refractions where the multi-view consistency is heavily violated in the Euclidean real space. Furthermore, we show that the above benefits can be achieved by designing a low-cost multi-space module and replacing the original output layer with it. Therefore, our multi-space approach serves as a general enhancement to the NeRF-based backbone, equipping most NeRF-like methods with the ability to model complex reflection and refraction, as shown in Fig. 1.

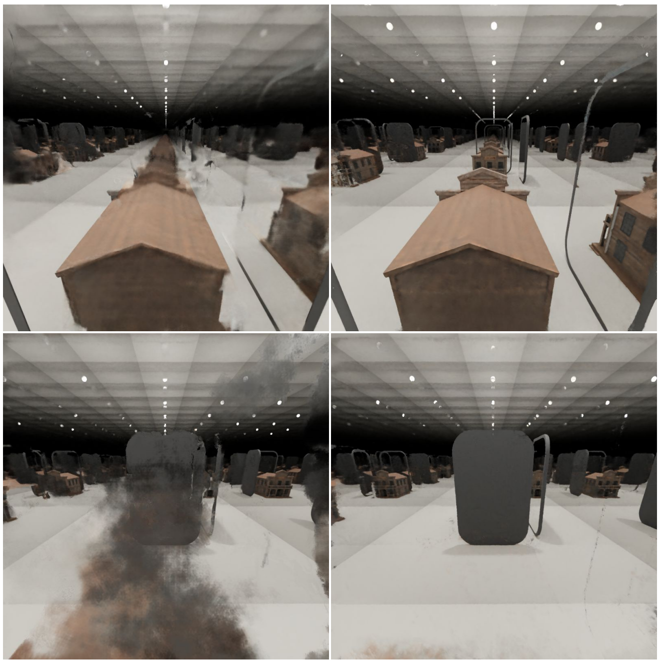

(a) Mip-NeRF 360 (b) MS-Mip-NeRF 360

Fig. 1: These are test views from the novel mirror-passing-through path. The first row is in front of the mirror, while the last row is behind the mirror.

- *All authors are with the TBI Center & VCIP, Nankai University.*
- *Bo Ren is the corresponding author.*
- *A preliminary version of this work appeared at CVPR [1].*

Existing datasets have not paid enough attention to the 360-degree rendering of scenes containing mirror-like objects, such as RFFR [16] just has forward-facing scenes, and the Shiny dataset in [17] with small viewpoints changes and cannot exhibit view-dependent effects in large angle scale. Therefore, we construct a novel dataset dedicated to evaluation for the 360-degree high-fidelity rendering of scenes containing complex reflections and refractions. In this dataset, we collect 33 synthesized scenes and 7 captured real-world scenes. Each synthesized scene consists of 120 images captured in the 360-degree circle path around reflective or refractive objects, with 100 randomly split for training, 10 for validation, and 10 for evaluation. Furthermore, we design more challenging paths for 10 of the synthesized scenes to benchmark the robustness of NeRF-based methods, including 360-degree spiral paths where cameras gradually spiral up from the equator to the pole and novel mirror-passing-through paths where cameras move through the mirrors back and force, each of which have 100 training views and 200 testing views following the convention of NeRF dataset [2]. Each real-world scene is captured randomly around scenes with reflective and refractive objects, consisting of 62 to 118 images, and organized under the convention of LLFF [18].

We then demonstrate the superiority of our approach by comparisons, using three representative baseline models, *i.e.*, NeRF [2], Mip-NeRF [19], and Mip-NeRF 360 [4], with and without our multi-space module. Besides, we investigate the grid-based acceleration methods, and we propose a hybrid multi-space module based on classic methods, *i.e.*, TensoRF [20] and iNGP [21], to demonstrate the compatibility of our scheme. 3D reconstruction methods have a stronger dependence on multi-view consistency; therefore, we experimentally integrate our multi-space module with a classic NeRF-based reconstruction method, *i.e.*, NeuS [22], and the rendering results indicate that our module is also beneficial to reconstruction methods. Experiments show that our approach not only improves performance by a large margin on scenes with reflection and refraction but also exhibits robustness on methods not specialized in rendering. Our main contributions are as follows:

- We propose a multi-space NeRF method that automatically handles mirror-like objects in 360-degree high-fidelity scene rendering, achieving significant improvements over the existing representative baselines both quantitatively and qualitatively.
- We design a lightweight module that can equip most NeRF-like methods with the ability to model reflection and refraction.
- We propose a hybrid multi-space scheme for TensoRF and iNGP, exhibiting the compatibility of our scheme with grid-based NeRF methods.
- We construct a dataset dedicated to evaluation for the 360-degree high-fidelity rendering of scenes containing complex reflections and refractions, including 33 synthesized scenes and 7 real captured scenes, with challenging camera paths.

2 Related Work

Coordinate-based novel view synthesis. NeRF [2] has bridged the gap between computer vision and computer graphics, and reveals a promising way to render high-quality photorealistic scenes with only posed images. The insights and the generalization ability of this scheme also facilitate various tasks, *i.e.*, 3D reconstruction [22]–[28], 3D-aware generation [29]–[33], 3D-aware edition [34]–[36], and avatar reconstruction and manipulation [37]–[40]. Researchers have made great efforts to enhance this scheme. Mip-NeRF [19] enhances the anti-aliasing ability of NeRF by featuring 3D conical frustum using integrated positional encoding. [41], [42] adapt this scheme to HDR images. [3], [4] extend NeRF and its variants to unbounded scenes. [22]–[24], [26], [27] construct the relationship between the SDF and the density in volumetric rendering of NeRF for 3D reconstruction. There are also many works trying to speed up the training and inference speed using explicit or hybrid representations [20], [21], [43]–[47].

Glossy materials with high specular have a great influence on NeRF-like methods. [48] is inspired by precomputation-based techniques [49] in computer graphics to represent and render view-dependent specular and reflection, but it fails to handle mirror-like reflective surfaces because the virtual images cannot be treated as textures. Guo *et al.* [16] propose to decompose reflective surfaces into a transmitted part and reflected part, which is the most relevant work to ours. However, such decomposition cannot handle 360-degree views with mirror-like objects, because the virtual images have no difference from real objects until the viewpoint moves beyond a certain angle. Zeng *et al.* [50] incorporates the ray tracing scheme into NeRF to model the reflection, but there also lacks views behind the mirrors.

Another line of work similar to ours is multiple neural radiance fields, but they do so for different purposes [30], [45], [51]–[53]. [30] uses object-level neural radiance fields for 3D-aware generation and composition. [45], [51] uses multiple small MLPs for efficient rendering. [52], [53] uses multiple object-level neural radiance fields for 3D scene decomposition and edition.

Commonly used datasets. Researchers have introduced or constructed many different datasets to facilitate the development of NeRF-based methods in various tasks. Mildenhall *et al.* [2] collect a dataset containing eight rendered sets of posed images about eight objects separately, and eight real captured forward-facing scenes with the camera poses and intrinsics estimated by COLMAP [54]. Nevertheless, these scenes lack reflection and refraction, which are very common. Wizadwongsa *et al.* [17] propose a dataset, namely Shiny, that contains eight more challenging scenes to test NeRF-like methods on view-dependent effects, but they are captured in a roughly forward-facing manner. Verbin *et al.* [48] create a dataset of six glossy objects, namely Shiny Blender, which are rendered under similar conditions as done in NeRF to test methods in modeling more complex materials. For unbounded scenes, Barron *et*

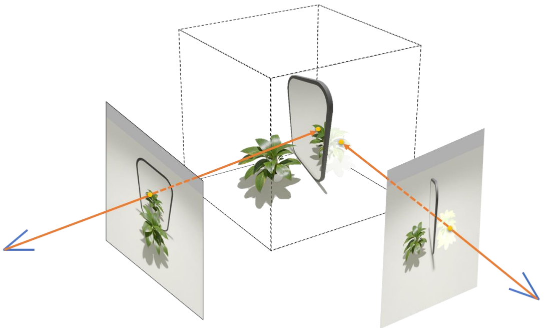
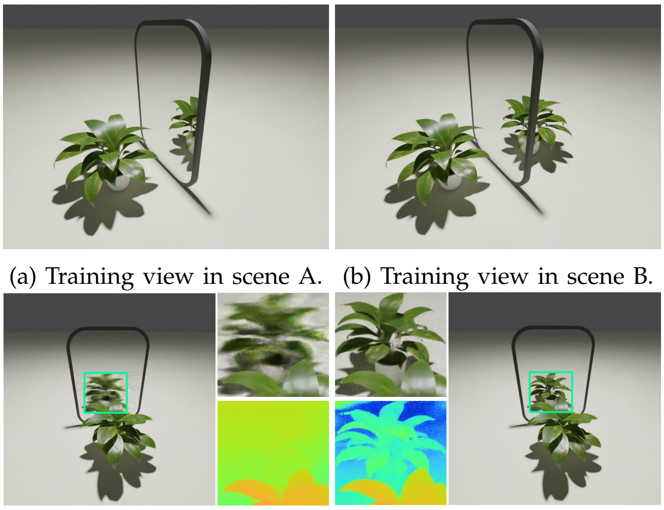


Fig. 2: The virtual image created by the mirror is visible only in a small range of views, which violates the multi-view consistency.



(a) Training view in scene A. (b) Training view in scene B.

(c) Render view in scene A. (d) Render view in scene B.

Fig. 3: The first row is training view examples in the two scenes. In scene A, there is only a plant in front of a mirror, while in scene B we carefully place another plant to match the exact position where the virtual image lies. The second row is test views with rendered depth from the vanilla NeRF trained on the toy scenes. As demonstrated, NeRF can avoid the trap of treating reflected images as textures when the 'virtual image' satisfies multi-view consistency.

al. [4] construct a dataset consisting of 5 outdoor scenes and 4 indoor scenes, while Zhang *et al.* [3] adopt Tanks and Temples (T&T) dataset [55] and the Light Field dataset [56]. Bemana1 *et al.* [57] capture a dataset consisting of refractive objects, which is composed of four scenes with cameras moving in a large range. Guo *et al.* [16] collect six forward-facing scenes with reflective and semi-transparent materials, which is, to date, the most relevant dataset to ours, but ours is much more challenging. DTU dataset [58] and BlendedMVS dataset [59] are commonly used as benchmarks for the evaluation of 3D reconstruction.

3 Method

3.1 Preliminaries: Neural Radiance Fields

Neural Radiance Fields (NeRF) [2] encodes a scene in the form of continuous volumetric fields into the weights of a multilayer perceptron (MLP), and adopts the absorption-only model in the traditional volumetric rendering to synthesize novel views. The training process only requires a sparse set of posed images and casts rays $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ through the scene, where $\mathbf{o} \in \mathbb{R}^3$ is the camera center and $\mathbf{d} \in \mathbb{R}^3$ is the view direction, and the rays can be calculated by intrinsics and poses from the training data. Given these rays, NeRF samples a set of 3D points $\{\mathbf{p}_i = \mathbf{o} + t_i\mathbf{d}\}$ by the distance to the camera t_i in the Euclidean space and projects these points to a higher dimensional space using the following function:

$$\gamma(\mathbf{p}) = [\sin(\mathbf{p}), \cos(\mathbf{p}), ..., \sin(2^{L-1}\mathbf{p}), \cos(2^{L-1}\mathbf{p})] \quad (1)$$

where L is a hyperparameter and $\mathbf{p}$ is a sampled point.

