

Self-NeRF: A Self-Training Pipeline for Few-Shot Neural Radiance Fields

Jiayang Bai
Nanjing University
jybai@smail.nju.edu.cn

Letian Huang
Central South University
lthuangg@gmail.com

Wen Gong
Nanjing University
lesliewinnie@163.com

Jie Guo
Nanjing University
jieguo@nju.edu.cn

Yanwen Guo
Nanjing University
ywguo@nju.edu.cn

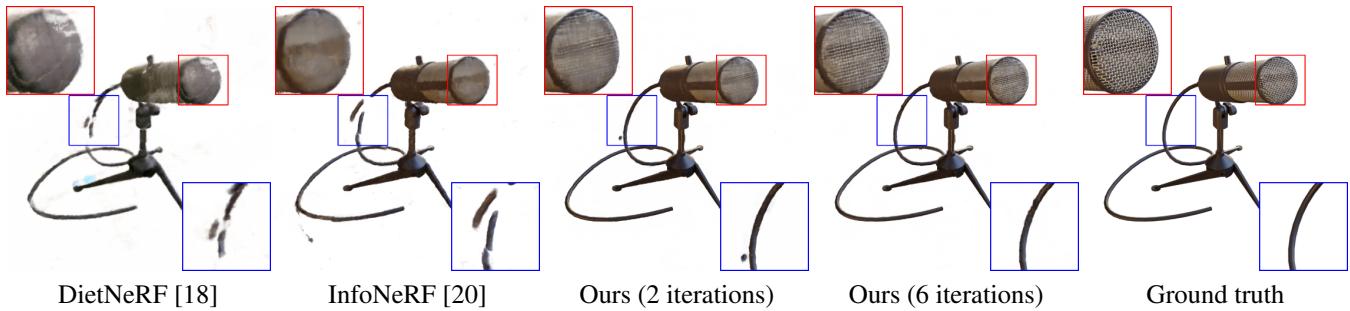


Figure 1: Qualitative comparison of different methods on the mic scene in 4-view settings. Compared with other methods, Self-NeRF yields a more realistic rendering with superior details. Note that Self-NeRF produces sharp details and reduces artifacts through iterative training.

Abstract

Recently, Neural Radiance Fields (NeRF) have emerged as a potent method for synthesizing novel views from a dense set of images. Despite its impressive performance, NeRF is plagued by its necessity for numerous calibrated views and its accuracy diminishes significantly in a few-shot setting. To address this challenge, we propose Self-NeRF, a self-evolved NeRF that iteratively refines the radiance fields with very few number of input views, without incorporating additional priors. Basically, we train our model under the supervision of reference and unseen views simultaneously in an iterative procedure. In each iteration, we label unseen views with the predicted colors or warped pixels generated by the model from the preceding iteration. However, these expanded pseudo-views are afflicted by imprecision in color and warping artifacts, which degrades the performance of NeRF. To alleviate this issue, we construct an uncertainty-aware NeRF with specialized embeddings. Some techniques such as cone entropy regularization are further utilized to leverage the pseudo-views in the most efficient manner. Through experiments under various settings, we verified that our Self-NeRF is robust to input with un-

certainty and surpasses existing methods when trained on limited training data.

1. Introduction

Synthesizing novel camera views given a set of known views is an important task in computer vision and a prerequisite to many AR and VR applications. Classic techniques have addressed this problem using structure-from-motion [16] or light fields [23]. Recently, Neural Radiance Fields (NeRF) [29] have gained tremendous popularity due to the impressive results in photo-realistic rendering. This approach trains learning-based models implicitly embedded within a 3D geometric context and reconstructs observed images using neural rendering techniques. Albeit effective, the performance of NeRF is highly influenced by the quality and the number of training views. When the known views are limited, NeRF collapses to trivial solutions [33] (e.g., producing zero density for the unobserved regions) and has the risk of overfitting to seen views. To make this challenging problem tractable, previous works attempt to incorporate some additional priors, such as a semantic feature [18],

ground truth depth [10] or normalizing flow [32]. Although these models yield adequate rendering performance, these additional priors are not always valid. Kim *et al.* [20] propose a prior-free approach that introduces ray entropy minimization and ray information gain reduction for each ray to alleviate the reconstruction inconsistency and overfitting issue. However, entropy regularization imposes sparsity on the estimated scene, resulting in artifacts and flaws in unseen viewpoints.

