

Single-view Neural Radiance Fields with Depth Teacher

Yurui Chen, Chun Gu, Feihu Zhang, Li Zhang

Abstract—Neural Radiance Fields (NeRF) have been proposed for photorealistic novel view rendering. However, it requires many different views of one scene for training. Moreover, it has poor generalizations to new scenes and requires retraining or fine-tuning on each scene. In this paper, we develop a new NeRF model for novel view synthesis using only a single image as input. We propose to combine the (coarse) planar rendering and the (fine) volume rendering to achieve higher rendering quality and better generalizations. We also design a depth teacher net that predicts dense pseudo depth maps to supervise the joint rendering mechanism and boost the learning of consistent 3D geometry. We evaluate our method on three challenging datasets. It outperforms state-of-the-art single-view NeRFs by achieving 5~20% improvements in PSNR and reducing 20~50% of the errors in the depth rendering. It also shows excellent generalization abilities to unseen data without the need to fine-tune on each new scene.

Index Terms—single-view, novel view synthesis, multi-plane images, neural radiance field, volume rendering.

I. INTRODUCTION

THE method Neural Radiance Fields (NeRF) [1] is proposed for photorealistic novel view synthesis. Given many views of the scene, it creates implicit multi-view geometry and learns for view synthesis. However, it has poor generalizations to new scenes and requires retraining or fine-tuning on each scene.

Recent work [2], [3] has explored the ways of using a single image to train NeRF. They introduce a convolutional feature encoder to learn the image representation which gives it some limited generalization abilities to unseen scenes. But, without fine-tuning, these methods produce many floats and artifacts in rendering novel views.

Multi-Plane Images (MPI) representation that learns multiple RGB images from a single image is also used in [4]–[6] for novel view synthesis. However, MPI heavily relies on the qualities of the planar images and needs plenty of image planes to avoid blurs. There is no strong 3D geometry constraint and it fails in many complex scenes.

MINE [7] introduces the volume rendering of NeRF into the MPI. It runs faster and produces better depth rendering quality compared with single-view NeRFs [2], [3]. However, the rendering quality heavily relies on the number of image planes. It needs high-resolution 4D volumes to store the 4-channel (RGB and volume density) image planes that cost a large amount of GPU memory in both training and prediction.

Li Zhang is the corresponding author. (e-mail:lizhangfd@fudan.edu.cn). Yurui Chen, Chun Gu and Li Zhang are with the School of Data Science, Fudan University. Feihu Zhang is with University of Oxford.

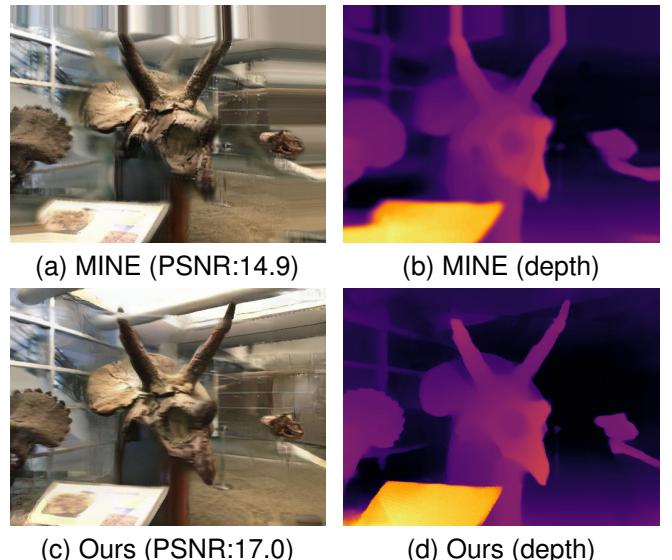


Fig. 1. Comparison with state-of-the-art methods. (a-b) RGB and depth rendering results of [7]. It produces many blurs and floats in the occluded regions and at the object/depth edges. (c-d) Our method employs a joint rendering mechanism that preserves more image details and predicts sharp depth edges.

In this paper, we propose a joint rendering mechanism that takes the MPI strategy for coarse sampling proposals and the MLP&volume-based rendering [1] for fine sampling and rendering. Then, both the coarse point samples and the fine samples are combined according to their geometry distribution to realize a more accurate joint rendering. More importantly, we introduce a depth teacher net that serves as the guidance for the joint rendering. The monocular depth teacher predicts dense pseudo depth maps that assist the consistent 3D geometry learning between the MPI, the fine volume, and the joint rendering. It also boosts the multi-view geometry consistency between the source view and the target novel views that helps handle the occlusions, reduce the blurs and floats, and render accurate depths.

In the experiments, we verify the effectiveness of our method on three challenging real-scene datasets (RealEstate10K [8], NYU [9] and NeRF-LLFF [1]) for novel view synthesis or depth estimation. Given a single image as input, our method is shown able to produce higher qualities in both the RGB image rendering and depth map prediction. It far outperforms state-of-the-art methods [2], [7] with improvements of 5~20% in PSNR and SSIM for the RGB rendering and reduces 20~50% of the errors for the depth prediction.

II. RELATED WORK

Neural radiance fields (NeRF) [1] have been proposed for photorealistic novel view rendering. However, the original NeRF requires many different views of one scene for training and cannot be used for single-view-based novel view synthesis.

a) Single-view NeRF: Recently, different single-view-based NeRF technologies have been proposed for novel view rendering. PixelNeRF [2] introduces NeRF for novel view rendering using one or a few images as input. It uses a CNN encoder to learn the image representation before the volume rendering. The CNN encoder gives some limited generalization abilities to unseen images. GRF [3] learns local features for each pixel and projects the features for novel view rendering.

MINE [7] introduces the neural radiance fields to the Multi-Plane Images (MPI) synthesis. It learns to predict a 4-channel image (RGB and volume density) at arbitrary depth values for contiguous novel-view depth and RGB synthesis. Similarly, AdaMPI [10] proposes to learn adaptive multi-plane images for single-view-based novel view synthesis in the wild.

Besides, some methods focus on shape or object information for novel view rendering. AutoRF [11] learns 3D object radiance fields from single-view observations. CodeNeRF [12] learns separated embedding to disentangle shape and texture. Sharf [13] uses shape information as guidance for learning NeRF from a single image.

There are also some other strategies introduced for single-view NeRF rendering. For example, PVSeRF [14] proposes a joint pixel-, voxel- and surface-aligned NeRF. Pix2NeRF [15] introduces pi-GAN [16] to NeRF for learning novel view synthesis using a single image as input.

Our method is a little similar to SinNeRF [17]. It constructs geometry pseudo labels and semantic pseudo labels to guide the NeRF training. However, SinNeRF has poor generalization abilities to other unseen scenes. It requires retraining or fine-tuning on each scene to achieve high-quality rendering. Different from SinNeRF, our method utilizes a monocular depth teacher network that can generate dense depth maps for both the source view and the target novel views. It does not need semantic information to further regulate the training. Moreover, it has great generalization abilities to new scenes. Namely, it can be easily applied to other unseen images that are not included in the training set.

b) Novel View Synthesis from Single Image: Multi-Plane Images (MPI) representation learned from a single image are used in [4]–[6] to represent the 3D scenes and synthesize novel views. Unsupervised manners are proposed for novel view synthesis in [18], [19]. Different transformer designs are introduced in [11], [20], [21] for single-view-based novel view synthesis. 3D information, such as depth [22], [23], mesh [24] or point cloud [25] information are also effectively employed for novel view synthesis in [22]–[24]. Besides, Liu et al. [26] propose long-range novel views generation. PixelSynth [27] fuses 3D reasoning with autoregressive modeling for scene synthesis. Srinivasan et al. [28] learn to synthesize an RGBD light field from a single image.

III. PRELIMINARY

a) Volume Rendering: NeRF [1] represents a scene as a continuous radiance field. It takes the 3D position $\mathbf{x}_i \in \mathbb{R}^3$ and the viewing direction as input and outputs the corresponding color \mathbf{c}_i with its differential density σ_i . NeRFs use volume rendering to render image pixels. For each 3D point x_i in the space, its color can be rendered through the camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ with N stratified sampled bins between the near and far bounds of the distance.

$$\hat{\mathbf{I}}(\mathbf{r}) = \sum_{i=1}^N T_i(1 - \exp(-\sigma_i \delta_i)) \mathbf{c}_i, \quad T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right). \quad (1)$$

Where \mathbf{o} is the origin of the ray, T_i is the accumulated transmittance along the ray, \mathbf{c}_i and σ_i are the corresponding color and density at the sampled point t_i . $\delta_j = t_{j+1} - t_j$ refers to the distance between the adjacent point samples.