Given the projected features $\{\gamma(\mathbf{p}_{\mathrm{i}})\}$ and the ray direction $\mathbf{d}$, the MLP outputs the densities $\{\sigma_i\}$ and colors $\{\mathbf{c}_i\}$, which are used to estimate the color $\mathbf{C}(\mathbf{r})$ of the ray using the quadrature rule reviewed by Max [60]:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i \quad (2)$$

with $T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j\delta_j)$ and $\delta_i = t_i - t_{i-1}$. Since the equation is differentiable, the model parameters can be optimized directly by Mean Squared Error (MSE) loss:

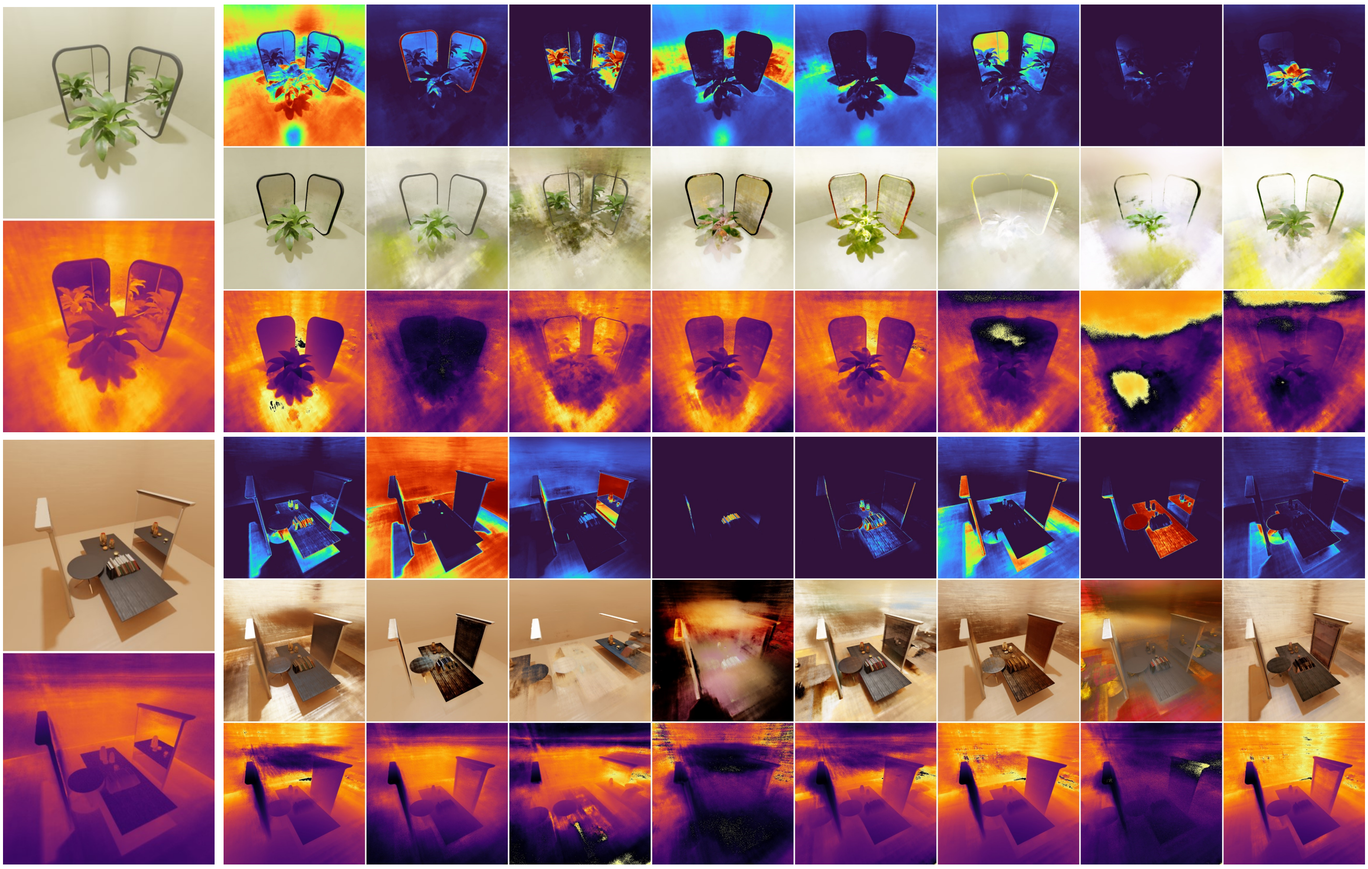
$$\mathcal{L} = \frac{1}{|\mathcal{R}|} \sum_{r\in\mathcal{R}} ||\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})||_2 \quad (3)$$

where $\mathcal{R}$ is a training batch of rays. Besides, NeRF also adopts a hierarchical sampling strategy to sample more points where higher weights are accumulated. With these designs, NeRF achieves state-of-the-art photorealistic results of novel view synthesis in most cases.

3.2 Multi-space Neural Radiance Field

The volumetric rendering equation and the continuous representation ability of MLPs do guarantee the success of NeRF-based methods in novel view synthesis, but as pointed out by previous works [12], [13], [16], there is also an unignorable property hidden in the training process that helps the convergence, which is the multi-view consistency. However, the multi-view consistency can be easily violated by any reflective surfaces. An example is shown in Fig. 2, when looking in front of a mirror one can observe the reflective virtual image as if there were an object behind it, but when looking from a side or backward, there is actually nothing behind the mirror. In practice, this means there will be completely conflictive training batches violating the fitting process of MLP.

To experimentally demonstrate the importance of multi-view consistency and its influence on the conventional NeRF network structure, We create two 360-degree toy scenes using an open source software Blender [61], each of which consists of 100 training images and 10 test images, training view examples are shown in Fig. 3a and Fig. 3b. The only difference between the two scenes is that we place a mirror-posed real object behind the mirror in the latter scene, but not



(a) Composed

(b) Weight maps, RGB images and depth maps of sub-spaces.

Fig. 4: We visualize composed RGB and depth maps of novel views and the decoded images with the corresponding weights and depth maps of all sub-spaces from our MS-NeRF$_B$ model in Sec. 6.4. The results show that our method successfully decomposes virtual images into certain sub-spaces.

in the former one. We train the vanilla NeRF separately on these toy scenes under the same setting and render some views from the test set as in Fig. 3c and Fig. 3d, which clearly shows that the vanished virtual image (*i.e.*, violation to the multi-view consistency) in some views leads the model to suboptimal results in reflection-related regions and produces blur in rendering. Interestingly, the conventional NeRF is still trying to fulfill the multi-view consistency assumption in the process. From the depth map in Fig. 3c, we can easily conclude that the conventional NeRF treats the viewed virtual image as a "texture" on the reflective surface, achieving a compromise between its principle assumption and the conflicts in training data, although the compromise leads to false understandings and worse rendering results of the real scenes.

Contrary to the conventional NeRF, inspired by the common perspective in Physics and Computer Graphics that reflective light can be viewed as "directly emitted" from its mirror-symmetric direction, from a possible "virtual source inside the virtual space in the mirror," we build our novel multi-space NeRF approach on the following assumption:

Assumption 1: At the existence of reflection and refraction, the real Euclidean scene space can be decomposed into multiple virtual sub-spaces. Each sub-space satisfies the multi-view consistency property.

	PSNR↑	SSIM↑	LPIPS↓
Mip-NeRF 360 R	31.35	0.948	**0.031**
Mip-NeRF 360 F	**31.40**	**0.948**	**0.031**
iNGP R	**33.43**	**0.964**	**0.015**
iNGP F	33.25	0.962	0.017

Table 1: Results on the Realistic Synthetic 360° dataset, 'R' indicates neural radiance fields, and 'F' indicates neural feature fields.

It follows that the composition weights of the sub-spaces can change according to the spatial location and the view direction. Thus all sub-space contributes dynamically to the final render result. In this way, the violation of the multi-view consistency in real Euclidean space when there is a reflective surface can be overcome by placing the virtual images in certain sub-spaces only visible from certain views, as shown in Fig. 4. The depth map shown in Fig. 5 further confirms the insight that our multi-space module equips the conventional NeRF with the ability of understanding the possible "virtual source inside the virtual space in the mirror", on the contrary, ordinary NeRF even fails to model complex reflections as textures.

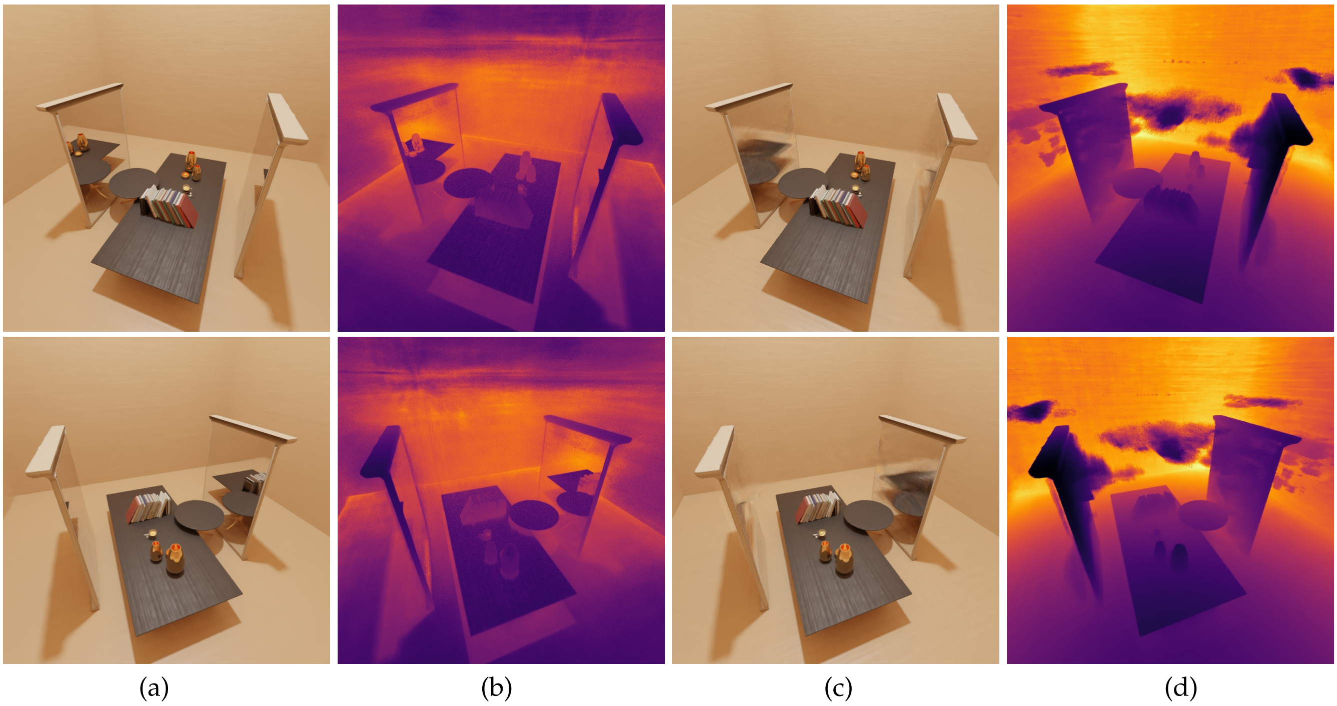


Fig. 5: (a) render result by MS-NeRF$_B$. (b) visualization of depth maps rendered by MS-NeRF$_B$. (c) render result by NeRF. (d) visualization of depth maps rendered by NeRF. The visualization from Sec. 6.4 indicates that our MS module understands the light transport at the occurrence of reflections, and the common parts can also be rendered correctly.

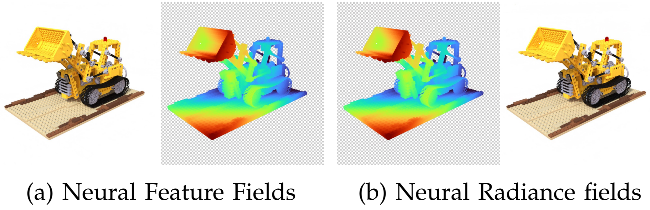


Fig. 6: The rendered depth and RGB map from neural radiance fields and neural feature fields based on Mip-NeRF 360.

4 Multi-Space module

In this section, we revise the neural feature fields in Sec. 4.1, , then we introduce our MLP-based MS module in Sec. 4.2, and we integrate out MLP-based MS module into pure MLP-based and grid-based NeRF methods to analyze the performance in Sec. 4.3, finally, we design another hybrid MS module in Sec. 4.4.

4.1 Neural Feature Fields

To implement our multi-space neural radiance fields, the underlying field must have the following intrinsics: the reconstructed scene must be 3D consistent, as the subspaces are independent scenes, and we try to model each subspace at scene-level; there must be ways to render 2D images and pixel-level composition maps for each views, because the model is only supervised by the composed images. Though we can modify neural radiance fields with an additional channel to output the composition information, we experimentally verify that adding additional channel is suboptimal in Sec. 6.5. We can also use neural feature fields to encode both RGB maps and composition maps for rendered views, which is originally designed for memory saving [30] replaces the output colors $\mathbf{c}_i$ of radiance fields by features $\{\mathbf{f}_i^k\}$ of d dimension. When rendering, neural feature fields follow the same 3D points sampling and volumetric rendering scheme as in Sec. 3.1, except the rendered map is feature map $\hat{\mathbf{F}}(\mathbf{r})$ instead of $\hat{\mathbf{C}}(\mathbf{r})$, then the color map $\hat{\mathbf{C}}(\mathbf{r})$ can be decoded from rendered feature map by a small MLP Θ:

$$\{\mathcal{F}(\mathbf{r})\} \xrightarrow{\Theta} \{\mathbf{C}(\mathbf{r})\} \tag{4}$$

and the model is supervised by the Ground-Truth images as in (3).