In this paper, we propose Self-NeRF to solve the few-shot novel view synthesis task without additional priors. Our key point is to design a self-training framework, in which we jointly learn from seen views and a large number of auxiliary unseen views. Self-training [52] is a classic method for semi-supervised learning. It aims to learn from unlabeled data by iteratively imputing the labels for samples predicted with high confidence. In the novel view synthesis task, the labeled data refers to seen views while the unlabeled data refers to unseen views. Inspired by self-training, our self-training backbone leverages confident predictions in the previous iteration to produce pseudo-views for unseen views. Our pseudo-views can be categorized into two types: the warped pseudo-views generated through forward warping and the predicted pseudo-views which are the outputs of the previous iteration. The former provide local texture guidance, while the latter help improve the perceptual capability of the global structures. In other words, pseudo-views add more information to guide model training.

However, these pseudo-views still contain uncertain regions with inaccurate colors due to warping artifacts or occlusion. Assuming that all input pixel colors are reliable, NeRF will faithfully learn these uncertain pixels in pseudo-views, which results in reconstruction inconsistency across multiple views and degenerate solutions. To avoid this degeneracy, we propose an uncertainty-aware NeRF that autonomously learns a field of uncertainty from pseudo-views. Based on the output of the uncertainty field, we alleviate the impact of uncertain pixels. Specifically, we introduce our specialized warping embeddings and uncertainty embeddings to model image-dependent uncertain colors. Furthermore, we leverage cone tracing technique and a cone entropy regularization within a conical frustum to represent fine details. The cone entropy regularization imposes the model to compact representation in the unobserved views instead of collapsing to a trivial solution. Our experiments have proved that the proposed Self-NeRF shows state-of-the-art performance on few-shot novel view synthesis as shown in Fig. 1.

In summary, the main contributions of our paper are:

- We propose Self-NeRF, a novel iterative training scheme for synthesizing novel views from few-shot images without additional priors.

- We prove the convergence of our iterative Self-NeRF through theoretical deduction and experiments.
- We introduce a practical implementation of Self-NeRF, which leverages an uncertainty-aware NeRF, specialized embeddings and a cone entropy regularization to avoid degeneration due to pseudo-views.

2. Related Work

2.1. Novel view synthesis

Given a dense sampling of views, earlier works use view interpolation [6] and light fields [23, 15] to reconstruct novel views. To better represent the 3D scene, some works utilize proxy geometry [9] and explicit representation such as layered representations [40, 42], voxel [43], mesh [17] and point cloud [36, 48]. Recently, a plethora of learning-based methods [12, 13, 19, 25, 57] has received growing attention. Simultaneously, there is another line of work that uses volumetric representations to address the task of photo-realistic view synthesis. Neural radiance fields (NeRF) [29] employ an implicit neural representation of a 3D scene and use volumetric rendering to generate photo-realistically unseen views. Mip-NeRF [2] follows the step of NeRF and reasons about volumetric frustums along a cone for anti-aliasing. Mip-NeRF 360 [3] further extends it to model unbounded scenes with a non-linear scene parameterization, online distillation, and a novel distortion-based regularizer. Furthermore, recent works have made tremendous efforts to improve the rendering speed [31, 38, 44, 53], artistic effects [45, 11, 56], and generalization ability [26, 46] of NeRF. Some works [27, 49, 55] also adapt NeRF for dynamic scenes.