Similarly, the depth map can be rendered as:

$$\hat{D} = \sum_{i=1}^N T_i(1 - \exp(-\sigma_i \delta_i)) z_i. \quad (2)$$

b) Planar Neural Radiance Field: Planar Neural Radiance Field, introduced by [7], is a perspective geometry for representing the camera frustum. For any pixel located on image coordinate (x, y) , it represents the 3D location with candidate depth z as (x, y, z) . Then, it learns D multi-plane images (in a shape of $depth(D) \times height(H) \times width(W) \times 4$) with (\mathbf{c}_i, σ_i) at each 3D location (x, y, z) to represent the image.

For novel view rendering, the intersection locations between the camera ray and each plane are sampled to achieve \mathbf{c}_i and σ_i for image rendering using Eq. (1). The number of the sampled point t_i equals the number of planes.

The planar rendering is faster than volume rendering and has better generalization abilities to unseen images. However, it's costly in terms of memory consumption since it requires storing a 4D volume for plane sampling. Also, the rendering quality heavily relies on the number of image planes and is thus not suitable for fine rendering.

IV. METHOD

In this section, we describe our DT-NeRF for single-view-based novel view synthesis. As illustrated in Figure 2, our method consists of two major parts: 1) the depth teacher net, that is based on the monocular depth estimation and predicts pseudo dense depth to supervise the student net rendering. 2) the student rendering net. It consists of a coarse planar rendering decoder and a fine volume rendering decoder. We use the depth teacher to supervise the joint rendering and boost the geometry consistency between the two renderings.

A. Feature Encoder

In order to generalize to new scenes without retraining or fine-tuning, we introduce a feature extractor to encode the input image. This allows the network to be trained across multiple scenes to learn a scene prior to generalizations to unseen data. The whole output image feature tensor is then used as the input of the coarse planar decoder to learn 4-channel (RGB- σ) MPI, while the sampled point features are fed to the fine volume decoder for fine volume rendering.

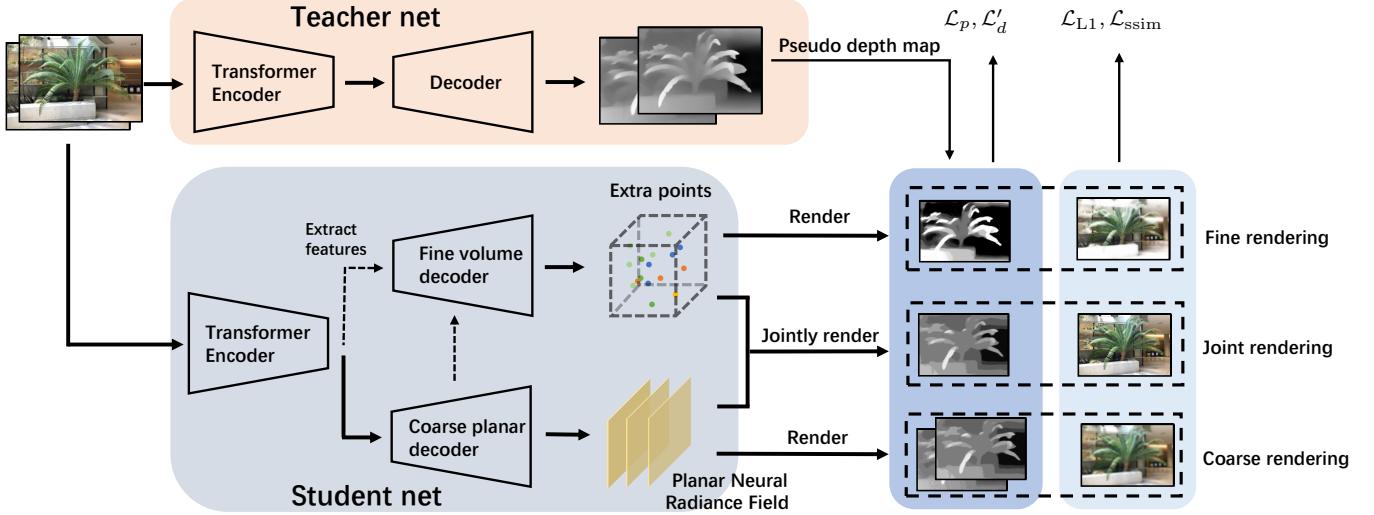


Fig. 2. Our model contains two parts: the teacher net and the student net. Student net is responsible for novel view synthesis task. It takes a single RGB image as input and outputs a coarse planar radiance field which is later refined by the extra points predicted by the fine decoder. We then combine coarse sampling and fine sampling to jointly render the high-quality novel views and depth maps. The teacher net aims at supervising the student net on depth estimation and boosting geometry consistency.

We use a vision transformer [29] as our feature encoder. The vision transformer contains a ResNet50 backbone and 12 transformer blocks. In order to reduce the size of the model, the transformer encoder can be shared by the teacher net and the student net.

B. Depth Teacher

Previous methods [5], [7] are basically supervised by color information (and some point clouds), their depth estimation is not always reliable, resulting in blurred boundaries or unnatural holes. To generate a more reasonable depth estimation and boost the 3D geometry consistency between the novel views and the input source image, we add a depth teacher net \mathcal{G} to generate a dense pseudo depth map from a single input RGB image. The dense depth maps of the source view and target view generated from \mathcal{G} are used as pseudo labels to supervise the student net for both RGB and depth rendering.

Our teacher net is adapted from [29] which consists of a transformer encoder and a convolutional depth prediction decoder. The dense depth map generated by the teacher net builds strong alignment between RGB labels and depth pseudo labels. It contains more 3D prior knowledge than the previous method [5], [7], and boosts the student net to learn more consistent geometries.

C. Student Joint Rendering

The teacher net outputs pseudo dense depth maps to supervise our student joint rendering networks. The student net consists of two components: 1) the coarse planar NeRF that learns coarse planar sampling and rendering as guidance to resample 2) the fine volume rendering. Finally both the samples are employed for joint rendering with Eq. (1) (illustrated in Figure 3). This is different from the coarse and fine sampling of the original NeRF [1] where only fine samples are used for final RGB and depth rendering.

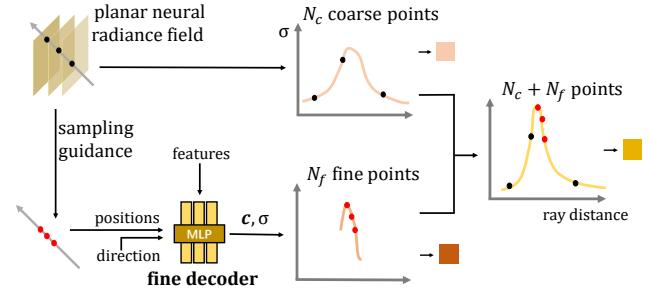


Fig. 3. Joint sampling and rendering consists of the coarse planar rendering (top) and the fine volume rendering (bottom). For rays emitted from the target view camera, the model can roughly estimate the probability density function of the weight T_i from N_c multi-plane images. Then, additional N_f point samples are selected by the importance sampling. The corresponding point features are extracted from the output of the feature encoder, and then input to the fine decoder together with positions and directions to obtain the RGB- σ . Finally, $N_c + N_f$ points are combined for a joint rendering.

a) *Coarse Planar Neural Radiance Field:* Our planar neural radiance field network consists of an encoder and a decoder. The feature encoder takes a source view image as input and outputs a feature tensor. The feature tensor is input to the coarse planar decoder and predicts a 4-channel image (c_i, σ_i) at each candidate depth z_i (total N_c candidate depth values).

Similar to [7], for each pixel (x_t, y_t) in the target novel views, we warp it to the source multi-plane images according to the camera rays.