Though neural feature fields is capable of encoding both RGB maps and subspace composition maps for each view, we need to further validate that the neural feature fields are 3D consistent. With the development of NeRF-based methods, there are two main branches of efforts that improve the rendering quality or rendering speed of NeRF-based methods, and the representative methods are Mip-NeRF 360 [4] and iNGP [21]. The former uses better positional parameterization, bigger MLPs, and improved regularization to improve rendering quality, and the latter designs unique hash positional encoding, occupancy grid-guided sampling strategy, and highly optimized CUDA library to improve the rendering speed. As we aim to design a unified framework that improves the rendering quality on reflective surfaces of both NeRF-based methods, we build neural feature fields based on Mip-NeRF 360 and iNGP

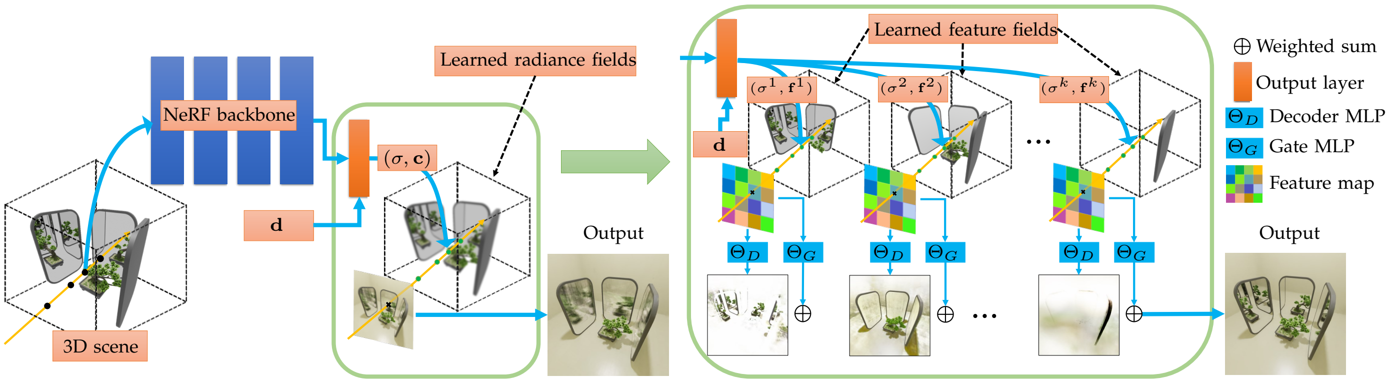


Fig. 7: Our multi-space module only modifies the output and volumetric rendering part of the network. The original NeRF calculates a pair of density σ and radiance $\mathbf{c}$ to get the accumulated color. Our output layer produces pairs of densities $\{\sigma^k\}$ and features $\{\mathbf{f}^k\}$, which correspond to multiple parallel feature fields. Then, we use volumetric rendering to get multiple feature maps. Two simple MLPs, *i.e.*, Decoder MLP and Gate MLP, are utilized to decode RGB maps and pixel-wise weights from these feature maps.

separately, and carry out experiments on the Realistic Synthetic 360° dataset from NeRF [2]. As shown in Tab. 1 and Fig. 6, neural feature fields are equivalent to neural radiance fields in terms of novel view synthesis.

4.2 Multi-Space module with feature fields

Based on the analysis in Sec. 4.1, we propose a compact Multi-Space module (MS module) using the neural feature field scheme [30], to sufficiently extract multi-space information from standard NeRF backbone network structures with only small computational overheads. Specifically, the MS module will replace the original output layer of the NeRF backbone. Below we describe the detailed architecture of our module.

As shown in Fig. 7, our MS module only modifies the output part of vanilla NeRF. Vanilla NeRF computes single density σ_i and radiance $\mathbf{c}_i$ for each position along a ray casting through the scene and performs volumetric rendering using (2) to get the accumulated color. On the contrary, our multi-head layer replaces the neural radiance field with the neural feature fields [30]. Specifically, the modified output layer gives K densities $\{\sigma_i^k\}$ and features $\{\mathbf{f}_i^k\}$ of d dimension for each position along a ray with each pair corresponding to a sub-space, where K and d are hyperparameters for the total sub-space number and the feature dimension of the neural feature field, respectively.

We then integrate features along the ray in each sub-space to collect K feature maps that encode the color information and visibility of each sub-space from a certain viewpoint. As all pixels are calculated the same way, we denote each pixel as $\{\mathcal{F}^k\}$ for simplicity and describe the operation at the pixel level. Each pixel $\{\mathcal{F}^k\}$ of the feature maps is calculated using:

$$\hat{\mathcal{F}^k}(\mathbf{r}) = \sum_{i=1}^{N} T_i^k(1 - \exp(-\sigma_i^k \delta_i))\mathbf{f}_i^k, \tag{5}$$

where the superscript k indicates the sub-space that the ray casts through. The k-th density σ_i^k and feature $\mathbf{f}_i^k$ correspond to the k-th sub-space. $T_i^k = \exp(-\sum_{j=1}^{i-1} \sigma_j^k \delta_j)$ and $\delta_i = t_i - t_{i-1}$ are similarly computed as in (2).

Then, $\{\mathcal{F}^k\}$ is decoded by two small MLPs, each with just one hidden layer. The first is a Decoder MLP that takes $\{\mathcal{F}^k\}$ as input and outputs RGB vectors. The second is a Gate MLP that takes $\{\mathcal{F}^k\}$ as input and outputs weights that control the visibility of certain sub-spaces. Specifically, we use:

$$\{\mathcal{F}^k\} \xrightarrow{\Theta_D} \{\mathbf{C}^k\}, \{\mathcal{F}^k\} \xrightarrow{\Theta_G} \{w^k\}, \tag{6}$$

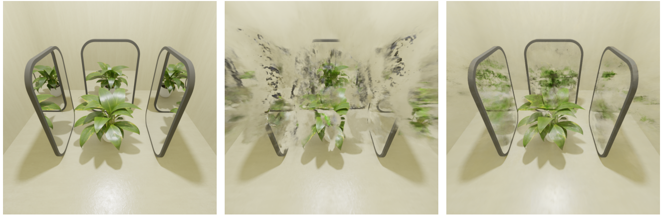
where Θ_D represents the Decoder MLP, and Θ_G represents the Gate MLP. In the end, the MS module applies the softmax function to $\{w^k\}$ as the color contribution of each subspace to form the final render results:

$$\hat{\mathbf{C}}(\mathbf{r}) = \frac{1}{\sum_{i=1}^{K} \exp(w^i)} \sum_{k=1}^{K} \exp(w^k)\mathbf{C}^k. \tag{7}$$

(7) needs no additional loss terms compared with the vanilla NeRF method. As a result, the above light-weighted MS module is able to serve as an enhancement module onto the conventional NeRF-like backbone networks, and we will show that our approach achieves significant enhancements in Sec. 6.4.

4.3 Grid-based Multi-Space NeRF with feature fields

Grid-based techniques [21] promote the development of NeRF-based methods by a huge step, especially in acceleration fields where grid-based NeRF methods are able to converge to comparably high quality in just minutes, while pure MLP-based methods require hours to days of training. These methods follow the design convention of small MLPs and learnable explicit parameters organized in 2D/3D grids, where the grids explicitly encode coordinate-related information, and the MLPs act more like decoders. Grid-based NeRF methods follow the rendering equation in (2), except that the positional encoding function (1) is replaced by a query function $Q(\mathbf{V}, \mathbf{p})$ mapping sampled points $\{\mathbf{p}_i\}$ into the values from the corresponding positions in the grid $\mathbf{V} \in \mathbb{R}^{\mathrm{d}_x \times \mathrm{d}_x \times \mathrm{d}_z \times \mathrm{d}_v}$, where $\mathrm{d}_x, \mathrm{d}_y, \mathrm{d}_z$ represents the grid resolution along x-, y-, and z-axis while d_v represents the dimension of queried features. As these



(a) Ground-Truth (b) w/o our module (c) w/ our module

Fig. 8: Visual comparison between iNGP with and without our MS module in Sec. 4.2.

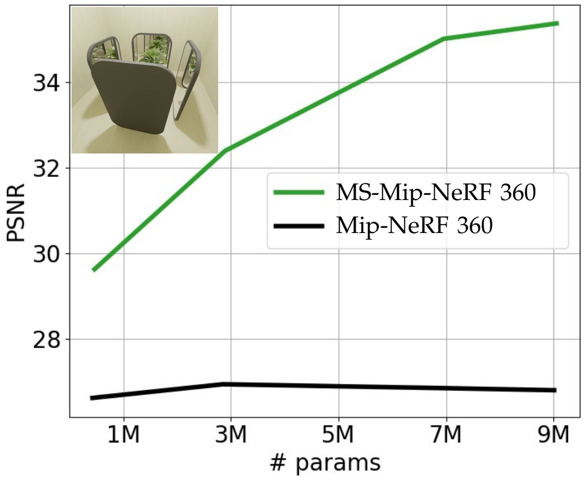
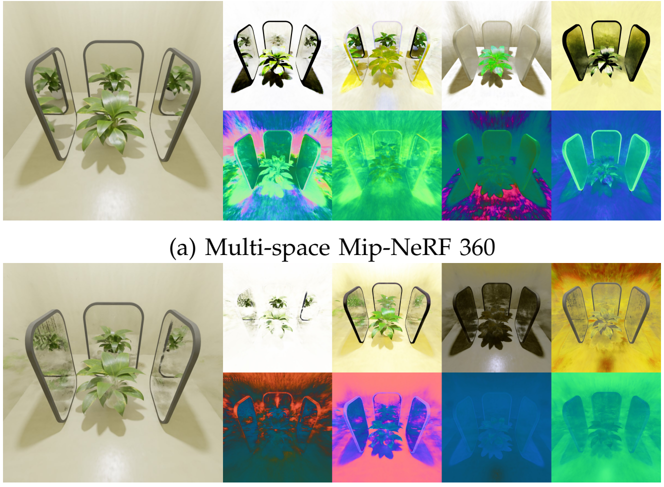


Fig. 9: We integrate our MLP-based multi-space module with 6 subspaces into the Mip-NeRF 360 method and scale the size of the NeRF MLP in the model by the network depth and width; then, we evaluate the performance by PSNR on the shown scene with the circle camera path.

methods render images from single radiance fields like previous methods, they also fail to render high-quality reflection and refraction.

We choose the iNGP [21] implemented with the proposal estimator in [62] as our baseline and simply integrate our MS module in Sec. 4.2 into it as multi-space iNGP. We conduct experiments on the Scene04 with circle camera path from our synthesized dataset in Sec. 5.2, because this scene contains relatively complex reflections and simple objects. As shown in Fig. 8, though our MS module prevents the model from collapsing, it still struggles to render high-quality reflections.

The results in Tab. 1 also show that the Mip-NeRF 360 gains performance increase from feature fields, while the performance of iNGP drops when integrated with feature fields, therefore, we suspect that the performance of our MS module with feature fields should be related to the MLP capacity of NeRF model. To validate the above speculation, we carry out experiments on Scene05 with the circle camera path from our synthesized dataset in Sec. 5.2, which contains infinite reflections and relatively simple objects, therefore, the performance improvements are clearer. We integrate the MS module in Sec. 4.1 with Mip-NeRF 360 and scale the base MLP size by network depth and width; the results in Fig. 9 show that bigger MLPs gain more performance increase from our MS module.



(a) Multi-space Mip-NeRF 360

(b) Multi-space iNGP

Fig. 10: Visualization of the sub-spaces rendered from Mip-NeRF 360 and iNGP with our MS module in Sec. 4.2. The left image is the composed image, and the right ones are sub-space RGB maps and sub-space feature maps with PCA transformation.

Also, we visualize the RGB maps and the feature maps visualized by PCA transformation of four sub-spaces rendered by the multi-space iNGP and multi-space Mip-NeRF 360 in Fig. 10, which shows that bigger MLPs with our MS module can reconstruct cleaner sub-spaces, while iNGP is designed with small MLPs and directly integratd with MS module from Sec. 4.2 can only reconstruct suboptimal subspaces.

4.4 Hybrid Multi-Space Module

Though MS module with feature fields exhibit good performance with MLP-based NeRF methods, it is not compatible with grid-based methods. Considering the representation ability of small MLPs, we comprise to model the sub-spaces in grid-based methods using neural radiance fields as shown in Fig. 11. Given sampled points $\{\mathbf{p}_i\}$ along the ray and the explicit grid parameters $\mathbf{V}$, we modify the output layer to map each features vector queried by $Q(\mathbf{V}, \mathbf{p})$ and the ray direction $\mathbf{d}$ to K densities $\{\sigma_i^k\}$ and colors $\{c_i^k\}$, which model multiple radiance fields instead of multiple feature fields, where K represent the total number of sub-spaces. Then, a multi-space integration in (5) is performed, except that we integrate K colors $\{c_i^k\}$ instead of features $\{\mathbf{f}_i^k\}$, to form K color maps $\{\mathbf{C}^k\}$.