2.2. Few-shot view synthesis

The requirement of numerous calibrated images is a key limitation of NeRF. Some works [10, 39] attempt to decrease the data-hungriness by using the depth priors. Without the depth priors, MetaNeRF [24] uses data-driven priors recovered from a domain of training scenes. Pixel-NeRF [54] and Peng *et al.* [35] utilize the implicit spatial information in the local CNN features to construct the radiance fields. MVSNeRF [5] and IBRNet [46] employ earlier multi-view stereo methods to produce a multi-view feature volume. DietNeRF [18] resorts to the pretrained CLIP-ViT [37] and adapts its projected image embeddings as features to add global semantic information for novel views. These methods heavily rely on external supervisory signals such as depth information or additional pretrained encoders to synthesize novel views. Recently, some studies [4, 1, 51, 8] have introduced the augmentation of warped views with only few-shot images to improve neural radiance fields, which provides more training constraints designed in a small patch. However, these methods achieve suboptimal

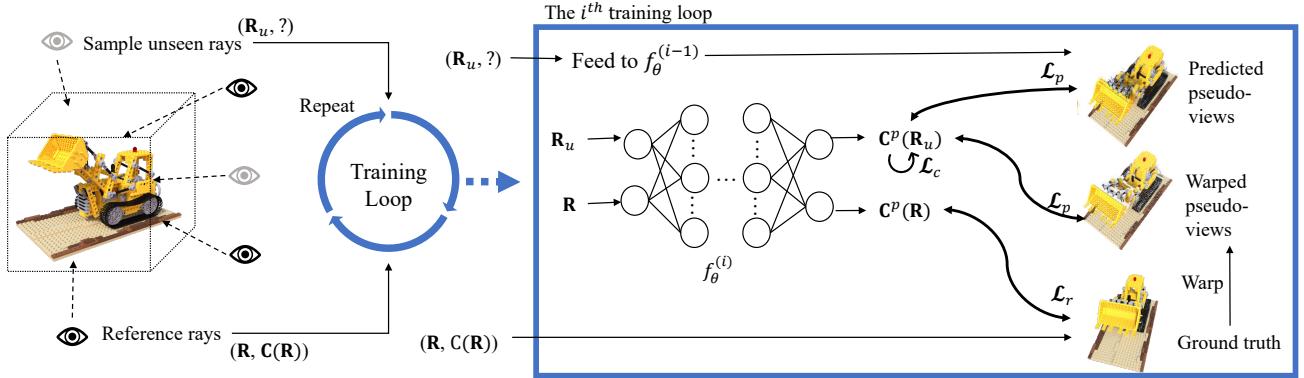


Figure 2: The iterative pipeline for Self-NeRF. In i^{th} iteration, we gather predicted pseudo-views synthesized by the model in the previous iteration for unseen views. The warped pseudo-views are collected through warping the seen views to unseen views. Then we train the uncertainty-aware NeRF f_θ^i in a supervised way with seen views and pseudo-views simultaneously.

performance since they ignore the uncertainty of warped views caused by the warping operations. Unlike the aforementioned works, some methods [20, 32, 7, 14] introduce a prior-free model. InfoNeRF [20] minimizes the ray entropy among seen and unseen poses and utilizes ray information gain reduction to alleviate reconstruction inconsistency across views. Although their strategies improve the quality of novel-view synthesis, they do not fully explore the full potential of unseen views. Artifacts such as blurring and cloud effects still exist in the synthesized image. By contrast, we fully leverage adequate pseudo-views and reduce the artifacts with our iterative training.

3. Method

3.1. Preliminaries

Neural radiance fields represent a 3D scene as a continuous implicit function f_θ , which outputs emitted radiance value and volume density when given a 3D position $\mathbf{x} \in \mathbb{R}^3$ and unit viewing direction $\mathbf{d} \in \mathbb{R}^3$. In practice, NeRF adapts a multi-layer perceptron (MLP) model to predict the corresponding volume density $\sigma \in [0, \infty]$ and color $\mathbf{c} \in [0, 1]^3$ given the queried \mathbf{x} and \mathbf{d} as follows:

$$(\sigma, \mathbf{c}) = f_\theta(\gamma(\mathbf{x}), \gamma(\mathbf{d})) \quad (1)$$

where γ is a predefined positional encoding.