We define the homography transformation $\mathcal{W}(\cdot)$ between the pixel coordinates of the target novel view (x_t, y_t) and the multi-plane images of the source view (x_i, y_i, z_i) . For simplicity, we use $(x_i, y_i, z_i) = \mathcal{W}_i(x_t, y_t)$ to denote the mapping to a plane with depth of z_i at the source camera (More details can be found in [7] or the supplementary). N_c coarse samples (c_i, σ_i) are selected at the warped location (x_i, y_i, z_i) in N_c image

planes. We then use Eq. (1) to render the coarse images.

b) *Fine Volume Rendering*: The planar rendering runs faster and has a better generalization to unseen data. However, it costs a large amount of memory to store multi-plane images and render high-resolution novel views. It's necessary to use a small N_c to reduce the memory costs. However, insufficient sampling from the image planes (a small N_c) usually leads to blurry novel views. Also, the coarse planar rendering doesn't take the ray direction into account and can't model complex view-dependent effects (e.g. lighting variations).

We build a fine volume MLP decoder to boost the rendering and select more important RGB- σ values of interested points in the space. The new importance sampling is guided by the coarse planar sampling results. Since the volume rendering of Eq. (1) can be interpreted as a weighted sum of all sampled colors c_i , we compute the weights of colors T_i after the rendering with the coarse planar neural field for each ray. We can then obtain a probability density function (PDF) estimation of the weight along the ray by normalizing these weights $\hat{T}_i = T_i / \sum_j T_j$. Then, we sample a new set of (a total of N_f) positions $\mathbf{x}_i = (x_i, y_i, z_i)$ from this distribution using inverse transform sampling [1].

According to the sampled 3D positions, we extract features from multi-plane images. Moreover, we project the 3D spatial coordinate \mathbf{x} to the source image plane using the projection matrix $\mathbf{P} \sim [\mathbf{R}, \mathbf{T}, \mathbf{K}]$ ($(x_s, y_s) = \mathbf{Px}$, where \mathbf{R}, \mathbf{T} are the camera rotation and translation from the target view to the source view and \mathbf{K} is the camera intrinsic parameters).

Given an input feature tensor (output by the feature extractor) \mathbf{F}_{src} , we use the projected pixel coordinates (x_s, y_s) to extract the new feature samples \mathbf{f} . Then, the sampled feature vectors \mathbf{f}_i together with the position and view direction are fed to the fine decoder network (implemented by MLP layers) and output the color \mathbf{c} and density σ . This is similar to the original NeRF [1].

c) *Joint Sampling and Rendering*: We combine the N_c coarse samples and the N_f fine samples by placing them correctly along the camera rays (as illustrated in Figure 3). The joint rendering takes the two types of \mathbf{c} and σ as inputs to the rendering function of Eq. (1) and outputs the better RGB and depth rendering results.

Since the volume rendering and the coarse planar rendering are not in the same space. To correctly place the coarse and fine point samples along the camera rays (as illustrated in Figure 3). It is important to preserve a consistent 3D geometry between the coarse MPI and the fine volume. We propose to use the dense pseudo depth maps predicted by our depth teacher to supervise the joint rendering and make them consistent with each other.

V. IMPLEMENTATION

In order to accelerate the convergence in training, we divide our training into two stages. We first fine-tune the teacher net, then, train our student net with the depth teacher fixed.

A. Depth Teacher Pre-training

Since the depth map predicted by the teacher net is not completely accurate and there may be problems with scale, we

pre-train/fine-tune the teacher net using the available sparse point cloud before training student net. Following [30], the pre-training is supervised by the projection color errors, the gradient consistency and the sparse 3d points (achieved by SfM algorithm [31]).

$$\mathcal{L}^G = \mathcal{L}_{proj} + \mathcal{L}_{smooth} + \mathcal{L}_p, \quad (3)$$

where the reprojection error \mathcal{L}_{proj} and the edge-aware smoothness loss \mathcal{L}_{smooth} are the same as those in [30], and \mathcal{L}_p denotes the point cloud loss (Eq. (7)).

B. Learning Joint Rendering

Our student net mainly has three parts: 1) the “coarse” view rendered by planar neural field, 2) the “fine” view rendered by fine volume rendering with only importance sampling points, and 3) the “joint” rendering results that combined both the coarse samples and the fine samples. Each part will output the corresponding RGB image $\hat{\mathbf{I}}$ and depth map estimation $\hat{\mathcal{D}}$ respectively. The joint rendering is supervised by

$$\mathcal{L} = \mathcal{L}_c + 0.4\mathcal{L}_f + \mathcal{L}_j. \quad (4)$$

Where $\{c, f, j\}$ represents the “coarse”, “fine” and “joint” rendering. Each part has the same loss that consists of $L1$ loss \mathcal{L}_{L1} , $SSIM$ loss [32] \mathcal{L}_{ssim} , sparse point cloud loss \mathcal{L}_p (if available) and pseudo depth loss \mathcal{L}'_d :

$$\mathcal{L}_{c/f/j} = \mathcal{L}_{L1} + \lambda_{ssim}\mathcal{L}_{ssim} + \lambda_p\mathcal{L}_p + \lambda'_d\mathcal{L}'_d. \quad (5)$$

$$\mathcal{L}_{L1} = |\hat{\mathbf{I}} - \mathbf{I}|, \mathcal{L}_{ssim} = 1 - SSIM(\hat{\mathbf{I}}, \mathbf{I}) \quad (6)$$

The RGB $L1$ loss and $SSIM$ loss make the target novel view $\hat{\mathbf{I}}$ generated by our model match the ground truth image \mathbf{I} .

If there are some sparse point clouds \mathbf{P} available (usually generated by SfM method [31]), the sparse point cloud loss can be used as

$$\mathcal{L}_p = \frac{1}{|\mathbf{P}|} \sum_{(x,y,z) \in \mathbf{P}} \left(\ln \frac{\hat{\mathcal{D}}(x,y)}{s} - \ln \frac{1}{z} \right), \quad (7)$$

where, $\hat{\mathcal{D}}$ represents the rendered disparity (inverse depth) map and s represents the scale factor relative to the point clouds set \mathbf{P} . The point cloud loss is applied for both the source view and the target novel view rendering.

Since the scale s between the disparity map generated by depth teacher network, student net and the point cloud are usually different, it is necessary to scale them to a uniform scale before supervising training. When point cloud \mathbf{P} is available, following [7], we unify the point cloud scale by

$$s = \exp \left[\frac{1}{|\mathbf{P}|} \sum_{(x,y,z) \in \mathbf{P}} \left(\ln \frac{1}{z} - \ln \hat{\mathcal{D}}(x,y) \right) \right]. \quad (8)$$

For the pseudo depth loss \mathcal{L}'_d , the $L1$ depth loss and the gradient loss are used as

$$\begin{aligned} \mathcal{L}'_d &= |\hat{\mathcal{D}} - \mathcal{D}^*| + \lambda_{grad}(|\partial_x(\hat{\mathcal{D}}) - \partial_x(\mathcal{D}^*)| \\ &+ |\partial_y(\hat{\mathcal{D}}) - \partial_y(\mathcal{D}^*)|). \end{aligned} \quad (9)$$

$\hat{\mathcal{D}}$ and \mathcal{D}^* are the scaled disparity (inverse depth) maps predicted from the student net and the teacher net respectively. ∂_x and ∂_y are gradients of the disparity maps. We apply these two losses in both the source and the target view to boost the geometry consistency between the student rendering net and the depth teacher net.

In practice, for stable training and faster convergence, we train our student net by two steps. We first train the coarse planar neural radiance field, then fix the coarse planar rendering, and set loss to $0.4\mathcal{L}_f + \mathcal{L}_j$ to train fine decoder.

C. Inpainting Refinement

Since it's difficult to render image borders of the novel views when it is out of the field of view of the source image, we implement a light-weighted inpainting module to refine both the RGB and the depth results at the occlusions and image borders. We leverage the predicted depth map of the source view to compute the occlusion mask of the target view by depth warping. We then follow [33] to learn the inpainting, but change to use a four-channel (RGB-D) representation for both the input and the output. Inspired by [10], we use the warp-back strategy to augment the data for training the inpainting module.