We decompose the multi-space composition information from the NeRF model, and compress it into a small MLP, as in scenes with reflections and refractions, the visibility of virtual images is only related to 3D positions and view directions. Specifically, we design another branch with a small pure MLP network, which maps each 3D positions $\{\gamma(\mathbf{p}_i)\}$, where $\gamma(\dots)$ is the positional encoding from (1), and view directions $\mathbf{d}$ to features $\{\mathbf{f}_i\}$ of d dimension. Note that we record one feature vector

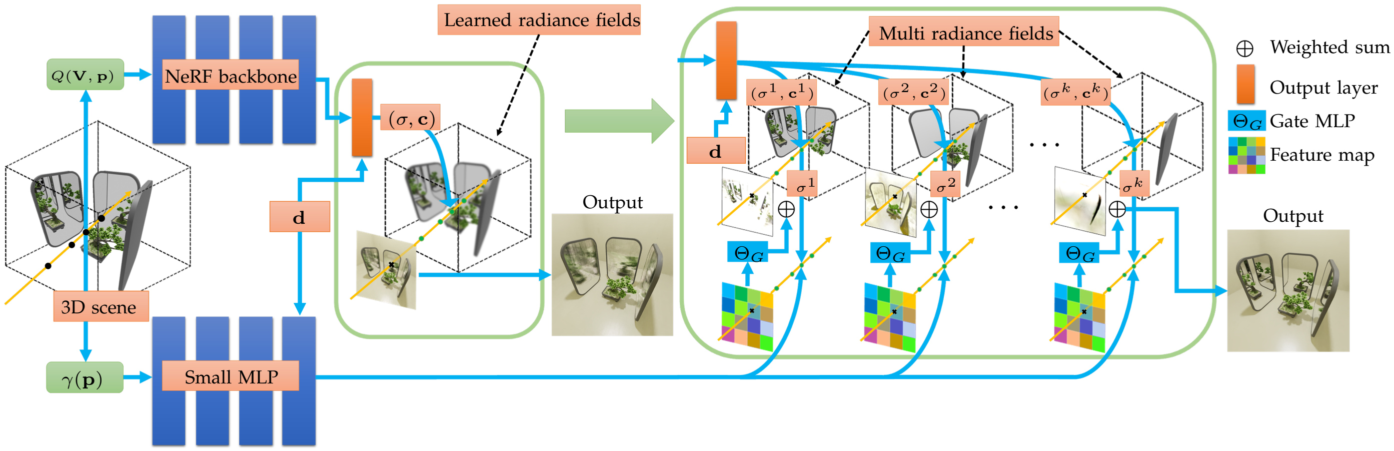


Fig. 11: The illustration of the proposed hybrid MS module. Considering the architecture of grid-based methods, we decompose the multi-space gate information from the radiance fields branch and use another small branch to model multiple feature fields, from which we decode the gate information. Along with the multiple radiance fields, we perform weighted sum to get the final rendering image.

for each position due to the model capacity. Then we put the features $\{\mathbf{f}_i\}$ into each sub-space along the rays, and perform volumetric rendering using the features $\{\mathbf{f}_i\}$ along with each of the K densities $\{\sigma_i^k\}$ from the multiple radiance fields branch to form K feature maps $\{\mathcal{F}^k\}$, as in (5). Finally, we decode the pixel-wise composition weight map from $\{\mathcal{F}^k\}$ following (6), and we compose the rendered results using the K color maps $\{\mathbf{C}^k\}$ from NeRF branch and the decoded weight map $\{w^k\}$ as in (7).

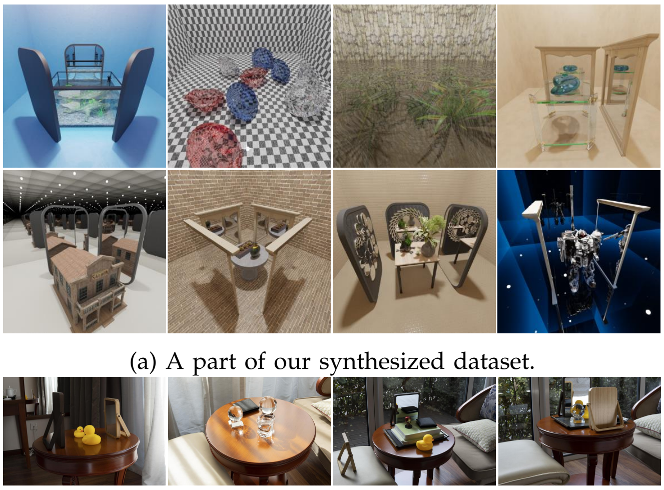
5 Dataset

5.1 Existing datasets

We briefly revisit the commonly used or most relevant datasets to our task and list their properties in Tab. 2. With these well-designed datasets, NeRF-based methods have achieved great improvements in many applications under various settings, such as novel view synthesis in unbounded scenes and 3D reconstruction. However, most existing dataset fails to cover scenes containing complex light paths with the camera moving 360-degree around, *e.g.*, a glass of water in front of a mirror, which is very common in our daily life. [16] propose the RFFR (Real Forward-Facing with Reflections) dataset, which contains 6 forward-facing scenes with reflective objects, such as transparent glass and mirrors. However, views behind the reflective objects cannot be evaluated, which is crucial for understanding reflections. [57] propose a dataset containing 4 scenes, where cameras move around the central refractive objects with a large view range up to 360-degree. However, it is a nearly object-level dataset, and the refractive is rather simple.

5.2 Our proposed dataset

As summarized in Sec. 5.1, there lacks a 360-degree dataset consisting of complex reflection and refraction to facilitate the related research. Therefore, we collect a 360-degree dataset comprising 33 synthetic scenes and 7 real captured scenes.



(a) A part of our synthesized dataset.

(b) A part of our real captured dataset.

Fig. 12: Demo scenes of our datasets (more in the supplementary). Our dataset exhibits diversities of reflection and refraction, which can serve as a benchmark for validating the ability to synthesize novel views with complex light paths.

For our synthesized part shown in Fig. 12a, we use an open source software Blender [61], and design our scenes with 3D models from BlenderKit, a community for sharing 3D models. As our dataset consists of complete scenes instead of single objects, we design three kinds of camera paths. The simplest one is a circle path, where we fix the height of our camera position with the camera looking at the center of the scene and moving the camera around a circle to render the whole scene. We render this path for all our scenes, where we uniformly sample 120 points along the circle and randomly choose 100 images for the training set, 10 for the validation set, and 10 for the test set. Besides, we design a 360-degree spiral path, where cameras gradually spiral up from the equator to the pole, looking at the center of the scene. To further evaluate the robustness of future

dataset	origin	applications	Type	viewpoints	properties	number
Realistic Synthetic 360°	[2]	view synthesis	S	360-degree	non-Lambertian	8
Real Forward-Facing	[2], [18]	view synthesis	R	forward-facing	non-Lambertian	8
Shiny	[17]	view synthesis	R	forward-facing	high-specular, refraction	8
Tanks and Temples(T&T)	[55]	view synthesis	R	360-degree	unbounded scenes	4
Mip-NeRF 360	[4]	view synthesis	R	360-degree	unbounded scenes	9
EikonalFields	[57]	view synthesis	R	360-degree	refraction	4
RFFR	[16]	view synthesis	R	forward-facing	reflection, semi-transparent	6
DTU	[58]	reconstruction	R	360-degree	non-Lambertian objects	15*
BlendedMVS	[59]	reconstruction	S	360-degree	non-Lambertian scenes	7*
Shiny Blender	[48]	view synthesis	S	360-degree	glossy materials	6
Ref-NeRF Real captured scenes	[48], [63]	view synthesis	R	360-degree	glossy materials	3

Table 2: Properties of a commonly used dataset for NeRF-based methods. 'S' and 'R' represent synthesized and real captured, respectively. We denote those unnamed datasets with the name of the methods. '*' refers to the number of scenes commonly used by NeRF-based methods, as the original dataset contains more scenes than noted, and we do not take them into consideration.

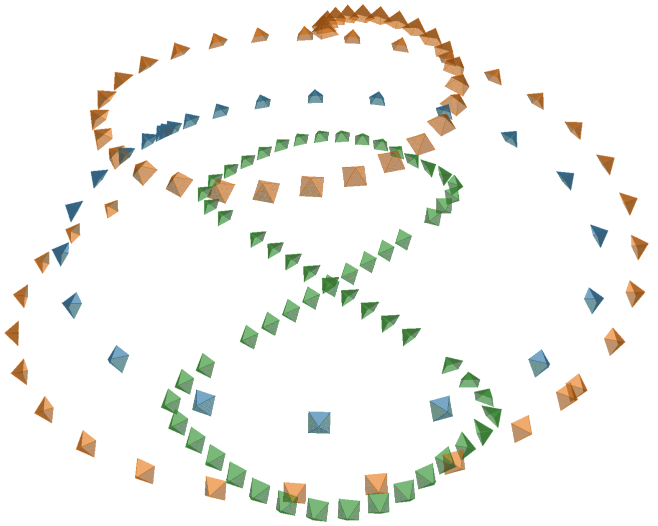


Fig. 13: Visualization of our designed camera paths. Blue cameras represent the circle path, green cameras belong to the mirror-passing-through path, and orange cameras represent the spiral path.

related methods, we design a novel mirror-passing-through path, where the cameras move through the mirrors in the scenes back and forth. We select 5 simple scenes and 5 difficult scenes from our dataset to render the above two paths, where we uniformly sample 300 points along the path and randomly choose 100 images as the training set and 200 as the test set. We visualize the three kinds of camera paths in Fig. 13.

The constructed dataset features a wide variety of scenes containing reflective and refractive objects. We include a variety of complexity of light paths, controlled by the number and the layout of the mirror(s) in the scene, where the number of mirrors ranges from 1 up to tens of small pieces. Note that even a scene in our dataset with only one mirror is more challenging than RFFR [16], as our camera moves from the front to the back of the mirror(s). Besides, we also construct rooms with mirror walls that can essentially be treated as unbounded scenes, where we add mirrors in the center of the room and create unbounded virtual images. We further build challenging scenes, including a combination of reflection and refraction. As shown in Fig. 14, our dataset exhibits much more challenging properties. Furthermore, we render instance-level masks for each reflective and refractive object in our synthesized scenes.

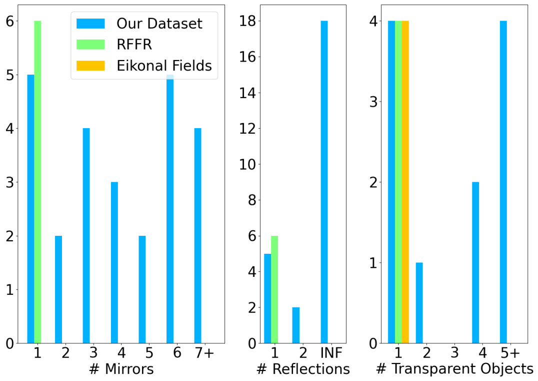


Fig. 14: We quantitatively compare our dataset with two most related datasets, *i.e.*, RFFR [16] and EikonalFields [57]. We statistically count the number of mirrors and transparent objects, which are directly related to reflection and refraction, respectively. Furthermore, we quantify the complexity of light paths by the maximum number of reflections occurring in the scene. Note that in our scenes with more than two mirrors, there must exist two facing mirrors; therefore, the maximum reflections jump from 2 to infinite.

We also include 7 captured real scenes with complex light conditions shown in Fig. 12b. We construct our scenes using two mirrors, one glass ball with a smooth surface, one glass ball with a diamond-like surface, a few toys, and a few books. We capture pictures randomly with 360-degree viewpoints.

6 Experiments

6.1 Baselines, Hyperparameters, and benchmarks

To thoroughly evaluate the superiority and robustness and demonstrate the potential of our method, we con-

	PSNR↑	SSIM↑	LPIPS↓	# Params
NeRF	23.79/35.24	0.662/0.902	0.240	1.159M
MS-NeRF$_S$	26.59/35.52	0.737/0.900	0.232	1.201M
MS-NeRF$_M$	26.71/35.72	0.741/0.901	**0.226**	1.245M
MS-NeRF$_B$	**26.95**/**35.87**	**0.748**/**0.903**	**0.226**	1.311M
Mip-NeRF	24.47/35.97	0.693/0.906	0.245	0.613M
Ref-NeRF	25.58/36.65	0.716/**0.911**	**0.210**	0.713M
MS-Mip-NeRF$_S$	28.08/36.60	0.779/0.906	0.224	0.634M
MS-Mip-NeRF$_M$	28.44/36.76	0.788/0.907	0.222	0.656M
MS-Mip-NeRF$_B$	**28.48**/**36.83**	**0.789**/0.907	0.220	0.689M
Mip-NeRF 360	24.20/37.06	0.733/**0.925**	**0.150**	9.007M
MS-Mip-NeRF 360	**28.35**/**37.65**	**0.822**/0.923	**0.150**	9.007M

(a) MS module from Sec. 4.2 on our synthetic dataset with circle paths.

	PSNR↑	SSIM↑	LPIPS↓	# Params
TensoRF	24.25/33.83	0.697/0.937	0.196	17.34|4.00e-2M
MS-TensoRF	**26.96**/**37.74**	**0.767**/**0.951**	**0.147**	17.34|4.60e-2M
iNGP	24.04/29.98	0.743/0.915	0.201	14.23|3.28e-2M
MS-iNGP	**26.59**/**33.67**	**0.800**/**0.930**	**0.157**	14.23|3.33e-2M

(b) MS module from Sec. 4.4 on our synthetic dataset with circle paths.

	PSNR↑	SSIM↑	LPIPS↓	# Params
Mip-NeRF 360	25.21/39.15	0.795/**0.969**	0.102	9.007M
MS-Mip-NeRF 360	**31.31**/**39.98**	**0.894**/0.966	**0.098**	9.052M
Mip-NeRF 360	24.87/34.02	0.828/**0.915**	**0.184**	9.007M
MS-Mip-NeRF 360	**26.74**/**34.03**	**0.851**/0.910	0.188	9.052M

(c) MS module from Sec. 4.2 on our synthetic dataset with spiral paths and mirror-passing-through paths. The first two rows are spiral paths, and the last two rows are mirror-passing-through paths.

	PSNR↑	SSIM↑	LPIPS↓	# Params
Mip-NeRF 360	26.70	0.889	**0.113**	9.007M
MS-Mip-NeRF 360	**28.14**	**0.891**	0.119	9.052M

(d) Comparisons on the real captured part of our dataset.

	PSNR↑	SSIM↑	LPIPS↓	# Params
NeRFReN	35.26	0.940	0.081	1.264M
MS-NeRF$_T$	**35.93**	**0.948**	**0.066**	1.295M

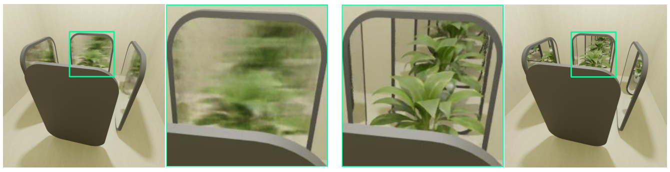
(e) Comparisons on RFFR dataset.