D. Training Details

For our experiments, N_c is fixed to 32, N_f is 16, λ_{ssim} , λ_{L1} and λ_p are set to 1. We use the Adam Optimizer [34] with an initial learning rate of 0.001 for both the planar radiance decoder and fine decoder, 1e-5 for transformer encoder, and 1e-4 for depth teacher \mathcal{G} 's decoder. Our fine decoder is a light-weighted MLP module which has 5 hidden layers with 64 channels. Fine decoder processes a 150-dimension feature and output 4-channel (\mathbf{c}, σ) prediction. Before feeding positions into the fine decoder, we apply a positional encoding [1] which maps 3D coordinates into a 63-dimension feature space.

VI. EXPERIMENTAL RESULTS

A. Datasets

We use NeRF-LLFF [1] and Realestate10k [8] datasets for training and validation. Besides, we also use NYUv2 dataset [9] as the test set to evaluate the depth rendering.

a) *NeRF-LLFF*: NeRF-LLFF [1] consists of 8 scenes. Following [7], we select images in each scene as the test set which consists of 35 images, and the rest 270 images are used as the training set. In experiments on NeRF-LLFF, we set $\lambda'_d = 10$, $\lambda_{grad} = 5$. The resolution is set to 512×384 . We fine-tune our depth teacher for 1,000 iterations, train coarse planar rendering for 20,000 iterations and another 10,000 iterations for fine rendering. We use a batch size of 4. The learning rate decays every 8000 steps.

b) *RealEstate10K*: RealEstate10K [8] is a large dataset that consists of more than 70,000 video sequences. Limited by the available computing resource, we randomly choose 1,000 sequences from the pre-split training set to train our model and test it on 600 randomly selected sequences from the test set. For training and testing, we sample the source and target view

TABLE I
PERFORMANCES OF OUR METHOD USING DIFFERENT N_c AND N_f ON NERF-LLFF DATASET.

N_c	N_f	LPIPS↓	SSIM↑	PSNR↑
MINE (32)	-	0.386	0.531	18.20
MINE (64)	-	0.424	0.539	18.13
16	0	0.328	0.548	18.78
16	16	0.396	0.585	19.07
16	32	0.347	0.608	19.17
32	0	0.305	0.612	19.41
32	16	0.317	0.637	19.54
32	32	0.292	0.650	19.57
64	0	0.315	0.626	19.36
64	16	0.293	0.641	19.38
64	32	0.291	0.642	19.39

pairs at a 10-frame interval from the video sequences, which gives us 32,000 training pairs (of source and target views) and 2,400 test pairs. In experiments on Realestate10k dataset, we set $\lambda'_d = 1$, $\lambda_{grad} = 20$. The input resolution is set to 384×256 . We fine-tune our \mathcal{G} 2,000 iterations, train the coarse planar rendering for 10,000 steps with a batch size of 24 and another 20,000 iterations for the fine decoder with a batch size of 8. The learning rate decays at 1000 steps and 8000 steps.

B. Ablation Study

As shown in Table IV, we verify the effectiveness of different rendering components and study the effects of different settings on the LLFF dataset. We find that both our planar rendering and the volume rendering perform better than the baseline PixelNeRF. The joint rendering achieves a better PSNR and SSIM compared with each individual rendering way. It should be noted that additional importance sampling can boost the planar rendering method for preserving more fine image/depth details. Since the number of planes sampled is limited, blur and artifacts are unavoidable. Figure 5 shows how extra fine points solve this problem and lead to a better rendering. And the full setting with the depth teacher supervision achieves the best PSNR, SSIM and LPIPS. As shown in Figure 1, compared with the pure planar rendering [7], the joint rendering predicts more details and fewer blurs in both the RGB and depth rendering. Figure 4 also compares our method with the pure planar NeRF (MINE) and the pure volume rendering PixelNeRF. The joint rendering without our depth teacher achieves better depths in object edges and occlusions. By introducing the depth teacher supervision \mathcal{L}'_d , the rendering quality of the depth maps is significantly improved.

Although teacher net (monocular depth estimation) is able to predict high-quality depth maps on the input image, it can't render depth maps for novel view. The student net renders both the RGB and the depth maps for the novel views which are very important for applications like video editing, augmented reality etc.

a) *Coarse/Fine Point Sampling*: As shown in Table I, we report the performance when using different coarse and fine samples. We find that using more point samples (either

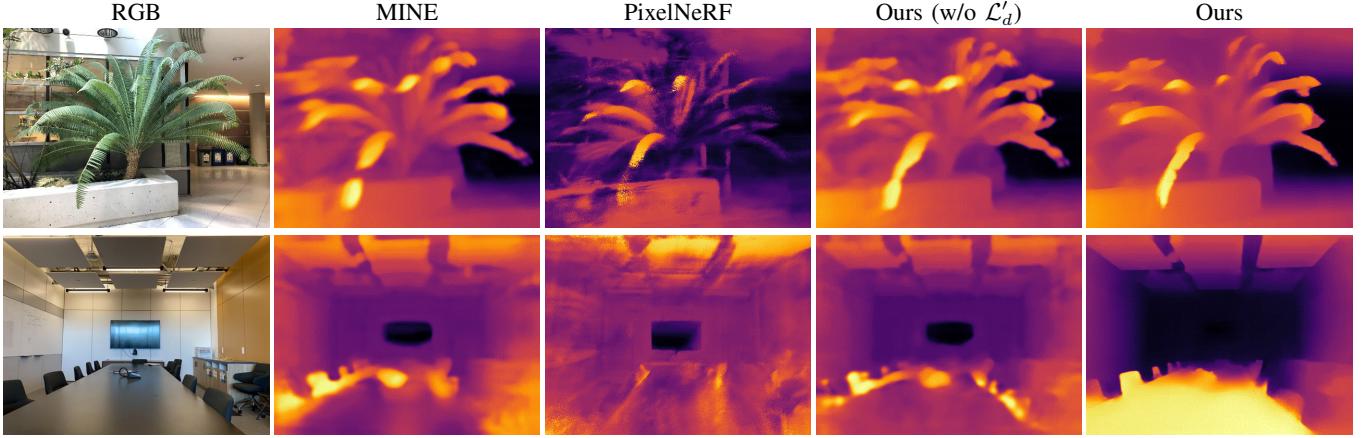


Fig. 4. Effects of the pseudo depth loss. Even without the depth teacher, our method can achieve better depth maps compared with MINE and PixelNeRF. The depth teacher with pseudo depth loss further improves the quality of the depth rendering.

TABLE II

COMPARISON WITH DIFFERENT ENCODER BACKBONES. WE TEST TWO DIFFERENT FEATURE ENCODER BACKBONES, THE DEFAULT TRANSFORMER BACKBONE ViT AND THE RESNET50 (USED IN MINE). WE SET $N_c = 32$, $N_f = 16$. OUR METHOD PERFORMS FAR BETTER THAN MINE WHEN USING THE SAME BACKBONE NETWORK (RESNET50). ViT SLIGHTLY IMPROVES THE PSNR ON LLFF DATASET WHEN COMPARED WITH RESNET50, AND THERE ARE NO SIGNIFICANTLY IMPROVEMENTS ON REALESTATE10K DATASET.

Method	Backbone	Dataset	PSNR↑
MINE	ResNet50	LLFF	18.2
Ours	ResNet50	LLFF	19.3
Ours	ViT	LLFF	19.5
MINE	ResNet50	RealEstate(small)	24.6
Ours	ResNet50	RealEstate(small)	25.0
Ours	ViT	RealEstate(small)	25.0

coarse planar sampling N_c or the fine volume samples N_f) will improve the rendering quality. When using 32 coarse planar samples and 32 fine volume samples, our method performs best. But, using many more planar samples (e.g. $N_c = 64$) will not produce better results.

It should be noted that additional importance sampling can boost the planar rendering method for preserving more fine image/depth details. Since the number of planes sampled is limited, blur and artifacts are unavoidable. Figure 5 shows how extra fine points solve this problem and lead to a better rendering.

b) *Different Backbones*: We test two different feature encoder backbones, the default transformer backbone ViT and the ResNet50 (used in MINE). When using the same ResNet50 backbone, our method achieves 19.3 in PSNR which is 6% improvement compared with MINE under the same setting. ViT slightly improves the PSNR on LLFF dataset when compared with ResNet50, and there is no improvement on RealEstate10K dataset. The improvements are mainly from our joint rendering and depth teacher guidance strategies.

C. View Synthesis on NeRF-LLFF

In table III, we compare our method with PixelNeRF [2] and MINE [7] on the LLFF dataset. We also compare to the

TABLE III

COMPARISON WITH MPI, PIXELNERF AND MINE ON NERF-LLFF DATASET.

Method	LPIPS↓	SSIM↑	PSNR↑
MPI [5]	0.502	0.356	14.6
PixelNeRF [2]	0.476	0.468	17.5
MINE [7]	0.386	0.531	18.2
Ours	0.317	0.637	19.5

TABLE IV

ABLATION STUDY ON NERF-LLFF ON DIFFERENT RENDERING COMPONENTS.

Method	LPIPS↓	SSIM↑	PSNR↑
PixelNeRF [2]	0.476	0.468	17.5
Planar rendering	0.326	0.608	18.9
Volume rendering	0.351	0.628	18.8
Planar+Volume	0.338	0.637	19.0
Full settings	0.317	0.637	19.5

pre-trained MPI [5] on LLFF using the provided checkpoint. Our method achieves the best performance in all the evaluation metrics of PSNR, SSIM and LPIPS. It outperforms the MINE by 20% in SSIM, 7% in PSNR and 18% in LPIPS. As shown in Figure 7, our method is able to render novel views with higher quality (more fine details and fewer blurs). Moreover, the depth rendering results of our DT-NeRF is also far better than MINE (Figure 4).

Our method also far outperforms the PixelNeRF [2]. As shown in Figure 6, we compare our DT-NeRF with PixelNeRF. We only use our volume rendering which just takes 16 samples in rendering the image view. As a comparison, PixelNeRF needs 96 samples. With just 1/6 point samples, our volume rendering can synthesize more realistic images than PixelNeRF. This is because our volume rendering is supervised by the depth teacher and learns better 3D geometry. Moreover, the point sampling is guided by the coarse planar sampling results which assist the more precise sampling around the object surfaces and avoids sampling a large number of useless points.

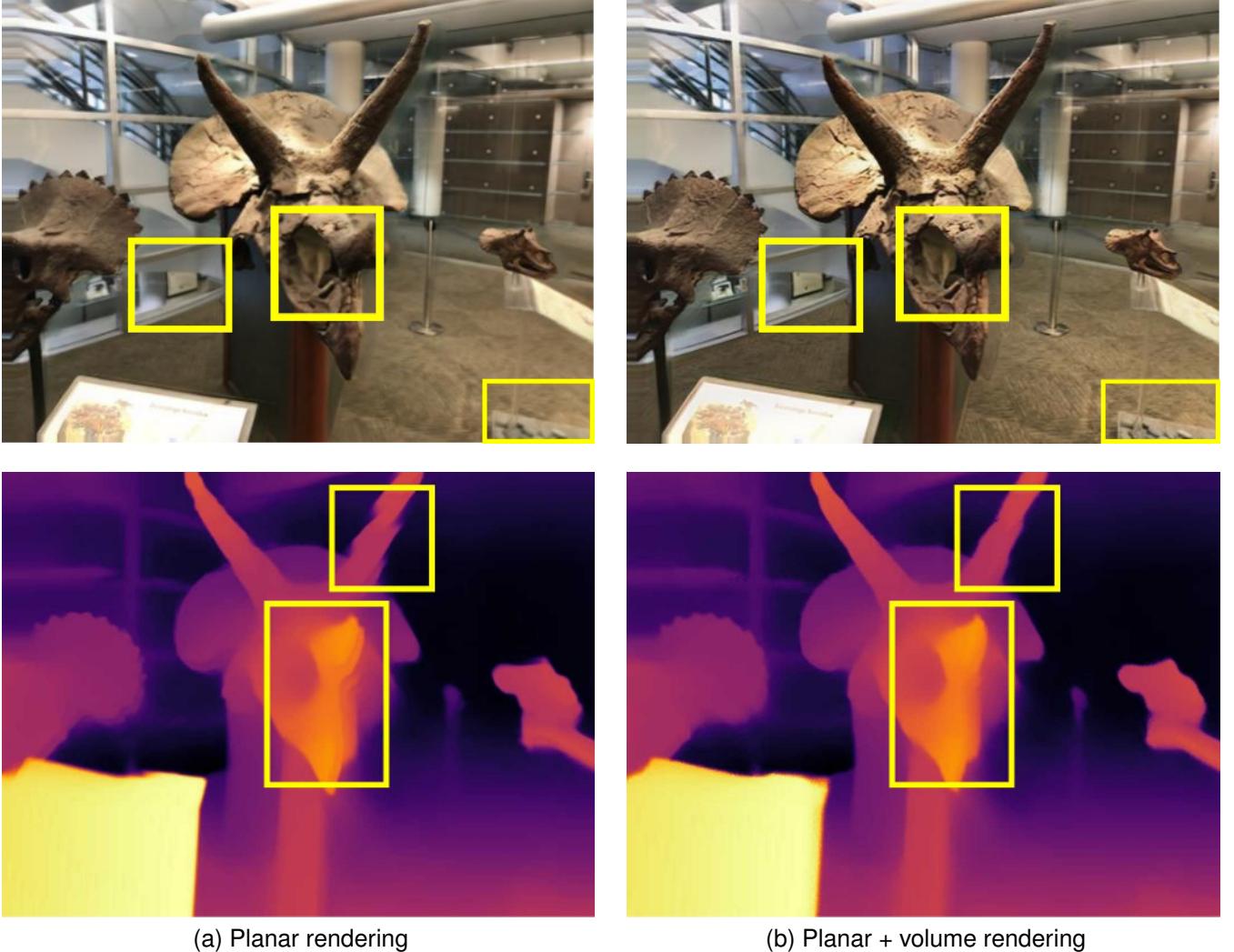


Fig. 5. Influence of fine volume rendering with importance sampling strategy. The improved areas are highlighted by yellow boxes. Compared with the pure planar rendering, extra fine point samples gives more image details and sharper depth edges.



Fig. 6. Comparisons with PixelNeRF using just our volume rendering. With just 16 points selected, our volume rendering are far better than PixelNeRF (96 point samples).

D. View Synthesis on RealEstate10K

We also evaluate our method on the RealEstate10K dataset. It is compared with the state-of-the-art MINE. Our method achieves the best PSNR, SSIM and LPIPS. It outperforms the MPI [5] by 4~10% in these three evaluation metrics, and also surpass the MINE [7] with a better RGB rendering quality. More importantly, compared with MPI and MINE, our DT-NeRF produces far better depth predictions with sharper depth edges and more accurate estimations in occluded regions in

Figure 8.

E. Depth Estimation on NYU-V2

We evaluate depth estimation on NYU-Depth V2 dataset [9]. We perform our method on the labeled subset which consists of 1449 densely labeled pairs of RGB and depth images taken from a variety of indoor scenes. To solve the scale ambiguity problem of predicted depth, following [7], [35], we scale and bias the predicted depth to minimize the L_2 depth error with respect to ground truth.

As shown in Table V, our model performs far better than MINE and MPI in all the evaluation metrics. Even with our model trained on LLFF which consists of only 8 scenes, it still has better depth results than MINE trained on the larger RealEstate10k dataset (1000 scenes and 32,000 training pairs). This is because our DT-NeRF training is guided by the dense pseudo depth maps which help it learn consistent 3D geometry across the source views and the target views in both the planar rendering and the volume rendering. Some results are shown in Figure 9.