Table 3: Quantitative comparisons with existing methods. Parameters with the pattern "A|B" refer to grid parameters and the MLP parameters; others are all MLP parameters. For PSNR and SSIM metrics on our synthesized dataset, we split the rendered images according to the masks of reflective and refractive areas, and the metrics are reported as "reflective and refractive areas/other areas".

duct various experiments based on different datasets with different baselines and our modules of different scales. We select 5 representative NeRF-based methods and categorize them into two parts: a) MLP-based NeRF methods, including NeRF [2], Mip-NeRF [19] and Mip-NeRF 360 [4]; b) grid-based NeRF method, including TensoRF [20] and iNGP [21].

	PSNR↑	SSIM↑	LPIPS↓	# Params
Mip-NeRF	30.74	0.942	**0.047**	1.264M
MS-Mip-NeRF$_B$	**30.81**	**0.943**	**0.047**	1.295M
Mip-NeRF	**25.78**	**0.775**	**0.213**	1.264M
MS-Mip-NeRF$_B$	25.59	0.764	0.223	1.295M

Table 4: Results on the Realistic Synthetic 360° dataset (first two rows) and Real Forward-Facing dataset (last two rows).



(a) Mip-NeRF 360 (b) MS-Mip-NeRF 360

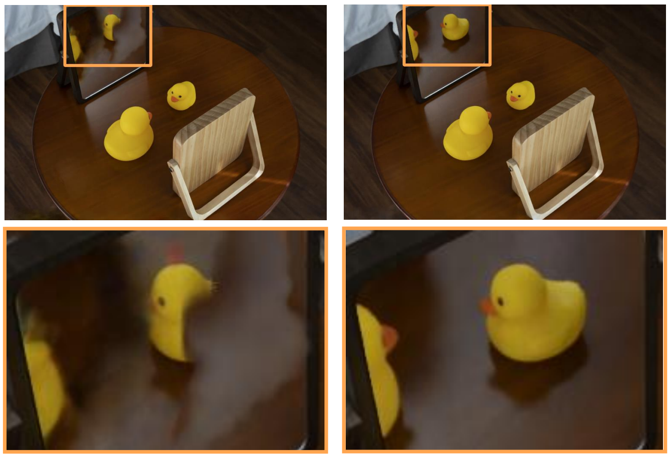
Fig. 15: Visual comparison between Mip-NeRF 360 and MS-Mip-NeRF 360. Our module can extend Mip-NeRF 360 to model unbounded virtual scenes.

MLP-based NeRF methods contain representative methods for novel view synthesis, which typically encode the underlying radiance fields into the weights of MLP and aim at rendering high-quality images. We conduct experiments with our MLP-based MS module in Sec. 4.2 based on these methods on our dataset to demonstrate the superiority of our scheme. Grid-based NeRF methods feature hybrid representation, specifically combinations of small MLP and discrete learnable/fixed parameters organized in 3D/2D grids, which heavily reduces the parameters to be optimized in each iteration and converge to relatively high-quality rendering in a very short time. TensoRF and iNGP construct the networks using learnable parameters in 2D and 3D grids respectively with small MLPs; therefore, we integrate our hybrid MS module described in Sec. 4.4 into them to validate our methods. We conduct various experiments with these baselines on different datasets to demonstrate the superiority and generalization of our scheme.

As our MLP-based MS module is quite simple, we can scale our module by three hyperparameters, which we refer to as K for the sub-space number, d for the dimension of output features, and h for the hidden layer dimension of Decoder MLP and Gate MLP, respectively. For the hybrid MS scheme, there are also hyperparameters K, d, and h, except that d and h only control the Gate MLP. All the training details can be found in the supplementary. We report our results with three commonly used metrics: PSNR, SSIM [64], and LPIPS [65].

6.2 Experiments on MLP-based NeRF methods

We scale our module from Sec. 4.2 and integrate it with NeRF [2], Mip-NeRF [19], and Mip-NeRF 360 [4] to conduct experiments, and we follow most default settings from [2], [4], [16], [19], [48], except that we



(a) Mip-NeRF 360 (b) MS-Mip-NeRF 360

Fig. 16: Visual comparison between Mip-NeRF 360 and MS-Mip-NeRF 360 on the real captured part of our dataset. Our method is robust enough to recover virtual images in the real world.

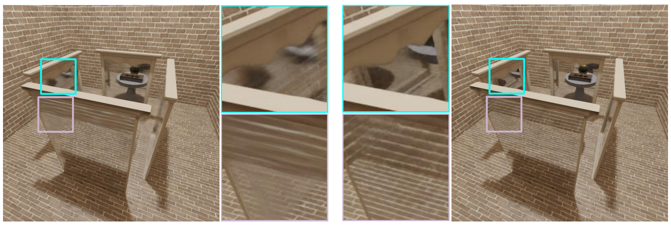
use 1024 rays per batch and train 200k iterations for all experiments on all scenes.

For NeRF [2] and Mip-NeRF [19] based experiments, we build MS-NeRF$_S$ and MS-Mip-NeRF$_S$ with hyperparameters $\{K = 6, d = 24, h = 24\}$, similarly MS-NeRF$_M$ and MS-Mip-NeRF$_M$ with hyperparameters $\{K = 6, d = 48, h = 48\}$, and MS-NeRF$_B$ and MS-Mip-NeRF$_B$ with hyperparameters $\{K = 8, d = 64, h = 64\}$. For Mip-NeRF 360 [4] based experiments, we construct MS-Mip-NeRF 360 with hyperparameters $\{K = 8, d = 32, h = 64\}$. Moreover, we provide a comparison with Ref-NeRF [48] because it uses Mip-NeRF as a baseline and possesses an outstanding ability to model glossy materials. As NeRF, Mip-NeRF, and Ref-NeRF are all designed for bounded scenes, we evaluate these methods on the first 25 scenes with circle paths while the last 8 scenes are equivalent to unbounded scenes. For Mip-NeRF 360 based experiments, we evaluate on all scenes with all paths.

We also compare our method with NeRFReN on the RFFR dataset [16]. NeRFReN is a specially designed two-branch network based on vanilla NeRF for mirror-like surfaces in forward-facing scenes. Thus we construct a tiny version of our method, referred to as MS-NeRF$_T$, based on NeRF with hyperparameters $\{K = 2, d = 128, h = 128\}$. Here we use two sub-spaces as NeRFReN tries to decompose reflective surfaces into two parts, and we want to show that our space decomposition is more effective. For a fair comparison, we re-train NeRFReN using the official code under the provided settings on the RFFR dataset, except that we set the number of the used mask to 0 as our method requires no extra mask.

6.3 Experiments on grid-based NeRF methods

We carry out experiments on grid-based methods to demonstrate the compatibility of our scheme with other applications instead of just pure MLP-based rendering methods.



(a) Ref-NeRF (b) MS-Mip-NeRF$_B$

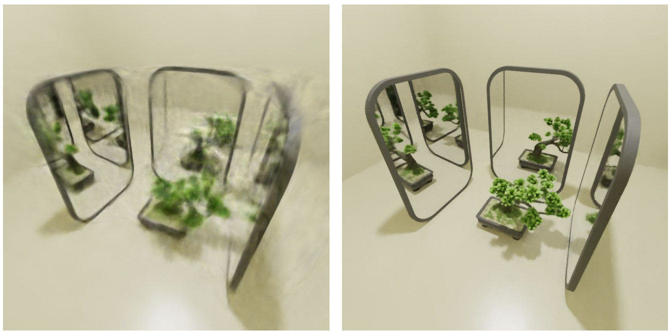
Fig. 17: Visual comparison between MS-Mip-NeRF$_B$ and Ref-NeRF. Our method significantly outperforms Ref-NeRF on reflective surfaces.

iNGP [21] and TensoRF [20] achieve fast convergence using combinations of learnable 3D/2D feature grids and very small MLPs. We separately construct the MS-iNGP and MS-TensoRF with our hybrid MS module proposed in Sec. 4.4 using hyperparameters $\{K = 4, d = 8, h = 32\}$, and we initialize the extra branch with a similar MLP but of smaller size to the one in the main branch. We follow all the default settings [20] for MS-TensoRF experiments, and we implement the MS-iNGP based on the implementation from [62] with their proposal estimator. We evaluate TensoRF on the first 25 scenes with circle paths, as this method is designed for bounded scenes. On the contrary, iNGP with the proposal estimator [62] is able to reconstruct unbounded scenes, therefore, we evaluate iNGP-related models on all scenes with circle paths.

6.4 Comparisons

Quantitative comparisons. As reported in Tab. 3a, our MLP-based module can be integrated into most NeRF-like models and improve performance by a large margin with minimal extra cost introduced. Especially in Mip-NeRF 360-based experiments, our module exhibits better results of 4.15 dB improvement in PSNR using merely 0.5% extra parameters without degrading the performance on non-mirror regions. Besides, our Mip-NeRF-based models also outperform Ref-NeRF [48] by a large margin on mirror, which is a variant based on Mip-NeRF with the outstanding ability to model glossy materials.

The synthetic part with mirror-passing-through paths and spiral paths and the real captured scenes are much more challenging, therefore, we demonstrate our results compared with the pure MLP-based state-of-the-art method Mip-NeRF 360 in Tab. 3c and Tab. 3d on these benchmarks. Our approach also shows large improvements, which indicates the robustness of our scheme. Specifically, the spiral camera paths introduce wider view changes on the virtual images, while the mirror-passing-through camera paths introduce rapid disappearance of virtual images along the rays, and the results indicate that our scheme is capable of handling these without degrading the performance on non-mirror regions.



(a) NeRFReN (b) MS-NeRF$_B$

Fig. 18: (a) Trained with accurately labeled masks, NeRFReN even fails to render ordinary parts of the scene in 360-degree scenes with mirrors. (b) Our method requires no extra manually labeled masks and renders high-quality images.

	PSNR↑	SSIM↑	LPIPS↓	# Params
Mip-NeRF 360	29.84	0.942	0.100	9.007M
MS-Mip NeRF 360$_h$	33.59	0.958	0.081	9.060M
MS-Mip NeRF 360$_p$	37.50	0.972	0.064	9.018M
MS-Mip NeRF 360	**38.84**	**0.977**	**0.054**	9.052M

(a) Ablation study on our MLP-based MS module architecture.

	PSNR↑	SSIM↑	LPIPS↓	# Params
MS-TensoRF-grid	32.07	0.932	0.130	21.73|5.60e-2M
MS-TensoRF	**34.72**	**0.938**	**0.113**	17.23|4.60e-2M

(b) Ablation study on our hybrid MS module architecture.

Table 5: Ablation study on our module architecture. Parameters with the pattern "A|B" refer to grid parameters and the MLP parameters; others are all MLP parameters.

Results in Tab. 3b indicate that our hybrid MS module is compatible with TensoRF and iNGP. As shown, our module can improve the performance on reflective regions and also help stabilize the reconstruction on other regions.

On the RFFR dataset, which contains forward-facing reflective surfaces in the scenes, our approach achieves better results when no manually labeled masks are provided in training as in Tab. 3e. For NeRFReN, we re-train the model using the official code following the provided setting, except the number of masks used for reflective surfaces is 0.

The Realistic Synthetic 360° dataset and Real Forward-Facing dataset are first introduced in [2], which are commonly used for evaluating the ability of NeRF-based methods in novel view synthesis. We also train Mip-NeRF and MS-Mip-NeRF$_B$ on these datasets because Mip-NeRF is a commonly used backbone for NeRF-based methods. The results are reported in Tab. 4, which demonstrate that our multi-space module has no influence on the representation ability of NeRF-based methods on common materials.

All of the above experiments demonstrate the superiority and compatibility of our method.

Qualitative comparisons and discussions. Besides quantitative comparisons, we summarize the advantages of our modules and support them by qualitatively or quantitatively comparing our methods with the corresponding baselines.

Qualitative comparisons with the state-of-the-art MLP-based method, Mip-NeRF 360, are shown in Fig. 1, Fig. 15, and Fig. 16. Our method renders high-fidelity virtual images, bounded and unbounded, in both synthetic and real-world scenes. Furthermore, our MS module is also robust in sudden changes when the camera moves through the mirror, as in Fig. 1.

A qualitative comparison with Ref-NeRF [48], which understands virtual images as textures using the conventional NeRF backbone, is shown in Fig. 17. As Ref-NeRF is also based on Mip-NeRF, we compare our Mip-NeRF-based variant with Ref-NeRF using the same baseline and use comparable parameters (specifically ours 0.689M and Ref-NeRF 0.713M) in the comparison. Again, the qualitative results show our significant improvements in rendering reflective surfaces.

We also compare with the NeRFReN model, which requires accurately labeled masks of the reflective regions during training and handles forward-facing reflective surfaces only. In this comparison, we train their model on our synthesized dataset with extra accurate reflection masks provided. Fig. 18 shows that their model fails to recover 360-degree high-fidelity rendering while our approach succeeds.

6.5 Ablation studies

In this section, we evaluate the design of our module and explore the relation between the number of sub-spaces and the number of virtual images.

Ablation on using neural feature field. We implement a module that simply outputs K scalars $\{\sigma_i^k\}$ and K RGB-G vectors $\{\hat{\mathbf{c}}_i^k\}$ of four dimensions, which consists of three RGB channels and one scaler that controls the gating information of multi-space composition. When rendering, we use the same integral equation as NeRF to accumulate the K RGB-G vectors $\{\hat{\mathbf{c}}_i^k\}$ and get the RGB-G maps of each sub-space, then we split the RGB-G maps into RGB maps and sub-space composition maps(gating maps), finally, we apply softmax over the gating maps along the sub-space channels and use it as the weight to compose sub-space RGB maps to form the final rendering results. We integrate this design into Mip-NeRF 360 noted as MS-Mip NeRF 360$_p$, where we set $K = 8$. We also integrate the hybrid module from Sec. 4.4 into Mip-NeRF 360 noted as MS-Mip NeRF 360$_h$, the results are in Tab. 5a. We also exhibit a few visual results in Fig. 19, which indicate that a simple multi-space radiance field assumption can help the model partially overcome the violation of reflections, but will also introduce the over-smoothing problem because of the lack of an efficient multi-space composition strategy.

Ablation on the sub-space number. In our Euclidean space, one can control the number of virtual sub-spaces

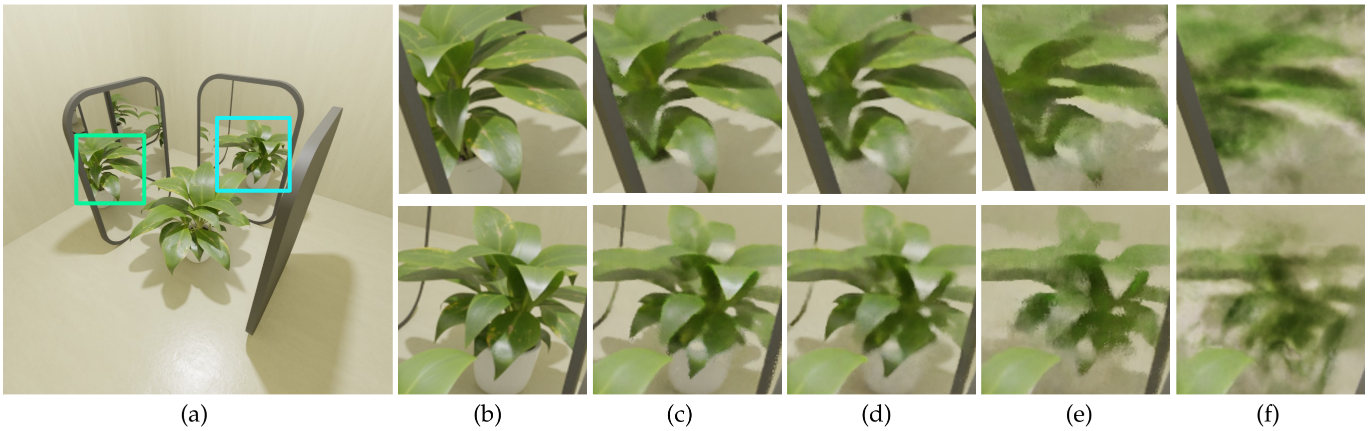


Fig. 19: Detailed comparisons on rendered images. (a) Overview; (b) Ground-Truth; (c) MS-Mip-NeRF 360; (d) MS-Mip NeRF 360$_p$; (e) MS-Mip NeRF 360$_h$; (f) Mip-NeRF 360.

by the number and the layout of the mirror(s). For example, when two mirrors are facing each other, there could be infinitely recursive virtual image spaces, but when two mirrors are placed back against each other, there will be just one virtual image behind each mirror. To provide a guideline for the design of our module, we choose two scenes consisting of two mirrors with different layouts from our synthesized part of the dataset and train NeRF-based variants of different sub-space numbers and different feature dimensions.

We construct our variants based on NeRF with the output feature dimensions $d \in \{24, 48, 64\}$ and the number of sub-spaces $K \in \{2, 4, 6, ..., 16\}$, then we train our models on the two scenes and report the results using PSNR. Our results in Fig. 20 show that the number of sub-spaces is not required to match the actual number of virtual image spaces, and 6 sub-spaces can guarantee stable learning for multi-space radiance fields. Moreover, feature fields with dimension $d = 24$ already encode enough information for composition, but for stable performance $d = 48$ is better.

Ablation on the hybrid MS module. We further design a grid-based MS module to provide a guideline for the MS module in grid-based NeRF methods. Specifically, we replace the positional encoding $\gamma(\mathbf{p})$ in our hybrid MS module from Sec. 4.4 with features queried by $Q(\mathbf{V}_{vis}, \mathbf{p})$, where $\mathbf{V}_{vis}$ is a grid similar to the $\mathbf{V}$ in the main branch but of smaller size. We refer to the model integrated with grid-based MS module as MS-TensoRF-grid, and we conduct comparisons with the MS-TensoRF model in Sec. 6.3 on the first 10 scenes in our synthesized scenes with circle paths. As in Tab. 5b, the results indicate that the multi-space scheme is more related to the positions, therefore, our hybrid MS module reaches better performance with less cost.

Ablation on the input image number. To further validate the robustness of our scheme, we compare the MS-Mip-NeRF 360 model with the Mip-NeRF 360 baseline on Scene01-Scene05 from our synthesized dataset with all three camera paths with 100, 75, or 30 randomly selected training views. We report the detailed results in Tab. 7, which indicates that our scheme robustly improves the performance of the baseline model on reflective areas with varying numbers of input images. We provide more visual comparisons in the Supplementary to better explore the behavior of our scheme.

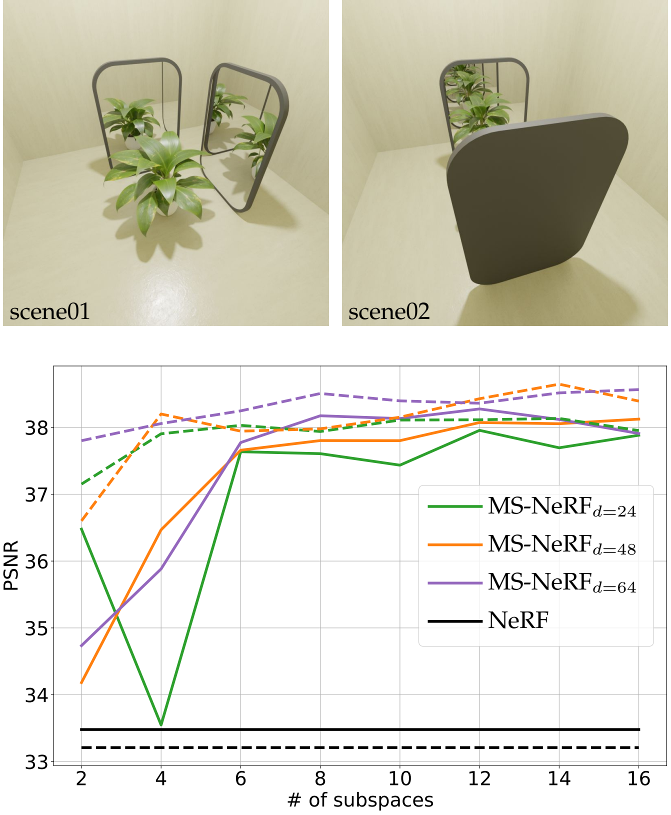
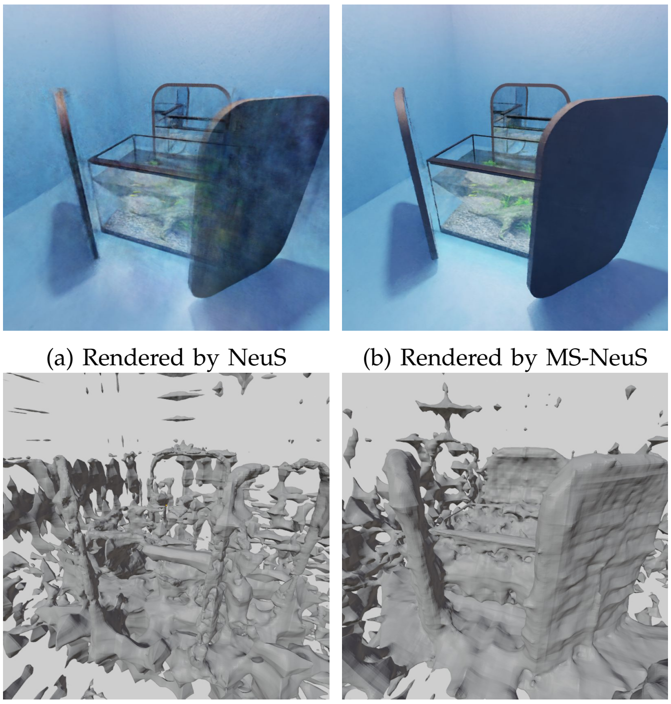


Fig. 20: We use PSNR to quantitatively evaluate the ablation experiments on scene01 and scene02 and plot the results with solid and dotted lines, respectively.

6.6 Reconstructed geometry

Though NeuS [22] integrates SDF fields with volumetric rendering for 3D reconstruction, it inherits the weaknesses of density-based volumetric rendering on the reconstruction of mirror surfaces. To validate the



(a) Rendered by NeuS (b) Rendered by MS-NeuS

(c) Mesh from NeuS (d) Mesh from MS-NeuS

Fig. 21: Our MS module helps NeuS render high quality novel views for reflections, and the reconstructed geometry of mirrors is improved.

	PSNR↑	SSIM↑	LPIPS↓	# Params
NeuS	25.86	0.831	0.258	27.96|0.16e-1M
MS-NeuS	**28.65**	**0.859**	**0.195**	27.96|0.37e-1M

Table 6: MS module from Sec. 4.2 based on NeuS on our synthesized dataset with circle paths. Parameters with the pattern "A|B" refer to grid parameters and the MLP parameters.

Model (# input images)	PSNR↑	SSIM↑	LPIPS↓
Mip-NeRF 360 (100)	21.84/40.17	0.660/0.980	0.087
MS-Mip-NeRF 360 (100)	**32.11/43.04**	**0.913/0.984**	**0.051**
Mip-NeRF 360 (75)	21.46/39.37	0.643/0.978	0.088
MS-Mip-NeRF 360 (75)	**31.44/42.20**	**0.890/0.982**	**0.055**
Mip-NeRF 360 (30)	19.08/34.51	0.549/0.966	0.102
MS-Mip-NeRF 360 (30)	**28.37/39.84**	**0.854/0.979**	**0.060**

(a) Circle camera path results.

Model (# input images)	PSNR↑	SSIM↑	LPIPS↓
Mip-NeRF 360 (100)	22.26/40.46	0.711/0.982	0.083
MS-Mip-NeRF 360 (100)	**32.30/42.39**	**0.905/0.984**	**0.049**
Mip-NeRF 360 (75)	21.69/39.60	0.692/0.980	0.084
MS-Mip-NeRF 360 (75)	**31.12/41.78**	**0.893/0.983**	**0.052**
Mip-NeRF 360 (30)	18.86/33.40	0.596/0.960	0.116
MS-Mip-NeRF 360 (30)	**25.14/36.49**	**0.792/0.969**	**0.086**

(b) Spiral camera path results.

Model (# input images)	PSNR↑	SSIM↑	LPIPS↓
Mip-NeRF 360 (100)	26.69/35.63	0.857/**0.962**	0.112
MS-Mip-NeRF 360 (100)	**29.39/35.83**	**0.896/0.962**	**0.104**
Mip-NeRF 360 (75)	25.25/**34.93**	0.841/**0.958**	0.118
MS-Mip-NeRF 360 (75)	**28.21**/34.60	**0.883**/0.955	**0.113**
Mip-NeRF 360 (30)	**21.60/25.11**	0.766/**0.889**	**0.237**
MS-Mip-NeRF 360 (30)	21.58/23.04	**0.782**/0.864	0.287

(c) Mirror-passing-through camera path results.

Table 7: Quantitative comparisons between MS-Mip-NeRF 360 and the baseline model on Scene01-Scene05 of all camera paths from our synthesized dataset with varying input images. We also separately report the metrics as done in Tab. 3.

method	training time↓	rendering time(per frame)↓
Mip-NeRF	5.20 h	21 s
MS-Mip-NeRF$_B$	6.10 h	23 s
Mip-NeRF 360	12.56 h	32 s
MS-Mip-NeRF 360	13.30 h	39 s
TensoRF	0.42 h	3.5 s
MS-TensoRF	1.25 h	11.3 s
iNGP	0.56 h	3.1 s
MS-iNGP	0.67 h	3.6 s

Table 8: The average training and rendering time comparisons on a single GeForce RTX 3090 GPU.

generalization of our scheme in solving multi-view inconsistency for novel view synthesis, we construct MS-NeuS by integrating our MS module from Sec. 4.2 into NeuS implemented in [66] with the hyperparameters $\{K = 4, d = 16, h = 32\}$. We conduct experiments on our synthesized dataset with circle paths, and the quantitative and qualitative comparisons on rendered views are in Tab. 6 and Fig. 21. As shown in Fig. 21, our scheme helps NeuS render better novel view images and reconstructs better geometry for the mirrors, but it still struggles to reconstruct clean geometry, because our MS module is designed for novel view synthesis and there is no geometric constraints on sub-spaces.