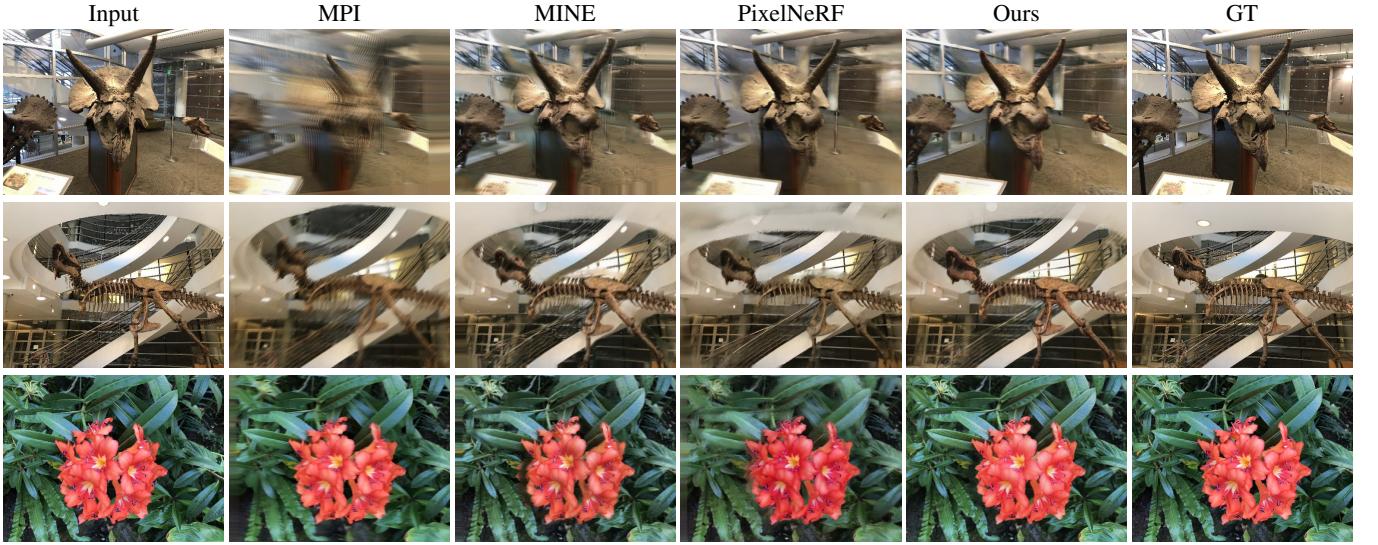


Fig. 7. Visual comparisons on NeRF-LLFF scenes. Models are trained on the LLFF training set and evaluated on the LLFF test set. Compared with MPI, MINE and PixelNeRF, our DT-NeRF produces clear scenes with more details and fewer blurs or floats.

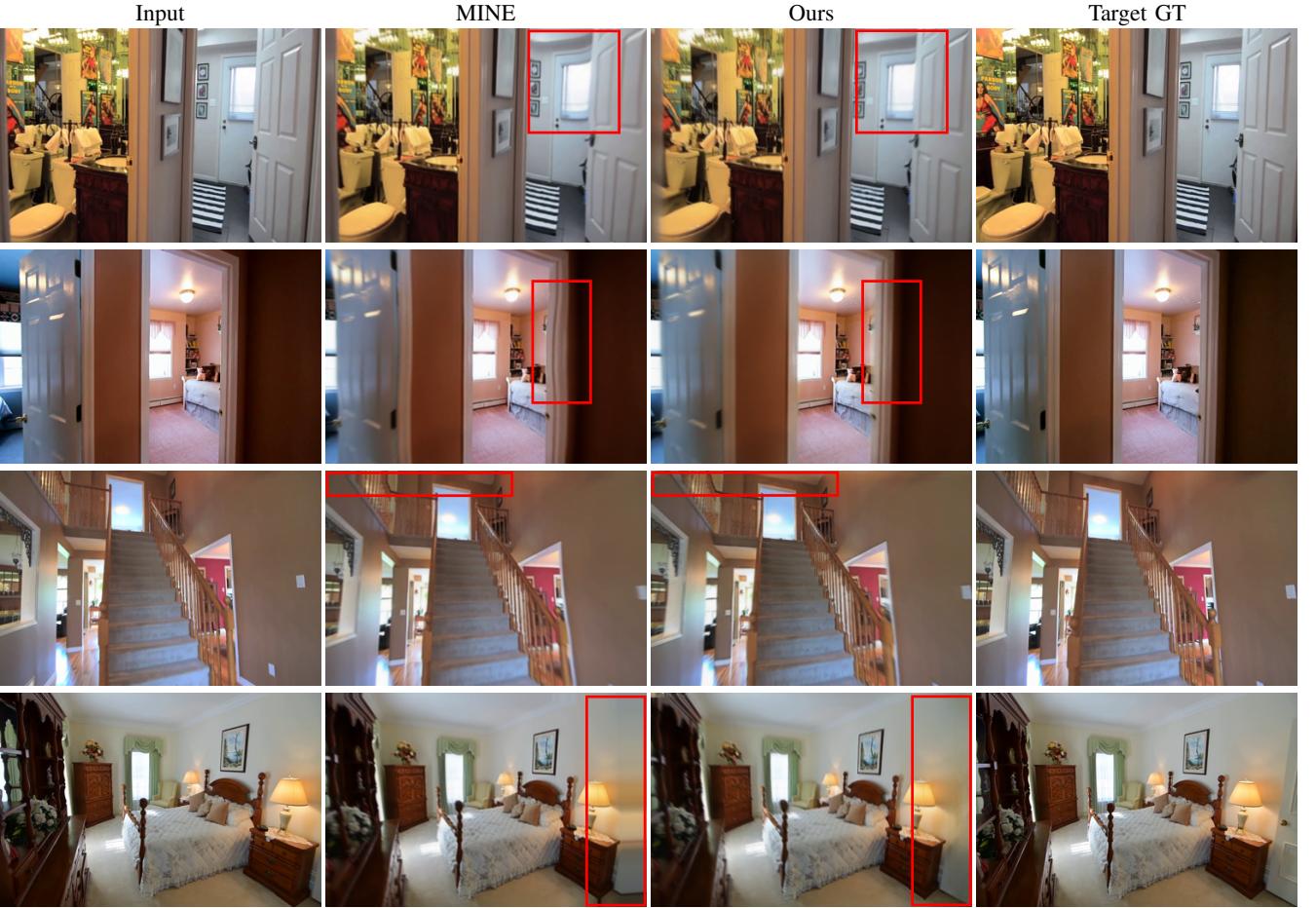


Fig. 8. Visual comparisons on RealEstate10K. Since our depth prediction is more accurate than MINE, our method can avoid many image distortions (as highlighted by the red windows).

F. Analysis on Student and Teacher Net

Table VIII shows \mathcal{L}'_d will improve student net's prediction quality of depth map. And the student net even has stronger generalization performance than fine-tuned teacher net on the

depth prediction. In Figure 10, we show comparisons of the predicted depth maps between the depth teacher and the student net. We use the models trained on RealEstate10K(small) to inference depth maps on the NYUv2 dataset. Student net

TABLE V

DEPTH ESTIMATION RESULTS ON NYU-DEPTH V2. \uparrow DENOTES HIGHER IS BETTER AND \downarrow DENOTES LOWER IS BETTER. BY TRAINING ON LLFF OR A SMALL PART OF REALESTATE10K DATASET, OUR DT-NERF FAR OUTPERFORMS BOTH MPI AND MINE THAT ARE TRAINED ON THE SAME DATASET.

Method	Supervision	Dataset	NYU-Depth V2 [9]					
			rel \downarrow	log10 \downarrow	RMS \downarrow	$\sigma_1\uparrow$	$\sigma_2\uparrow$	$\sigma_3\uparrow$
MPI [5]	RGB	RealEstate10K	0.18	0.07	0.60	0.74	0.94	0.98
MINE [7]	RGB	LLFF	0.32	0.12	0.93	0.51	0.81	0.92
MINE [7]	RGB	RealEstate10K(small)	0.18	0.07	0.58	0.75	0.94	0.98
Ours	RGB	LLFF	0.14	0.06	0.48	0.82	0.97	0.99
Ours	RGB	RealEstate10K(small)	0.12	0.05	0.43	0.86	0.97	0.99

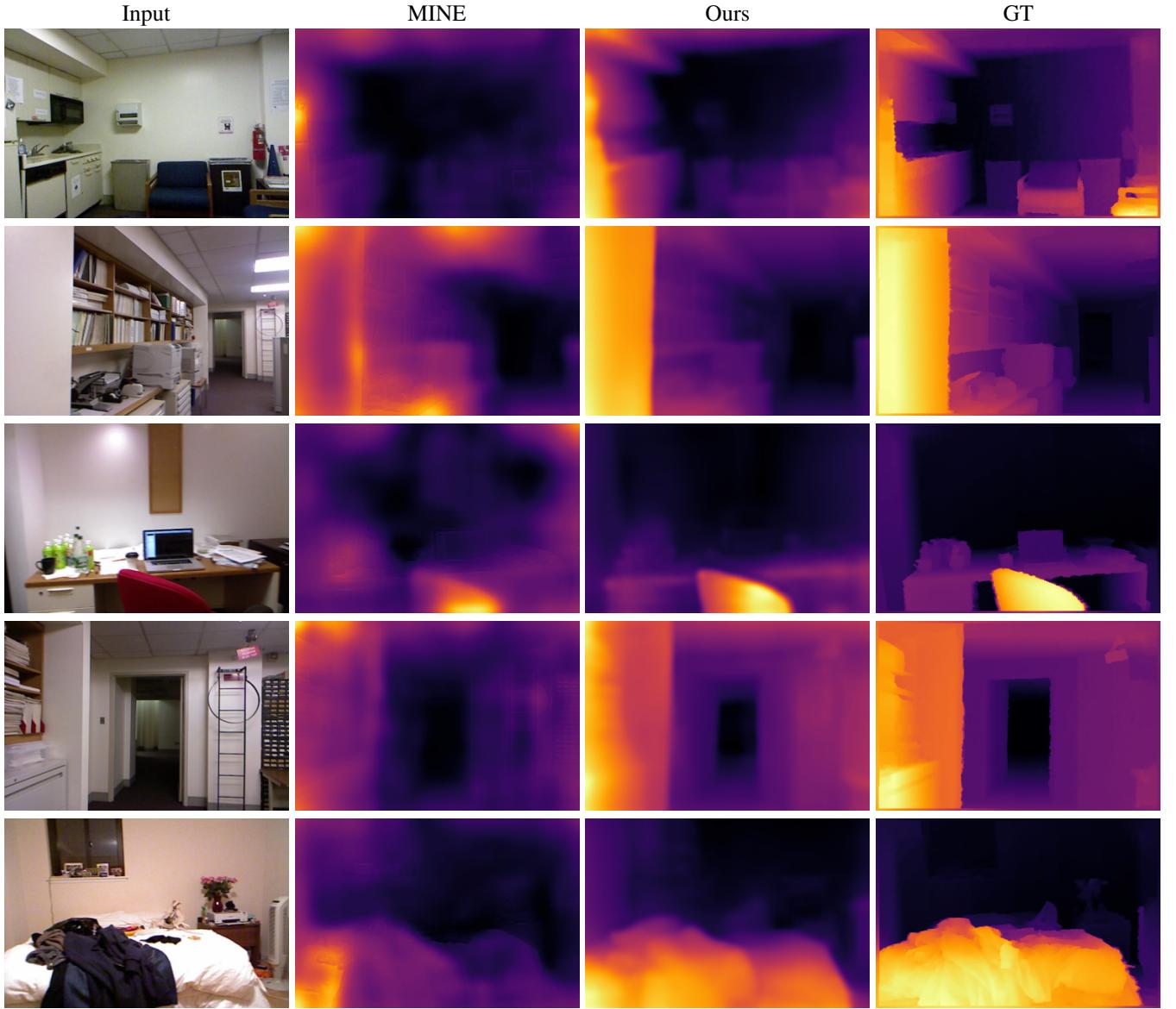


Fig. 9. Qualities of depth maps on NYU-Depth V2 dataset. Models are trained on the RealEstate10K dataset. Our method produces better depth maps with more structure details and sharper depth edges.

performs a little better in four of the six evaluation metrics.

Although teacher net (monocular depth estimation) is able to predict high-quality depth maps on the input image, it can't render depth maps for novel view. The student net renders both the RGB and the depth maps for the novel views which are

very important for applications like video editing, augmented reality etc.

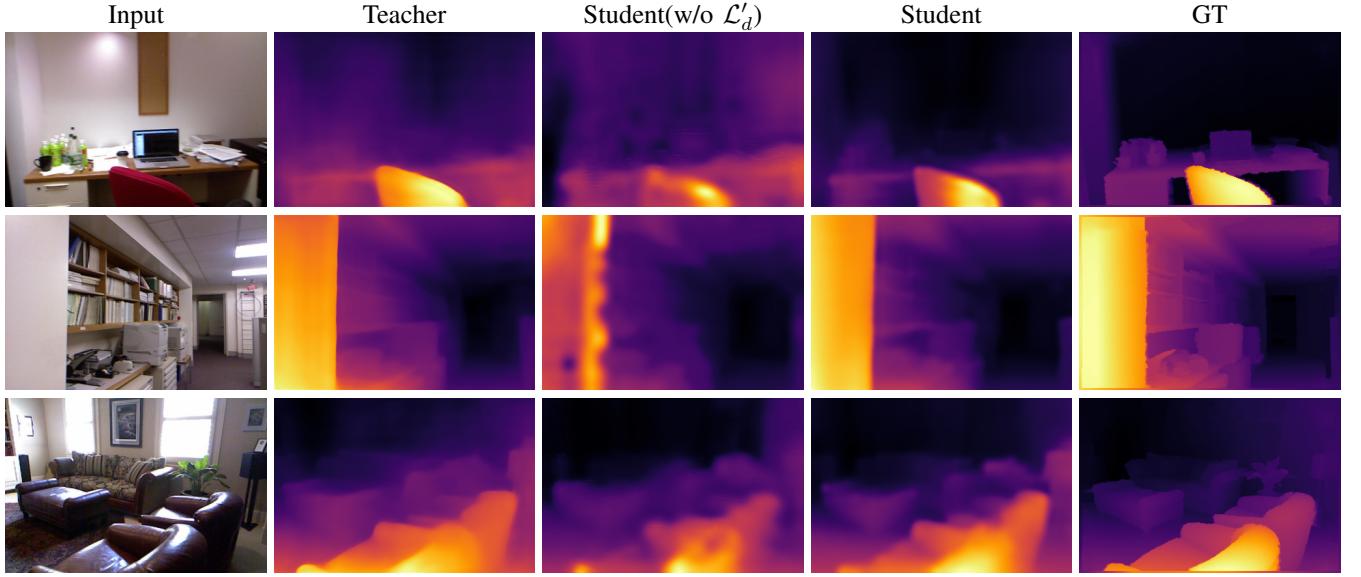


Fig. 10. Comparisons between the teacher net and the student net. Model is trained RealEstate10K. The predicted depth results by the student net and the teacher net are tested on NYU-depth V2 respectively. The depth supervision with pseudo depth loss \mathcal{L}'_d can significantly improve the quality of the student net's prediction.

TABLE VI
EVALUATION ON REALESTATE10K. MODELS ARE TRAINED ON 1000
REALESTATE10K SCENES AND TESTED ON 600 SCENES.

Method	LPIPS↓	SSIM↑	PSNR↑
MPI	0.159	0.793	23.9
MINE	0.146	0.821	24.6
Ours($N_f = 16$)	0.152	0.831	25.0
Ours($N_f = 32$)	0.144	0.833	24.9

TABLE VII

GENERALIZATION ABILITIES ON UNSEEN “ROOM” IN LLFF DATASET. NOTE THAT SINNERF AND PIXELNERF ARE FINE-TUNED ON THE TEST SCENES.