6.7 Rendering speed

Our scheme performs multiple volumetric rendering operations for each pixel, therefore, the time consumption is related to the importance sampling strategy and the backbone networks. We report the average training and rendering times in Tab. 8. For pure MLP-based methods, the bottleneck of time consumption lies in the network inference, and the number of sampled points for each ray is constant, therefore, integrating our scheme has a relatively small inference on the speed. TensoRF uses occupancy grid guided sampling strategy, and in our multi-space version, we record all occupied positions across all sub-spaces into one occupancy grid, therefore, the sampled points for each ray are much denser, therefore, the time consumption increases by a large margin. On the contrary, our iNGP version is constructed based on the NerfAcc framework with the proposal network-guided sampling strategy, and the number of sampled points is constant, therefore, the time consumption increases by less than one second. For all our modules, there are redundant points along the rays, and the possible solution is to sample points in different sub-spaces adaptively, which we leave for future research.

7 Discussion

Along with our motivation in Sec. 3.2 and extensive experiments in Sec. 6, we conduct more experimental analysis to investigate the mechanism of sub-space decomposition. The experiments in Sec. 7.1 reveal that our multi-space scheme equips the density fields with the ability to handle multi-view inconsistency without photometric supervision. The experiments in Sec. 7.2 further demonstrate that our multi-space scheme successfully decomposes multi-view inconsistent parts into different sub-spaces under the supervision of different losses. The above experiments confirm that the multi-space decomposition is only determined by the virtual or real images, and requires no additional regularizations.

7.1 Multi-space scheme with density fields

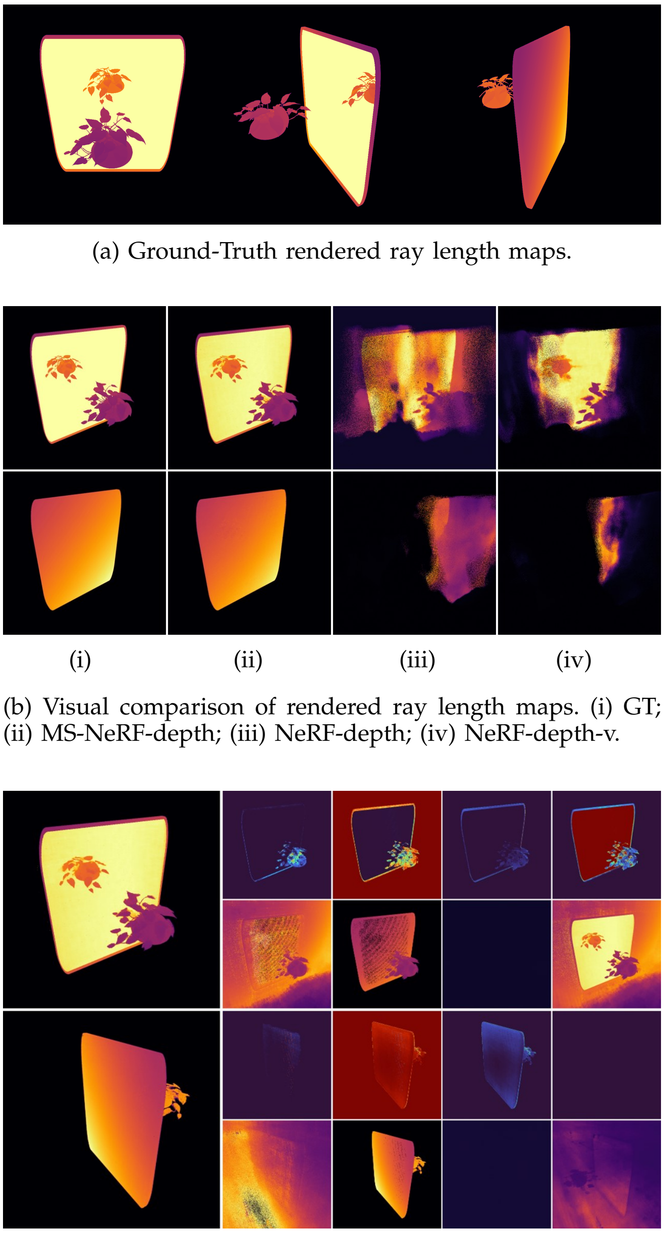
To investigate the mechanism of the multi-space scheme decomposing multi-view inconsistent space into sub-spaces, we conduct experiments on the density field without the radiance field head, which demonstrates clearer decomposition. Specifically, we choose the original NeRF model with only the density output head, referred to as NeRF-depth, therefore, the model is supervised by rendered depth maps via the rendering equation (2) as:

$$\hat{\mathbf{D}}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1-\exp(-\sigma_i\delta_i))t_i \tag{8}$$

with $T_i = \exp(-\sum_{j=1}^{i-1}\sigma_j\delta_j)$, $\delta_i = t_i - t_{i-1}$, and $\hat{\mathbf{D}}(\mathbf{r})$ is the rendered depth map. We construct the MS-NeRF-depth model by integrating the multi-space module in Sec. 4.2 into NeRF-depth with hyperparameters $\{K = 4, d = 8, h = 32\}$, and similarly only depth map is composed by:

$$\hat{\mathbf{D}}(\mathbf{r}) = \frac{1}{\sum_{i=1}^{K}\exp(w^i)}\sum_{k=1}^{K}\exp(w^k)\hat{\mathbf{D}}^k. \tag{9}$$

where w^k is the sub-space composition weights as in (6), and $\hat{\mathbf{D}}^k$ is the sub-space depth map. For fair comparison, we also build the NeRF-depth-v model based on NeRF-depth, which takes view directions as input before the output layer in addition to the positions because our module requires view directions as input. All the models are supervised by the depth map, and we directly utilize the MSE loss. We train these models on rendered ray length maps, where each pixel contains the physical transportation length of the ray traveling from the camera to the first non-reflective surfaces, following the circle camera path scene configuration as in Fig. 22a.


(a) Ground-Truth rendered ray length maps.

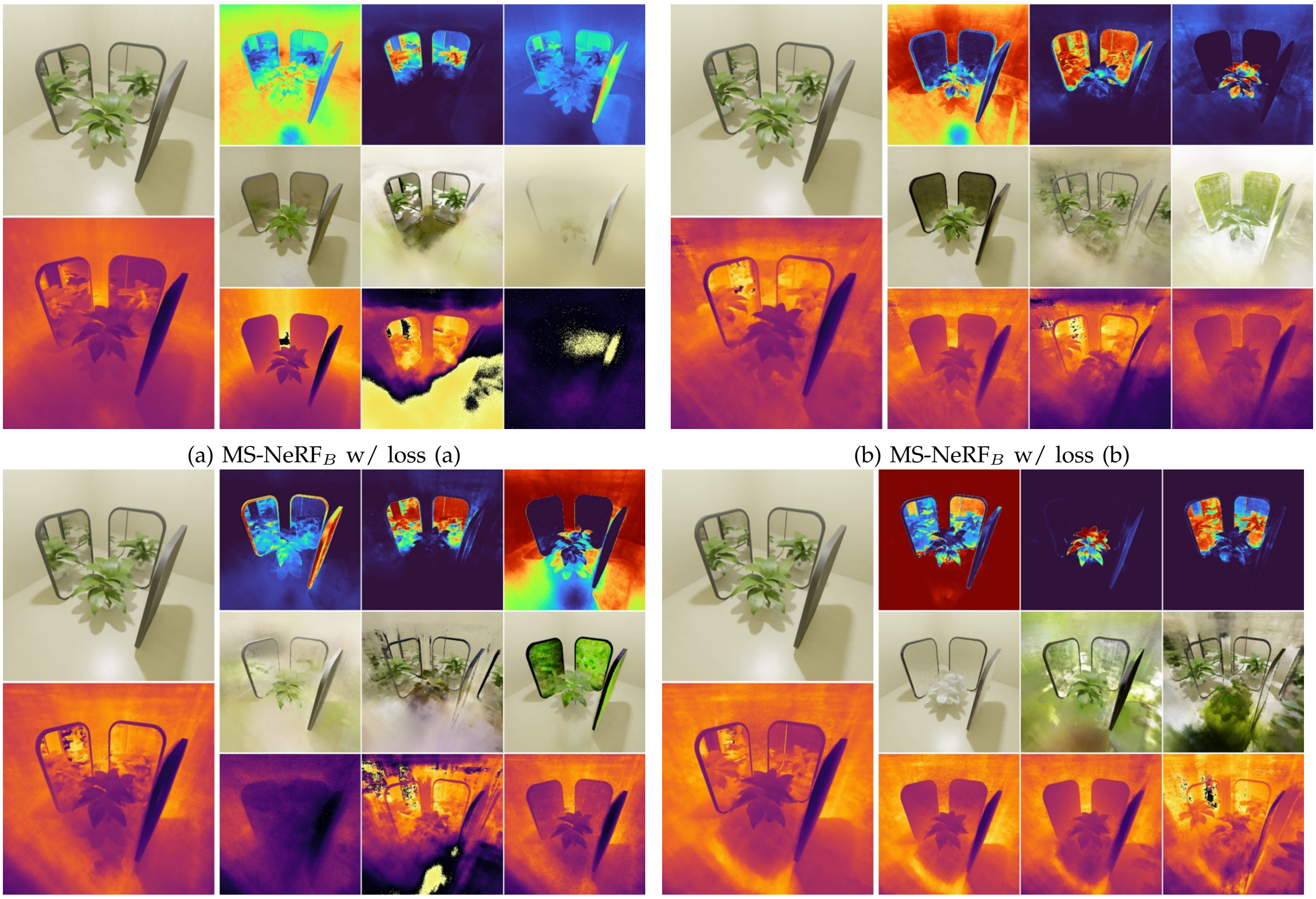
(b) Visual comparison of rendered ray length maps. (i) GT; (ii) MS-NeRF-depth; (iii) NeRF-depth; (iv) NeRF-depth-v.

(i) Full render result. (ii) Weight and ray length of sub-spaces.

(c) Visualization of full rendered results and the decomposition results.

Fig. 22: Qualitative experiments of the multi-space scheme with density fields only.

As in Fig. 22b(iii) and Fig. 22b(iv), both the NeRF-depth and NeRF-depth-v fail to reconstruct the underlying density fields due to the multi-view inconsistency of the ray transportation length caused by reflected rays. On the contrary, our multi-space scheme successfully handles reflected rays as in Fig. 22b(ii), and the visualization in Fig. 22c proves that successful handling is accomplished by automatically separating the virtual images from the real ones. The experiments confirm that our scheme can handle multi-view inconsistency not only in the appearance domain but also in the geometry domain, and the sub-space decomposition follows the principle of separating multi-view inconsistent parts to different sub-spaces where multi-view inconsistency is well-preserved.



(a) MS-NeRF$_B$ w/ loss (a)

(b) MS-NeRF$_B$ w/ loss (b)

(c) MS-NeRF$_B$ w/ loss (c)

(d) MS-NeRF$_B$ w/ loss (d)

Fig. 23: Visualization of the sub-space decomposition from MS-NeRF$_B$ supervised by different losses. The left image contains the final rendered RGB and depth map, and the right one contains weight maps, RGB images, and depth maps of sub-spaces.

	PSNR↑	SSIM↑	LPIPS↓
NeRF w/ loss (b)	22.00/38.88	0.638/0.972	0.109
MS-NeRF$_B$ w/ loss (a)	24.31/39.24	0.737/**0.976**	0.093
MS-NeRF$_B$ w/ loss (b)	**28.09**/39.94	**0.831**/0.974	**0.075**
MS-NeRF$_B$ w/ loss (c)	27.00/**38.97**	0.803/0.971	0.082
MS-NeRF$_B$ w/ loss (d)	26.06/38.35	0.763/0.969	0.096

Table 9: Quantitative evaluation with different photometric losses. The training loss combinations are (a) $\mathcal{L}_{MAE}$; (b) $\mathcal{L}_{MSE}$; (c) $0.5 \times \mathcal{L}_{MSE} + 0.5 \times \mathcal{L}_{SSIM}$; and (d) $0.9 \times \mathcal{L}_{MSE} + 0.1 \times \mathcal{L}_{LPIPS}$.

7.2 Multi-space scheme with different supervision

To investigate the relation between sub-space decomposition and different supervision losses, we choose the commonly used photometric losses, including Mean Absolute Error (MAE) loss $\mathcal{L}_{MAE}$, Mean Squared Error (MSE) loss $\mathcal{L}_{MSE}$, Structural Similarity Index Measure (SSIM) loss $\mathcal{L}_{SSIM}$, and Learned Perceptual Image Patch Similarity (LPIPS) loss $\mathcal{L}_{LPIPS}$, to supervise the MS-NeRF$_B$ model on Scene01-Scene05 with circle paths from our synthesized dataset.

We provide the quantitative evaluation in Tab. 9, which demonstrates that our multi-space scheme is compatible with different supervision losses. Besides, the visualization of decomposition in Fig. 23 confirms our motivation from Sec. 3.2 that the sub-space decomposition is only determined by whether the images are real or virtual.

8 Conclusion

In this paper, we tackle the long-standing problem of rendering reflective surfaces in NeRF-based methods. We introduce a multi-space NeRF method that decomposes the Euclidean space into multiple virtual sub-spaces. Our proposed MS-NeRF approach achieves significantly better results compared with conventional NeRF-based methods. Moreover, a light-weighted design of the MS module allows our approach to serve as an enhancement to the conventional NeRF-based methods. We also constructed a novel dataset for the evaluation of similar tasks, hopefully helping future research in the community.

References

[1] Z.-X. Yin, J. Qiu, M.-M. Cheng, and B. Ren, “Multi-space neural radiance fields,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 12 407–12 416. 1

[2] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021. 1, 2, 3, 6, 9, 10, 11, 12

[3] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, "Nerf++: Analyzing and improving neural radiance fields," *arXiv:2010.07492*, 2020. 1, 2, 3

[4] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fiecvprlds," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 5470–5479. 1, 2, 3, 5, 9, 10, 11

[5] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 10 318–10 327. 1

[6] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, "Nerfies: Deformable neural radiance fields," in *Int. Conf. Comput. Vis.*, 2021, pp. 5865–5874. 1

[7] E. Tretschk, A. Tewari, V. Golyanik, M. Zollhöfer, C. Lassner, and C. Theobalt, "Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video," in *Int. Conf. Comput. Vis.*, 2021, pp. 12 959–12 970. 1

[8] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 7210–7219. 1

[9] X. Chen, Q. Zhang, X. Li, Y. Chen, Y. Feng, X. Wang, and J. Wang, "Hallucinated neural radiance fields in the wild," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 12 943–12 952. 1

[10] J. Zhang, G. Yang, S. Tulsiani, and D. Ramanan, "Ners: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29 835–29 847, 2021. 1

[11] J. Sun, X. Chen, Q. Wang, Z. Li, H. Averbuch-Elor, X. Zhou, and N. Snavely, "Neural 3d reconstruction in the wild," in *ACM SIGGRAPH 2022 Conference Proceedings*, 2022, pp. 1–9. 1

[12] N. Jain, S. Kumar, and L. Van Gool, "Robustifying the multi-scale representation of neural radiance fields," *arXiv:2210.04233*, 2022. 1, 3

[13] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "Barf: Bundle-adjusting neural radiance fields," in *Int. Conf. Comput. Vis.*, 2021, pp. 5741–5751. 1, 3

[14] A. Tewari, J. Thies, B. Mildenhall, P. Srinivasan, E. Tretschk, W. Yifan, C. Lassner, V. Sitzmann, R. Martin-Brualla, S. Lombardi *et al.*, "Advances in neural rendering," *Computer Graphics Forum*, vol. 41, no. 2, pp. 703–735, 2022. 1

[15] W. Bian, Z. Wang, K. Li, J.-W. Bian, and V. A. Prisacariu, "Nope-nerf: Optimising neural radiance field with no pose prior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 4160–4169. 1

[16] Y.-C. Guo, D. Kang, L. Bao, Y. He, and S.-H. Zhang, "Nerfren: Neural radiance fields with reflections," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 18 409–18 418. 1, 2, 3, 8, 9, 10, 11

[17] S. Wizadwongsa, P. Phongthawee, J. Yenphraphai, and S. Suwajanakorn, "Nex: Real-time view synthesis with neural basis expansion," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 8534–8543. 2, 9

[18] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–14, 2019. 2, 9

[19] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in *Int. Conf. Comput. Vis.*, 2021, pp. 5855–5864. 2, 10, 11

[20] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," in *European Conference on Computer Vision (ECCV)*, 2022. 2, 10, 11

[21] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, jul 2022. 2, 5, 6, 7, 10, 11

[22] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 171–27 183, 2021. 2, 13

[23] M. Oechsle, S. Peng, and A. Geiger, "Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction," in *Int. Conf. Comput. Vis.*, 2021, pp. 5589–5599. 2

[24] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, "Volume rendering of neural implicit surfaces," in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 2

[25] Z. Yu, A. Chen, B. Antic, S. P. Peng, A. Bhattacharyya, M. Niemeyer, S. Tang, T. Sattler, and A. Geiger, "Sdfstudio: A unified framework for surface reconstruction," 2022. [Online]. Available: https://github.com/autonomousvision/sdfstudio 2

[26] L. Yariv, P. Hedman, C. Reiser, D. Verbin, P. P. Srinivasan, R. Szeliski, J. T. Barron, and B. Mildenhall, "Bakedsdf: Meshing neural sdfs for real-time view synthesis," *arXiv*, 2023. 2

[27] Z. Li, T. Müller, A. Evans, R. H. Taylor, M. Unberath, M.-Y. Liu, and C.-H. Lin, "Neuralangelo: High-fidelity neural surface reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[28] J. Qiu, P.-T. Jiang, Y. Zhu, Z.-X. Yin, M.-M. Cheng, and B. Ren, "Looking through the glass: Neural surface reconstruction against high specular reflections," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 20 823–20 833. 2

[29] E. R. Chan, C. Z. Lin, M. A. Chan, K. Nagano, B. Pan, S. De Mello, O. Gallo, L. J. Guibas, J. Tremblay, S. Khamis *et al.*, "Efficient geometry-aware 3d generative adversarial networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 16 123–16 133. 2

[30] M. Niemeyer and A. Geiger, "Giraffe: Representing scenes as compositional generative neural feature fields," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 11 453–11 464. 2, 5, 6

[31] A. Jain, B. Mildenhall, J. T. Barron, P. Abbeel, and B. Poole, "Zero-shot text-guided object generation with dream fields," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 867–876. 2

[32] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," *arXiv preprint arXiv:2209.14988*, 2022. 2

[33] Z. Wang, C. Lu, Y. Wang, F. Bao, C. Li, H. Su, and J. Zhu, "Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation," *arXiv preprint arXiv:2305.16213*, 2023. 2

[34] C. Wang, M. Chai, M. He, D. Chen, and J. Liao, "Clip-nerf: Text-and-image driven manipulation of neural radiance fields," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 3835–3844. 2

[35] Y.-J. Yuan, Y.-T. Sun, Y.-K. Lai, Y. Ma, R. Jia, and L. Gao, "Nerf-editing: geometry editing of neural radiance fields," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 18 353–18 364. 2

[36] A. Haque, M. Tancik, A. Efros, A. Holynski, and A. Kanazawa, "Instruct-nerf2nerf: Editing 3d scenes with instructions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 2

[37] Y. Zheng, V. F. Abrevaya, M. C. Bühler, X. Chen, M. J. Black, and O. Hilliges, "Im avatar: Implicit morphable head avatars from videos," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 13 545–13 555. 2

[38] B. Jiang, Y. Hong, H. Bao, and J. Zhang, "Selfrecon: Self reconstruction your digital avatar from monocular video," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 5605–5615. 2

[39] Y. Feng, J. Yang, M. Pollefeys, M. J. Black, and T. Bolkart, "Capturing and animation of body and clothing from monocular video," *arXiv:2210.01868*, 2022. 2

[40] Y. Cao, Y.-P. Cao, K. Han, Y. Shan, and K.-Y. K. Wong, "Dreamavatar: Text-and-shape guided 3d human avatar generation via diffusion models," 2023. 2

[41] B. Mildenhall, P. Hedman, R. Martin-Brualla, P. P. Srinivasan, and J. T. Barron, "Nerf in the dark: High dynamic range view synthesis from noisy raw images," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 16 190–16 199. 2

[42] X. Huang, Q. Zhang, Y. Feng, H. Li, X. Wang, and Q. Wang, "Hdr-nerf: High dynamic range neural radiance fields," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 18 398–18 408. 2

[43] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, "Plenoctrees for real-time rendering of neural radiance fields," in *Int. Conf. Comput. Vis.*, 2021, pp. 5752–5761. 2

[44] S. J. Garbin, M. Kowalski, M. Johnson, J. Shotton, and J. Valentin, "Fastnerf: High-fidelity neural rendering at 200fps," in *Int. Conf. Comput. Vis.*, 2021, pp. 14 346–14 355. 2

[45] C. Reiser, S. Peng, Y. Liao, and A. Geiger, "Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps," in *Int. Conf. Comput. Vis.*, 2021, pp. 14 335–14 345. 2

[46] C. Sun, M. Sun, and H.-T. Chen, "Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 5459–5469. 2

[47] Z. Chen, T. Funkhouser, P. Hedman, and A. Tagliasacchi, "Mobilenerf: Exploiting the polygon rasterization pipeline for efficient

neural field rendering on mobile architectures," *arXiv:2208.00277*, 2022. 2

[48] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan, "Ref-NeRF: Structured view-dependent appearance for neural radiance fields," *CVPR*, 2022. 2, 9, 10, 11, 12

[49] R. Ramamoorthi *et al.*, "Precomputation-based rendering," *Foundations and Trends® in Computer Graphics and Vision*, vol. 3, no. 4, pp. 281–369, 2009. 2

[50] J. Zeng, C. Bao, R. Chen, Z. Dong, G. Zhang, H. Bao, and Z. Cui, "Mirror-nerf: Learning neural radiance fields for mirrors with whitted-style ray tracing," in *Proceedings of the 31th ACM International Conference on Multimedia*, 2023. 2

[51] G.-W. Yang, W.-Y. Zhou, H.-Y. Peng, D. Liang, T.-J. Mu, and S.-M. Hu, "Recursive-nerf: An efficient and dynamically growing nerf," *IEEE Trans. Visual. Comput. Graph.*, 2022. 2

[52] B. Yang, Y. Zhang, Y. Xu, Y. Li, H. Zhou, H. Bao, G. Zhang, and Z. Cui, "Learning object-compositional neural radiance field for editable scene rendering," in *Int. Conf. Comput. Vis.*, 2021, pp. 13 779–13 788. 2

[53] M. Guo, A. Fathi, J. Wu, and T. Funkhouser, "Object-centric neural scene rendering," *arXiv:2012.08503*, 2020. 2

[54] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 4104–4113. 2

[55] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–13, 2017. 3, 9

[56] K. Yücer, A. Sorkine-Hornung, O. Wang, and O. Sorkine-Hornung, "Efficient 3d object segmentation from densely sampled light fields with applications to 3d reconstruction," *ACM Trans. Graph.*, vol. 35, no. 3, pp. 1–15, 2016. 3

[57] M. Bemana, K. Myszkowski, J. Revall Frisvad, H.-P. Seidel, and T. Ritschel, "Eikonal fields for refractive novel-view synthesis," in *ACM SIGGRAPH 2022 Conference Proceedings*, 2022, pp. 1–9. 3, 8, 9

[58] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and H. Aanæs, "Large scale multi-view stereopsis evaluation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 406–413. 3, 9

[59] Y. Yao, Z. Luo, S. Li, J. Zhang, Y. Ren, L. Zhou, T. Fang, and L. Quan, "Blendedmvs: A large-scale dataset for generalized multi-view stereo networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 1790–1799. 3, 9

[60] N. Max, "Optical models for direct volume rendering," *IEEE Trans. Visual. Comput. Graph.*, vol. 1, no. 2, pp. 99–108, 1995. 3

[61] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2022. [Online]. Available: http://www.blender.org 3, 8

[62] R. Li, H. Gao, M. Tancik, and A. Kanazawa, "Nerfacc: Efficient sampling accelerates nerfs." *arXiv preprint arXiv:2305.04966*, 2023. 7, 11

[63] P. Hedman, P. P. Srinivasan, B. Mildenhall, J. T. Barron, and P. Debevec, "Baking neural radiance fields for real-time view synthesis," in *Int. Conf. Comput. Vis.*, 2021, pp. 5875–5884. 9

[64] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004. 10

[65] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 586–595. 10

[66] Y.-C. Guo, "Instant neural surface reconstruction," 2022, https://github.com/bennyguo/instant-nsr-pl. 14

Ze-Xin Yin is currently a PhD student at the College of Computer Science at Nankai University. He has finished his Master's project under the supervision of Prof. Ming-Ming Cheng and Asst. Prof. Bo Ren. His research interests mainly focus on neural radiance fields and 3D computer vision.

Peng-Yi Jiao received his bachelor's degree from Beijing Institute of Technology in 2019. He is currently pursuing the master's degree with the College of Computer Science, Nankai University. His research interests include neural radiance fields, computer graphics, and computer vision.

Jia-Xiong Qiu is a Ph.D. student from the College of Computer Science at Nankai University. He received his master degree supervised at University of Electronic Science and Technology of China in 2020. He obtained his bachelor degree at Dalian Maritime University in 2017. His research interests include computer vision, computer graphics, robotics, and deep learning.

Ming-Ming Cheng received his PhD degree from Tsinghua University in 2012. He is currently a research fellow in The University of Oxford, working with Prof. Philip Torr. His research interests includes computer graphics, computer vision, image processing, and image retrieval. He has received the Google PhD fellowship award, the IBM PhD fellowship award, and the "New PhD Researcher Award" from Chinese Ministry of Education.

Bo Ren received his B.S. and Ph.D. degrees from Tsinghua University in 2010 and 2015 respectively. He is currently an associate professor at College of Computer Science, Nankai University. His research interests lies in computer graphics, computer vision and artificial intelligence. Current researches involve learning-based/physically-based simulation, 3D scene geometry reconstruction and analysis.