Method	Dataset	LPIPS↓	SSIM↑	PSNR↑
SinNeRF	LLFF-room	0.431	0.642	18.10
PixelNeRF	LLFF	0.419	0.675	18.23
Ours	Realestate10k	0.488	0.613	18.09
Ours	LLFF	0.339	0.745	20.65

G. Generalization

Our DT-NeRF is shown able to generalize to new scenes without fine-tuning or retraining on each of them. This is because we design a depth teacher net to supervise the joint student rendering mechanism and boost the learning of consistent 3D geometry. In this section, we pretrain our DT-NeRF on the RealEstate10K datasets and then compare it with the fine-tuned (on the target NeRF-LLFF scenes) SinNeRF and PixelNeRF on the unseen NeRF-LLFF scenes.

a) *Comparison with Fine-tuned SinNeRF:* Our DT-NeRF shares a similarity with SinNeRF [17]. Both models use a pre-trained teacher net to supervise the training process, while DT-NeRF uses a dense monocular depth teacher, SinNeRF uses a semantic teacher and a geometry teacher. SinNeRF has no generalization abilities to unseen data/images. It needs to

be retrained or fine-tuned on each scene. We train SinNeRF for each scene (512×384) and use the ground truth depth [17] to supervise. Our method has excellent generalization abilities to new scenes. So there is no need to fine-tune it on the test scenes. We directly use our model trained on a small part of RealEstate10K dataset (1000 scenes) and perform novel view synthesis on “room” scene in LLFF dataset. After training on the “room” scene, SinNeRF achieves 18.10 in PSNR. Our unadapted method achieves a similar 18.09 in PSNR.

b) *Comparison with Fine-tuned PixelNeRF:* We also compare our DT-NeRF (trained on unrelated RealEstate10K dataset) with the PixelNeRF [2] that is trained/fine-tuned on the LLFF dataset (including the test room scenes). Even without seeing the LLFF scenes during the training, our method can still produce a similar performance (18.09 in PNSR) that is close to the test-data-fine-tuned PixelNeRF (18.23 in PSNR). After fine-tuning, our DT-NeRF achieves a 14% improvement in PSNR which is far better than the fine-tuned PixelNeRF and SinNeRF.

VII. LIMITATIONS AND FAILURE CASE

Similar to all existing methods for single-image-based novel view synthesis, our method also fails when there are large changes or translations of view angles (e.g. $> 60^\circ$). In this case, almost the whole novel-view image is in the occlusion part. A single-image input doesn't provide enough information for predicting such novel views and thus will produce many artifacts.

VIII. CONCLUSION

In this paper, we have proposed DT-NeRF for photorealistic novel view rendering using only a single view image as input. This is achieved by combining plane rendering and volume rendering for better rendering quality and better generalizations

TABLE VIII
STUDENT/TEACHER NET TRAINED/FINE-TUNED ON REALESTATE10K (SMALL) AND EVALUATED ON NYU-DEPTH V2.

Method	rel \downarrow	log10 \downarrow	RMS \downarrow	$\sigma 1 \uparrow$	$\sigma 2 \uparrow$	$\sigma 3 \uparrow$
Teacher	0.127	0.054	0.452	0.853	0.973	0.993
Student(w/o \mathcal{L}'_d)	0.138	0.058	0.487	0.830	0.964	0.990
Student	0.123	0.052	0.434	0.859	0.972	0.992

to new scenes. We also design an effective depth teacher network that produces dense pseudo depths to supervise the joint rendering and learn consistent 3D geometries. DT-NeRF is shown able to outperform state-of-the-art single-view NeRFs in both the RGB and the depth rendering.

REFERENCES

- [1] 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,” in *ECCV*, 2020.
- [2] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, “pixelnerf: Neural radiance fields from one or few images,” in *CVPR*, 2021.
- [3] A. Trevithick and B. Yang, “Grf: Learning a general radiance field for 3d representation and rendering,” in *ICCV*, 2021.
- [4] Y. Wu, G. Meng, and Q. Chen, “Embedding novel views in a single jpeg image,” in *ICCV*, 2021.
- [5] R. Tucker and N. Snavely, “Single-view view synthesis with multiplane images,” in *CVPR*, 2020.
- [6] Y. Wu, Z. Zou, and Z. Shi, “Remote sensing novel view synthesis with implicit multiplane representations,” *arXiv preprint*, 2022.
- [7] J. Li, Z. Feng, Q. She, H. Ding, C. Wang, and G. H. Lee, “Mine: Towards continuous depth mpi with nerf for novel view synthesis,” in *ICCV*, 2021.
- [8] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, “Stereo magnification: Learning view synthesis using multiplane images,” *arXiv preprint*, 2018.
- [9] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from rgbd images,” in *ECCV*, 2012.
- [10] Y. Han, R. Wang, and J. Yang, “Single-view view synthesis in the wild with learned adaptive multiplane images,” in *SIGGRAPH*, 2022.
- [11] X. Ren and X. Wang, “Look outside the room: Synthesizing a consistent long-term 3d scene video from a single image,” in *CVPR*, 2022.
- [12] W. Jang and L. Agapito, “Codenerf: Disentangled neural radiance fields for object categories,” in *ICCV*, 2021.
- [13] K. Rematas, R. Martin-Brualla, and V. Ferrari, “Sharf: Shape-conditioned radiance fields from a single view,” in *ICML*, 2021.
- [14] X. Yu, J. Tang, Y. Qin, C. Li, L. Bao, X. Han, and S. Cui, “Pvserf: Joint pixel-, voxel- and surface-aligned radiance field for single-image novel view synthesis,” *CoRR*, 2022.
- [15] S. Cai, A. Obukhov, D. Dai, and L. Van Gool, “Pix2nerf: Unsupervised conditional π -gan for single image to neural radiance fields translation,” in *CVPR*, 2022.
- [16] E. R. Chan, M. Monteiro, P. Kellnhofer, J. Wu, and G. Wetzstein, “pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis,” in *CVPR*, 2021.
- [17] D. Xu, Y. Jiang, P. Wang, Z. Fan, H. Shi, and Z. Wang, “Sinnerf: Training neural radiance fields on complex scenes from a single image,” *arXiv preprint*, 2022.
- [18] P. Z. Ramirez, A. Tonioni, and F. Tombari, “Unsupervised novel view synthesis from a single image,” *arXiv preprint*, 2021.
- [19] X. Liu, T. Che, Y. Lu, C. Yang, S. Li, and J. You, “Auto3d: Novel view synthesis through unsupervisedly learned variational viewpoint and global 3d representation,” in *ECCV*, 2020.
- [20] C. Zhang, C. Lin, K. Liao, L. Nie, and Y. Zhao, “Sivsformer: Parallax-aware transformers for single-image-based view synthesis,” in *VR*, 2022.
- [21] R. Rombach, P. Esser, and B. Ommer, “Geometry-free view synthesis: Transformers and no 3d priors,” in *ICCV*, 2021.
- [22] M.-L. Shih, S.-Y. Su, J. Kopf, and J.-B. Huang, “3d photography using context-aware layered depth inpainting,” in *CVPR*, 2020.
- [23] J. Watson, O. M. Aodha, D. Turmukhambetov, G. J. Brostow, and M. Firman, “Learning stereo from single images,” in *ECCV*, 2020.
- [24] R. Hu, N. Ravi, A. C. Berg, and D. Pathak, “Worldsheet: Wrapping the world in a 3d sheet for view synthesis from a single image,” in *ICCV*, 2021.
- [25] O. Wiles, G. Gkioxari, R. Szeliski, and J. Johnson, “Synsin: End-to-end view synthesis from a single image,” in *CVPR*, 2020.
- [26] A. Liu, R. Tucker, V. Jampani, A. Makadia, N. Snavely, and A. Kanazawa, “Infinite nature: Perpetual view generation of natural scenes from a single image,” in *ICCV*, 2021.
- [27] C. Rockwell, D. F. Fouhey, and J. Johnson, “Pixelsynth: Generating a 3d-consistent experience from a single image,” in *ICCV*, 2021.
- [28] P. P. Srinivasan, T. Wang, A. Sreelal, R. Ramamoorthi, and R. Ng, “Learning to synthesize a 4d rgbd light field from a single image,” in *ICCV*, 2017.
- [29] R. Ranftl, A. Bochkovskiy, and V. Koltun, “Vision transformers for dense prediction,” in *ICCV*, 2021.
- [30] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, “Digging into self-supervised monocular depth estimation,” in *ICCV*, 2019.
- [31] J. L. Schönberger and J.-M. Frahm, “Structure-from-motion revisited,” in *CVPR*, 2016.
- [32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *TIP*, 2004.
- [33] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky, “Resolution-robust large mask inpainting with fourier convolutions,” *arXiv preprint*, 2021.
- [34] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint*, 2014.
- [35] S. Niklaus, L. Mai, J. Yang, and F. Liu, “3d ken burns effect from a single image,” *ToG*, 2019.