To render the RGB color at the target pixel, NeRF samples points along the ray \mathbf{R} and integrates colors and densities based on the volume rendering. The ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ is emitted from the camera’s center \mathbf{o} along the direction \mathbf{d} . We compute the 3D position $\mathbf{x}_k = \mathbf{r}(t_k)$ for each sample point t_k . Thus the rendered color can be formulated as:

$$\mathbf{C}^r(\mathbf{R}) = \sum_i^N \exp\left(-\sum_j^{i-1} \sigma_j \delta_j\right) (1 - \exp(\sigma_i \delta_i)) \mathbf{c}_i \quad (2)$$

where N indicates the total number of sample points along \mathbf{R} , and δ_i represents the distance between the i^{th} and $(i+1)^{th}$ points. NeRF casts a single ray per pixel and may produce renderings that are excessively blurred or aliased. To ameliorate this issue, mip-NeRF [2] casts a cone that passes through the pixel’s center. To this aim, mip-NeRF derives integrated positional encoding (IPE), which is the integration over a volume covered by a conical frustum. Mip-NeRF reduces objectionable aliasing artifacts and significantly improves NeRF’s ability to represent fine details, while also being faster meanwhile.

Despite its impressive performance, Niemeyer *et al.* [32] find that the quality of mip-NeRF’s view synthesis drops significantly with only few views. Mip-NeRF fails to generalize well to novel views at test time due to training divergence. In addition, blurry artifacts or floaters may appear because of the inherent ambiguity of few-shot input.

3.2. Motivation and overview

For the novel view synthesis task, we treat pixels in training images as labeled data while the cast rays in novel views are considered as unlabeled data. Consequently, this task can be solved in a semi-supervised learning method. In this paper, we utilize an inductive semi-supervised learning method to harness large amounts of unseen views in combination with smaller sets of seen views. Specifically, we construct a novel iterative scheme in a self-learning manner [50]. The typical self-training algorithm [30, 47, 58] has three main steps: 1) train a good teacher model with labeled data, 2) use the teacher model to produce pseudo-labels on unlabeled data 3) train a student model on labeled and pseudo-labeled data simultaneously. In Self-NeRF, we iterate this algorithm a few times by putting back the student as a teacher to relabel the unlabeled data and training a new



Figure 3: Given four seen views (left), we obtain the predicted pseudo-views (middle) and warped pseudo-views (right) for unseen views.

student. We denote the student model in i^{th} iteration as f_θ^i , thus the trained teacher model in i^{th} iteration is $f_\theta^{(i-1)}$. In other words, the working pipeline of Self-NeRF is to train the model f_θ^i iteratively using the seen views and pseudo-views generated by $f_\theta^{(i-1)}$. The overview of Self-NeRF is shown in Fig. 2.

We observe that the performance of Self-NeRF is highly influenced by the quality of pseudo-views and the capability of the model in the iteration. To produce realistic rendering, we carefully design our pseudo-views (Sec. 3.3) and propose an uncertainty-aware model (Sec. 3.4). Sec. 3.5 further describes the inference and optimization of our model. In addition, we discuss the convergence of our iterative training and prove the feasibility of applying self-training on the novel view synthesis task in Sec. 3.6.

3.3. Pseudo-views in Self-NeRF

In i^{th} iteration, we first gather pseudo-views synthesized by f_θ^{i-1} for unseen views. As shown in Fig. 3, predicted pseudo-views capture the main structure of the scene, thus adding them to training views helps improve the perceptual capability of the global structures. However, predicted pseudo-views may introduce color shifts due to training divergence, even when these pixels are visible in the training images. To alleviate this issue, we generate warped pseudo-views through the forward warping. In more detail, we warp seen views \mathbf{I}_i to unseen views using the predicted depth map \mathbf{D}_i from f_θ^{i-1} and get warped pseudo-views \mathbf{I}_j . For pixel $p_i \in \mathbf{I}_i$ in the seen view, the corresponding pixel p_j in the unseen view is:

$$p_j = K_j T_{ij} (K_i^{-1} d_i p_i) \quad (3)$$

where $d_i \in \mathbf{D}_i$ is depth of p_i predicted by f_θ^{i-1} , K_i is the camera intrinsic matrices of \mathbf{I}_i and T_{ij} refers to the transform matrices from \mathbf{I}_i to \mathbf{I}_j . Pixels in warped pseudo-views are reprojected from the seen views, thus providing local texture guidance.

3.4. Uncertainty-aware model in Self-NeRF

While proving global structure information and local texture guidance, pseudo-views still contain uncertain pixels for unobserved regions. Assuming that all training pixels

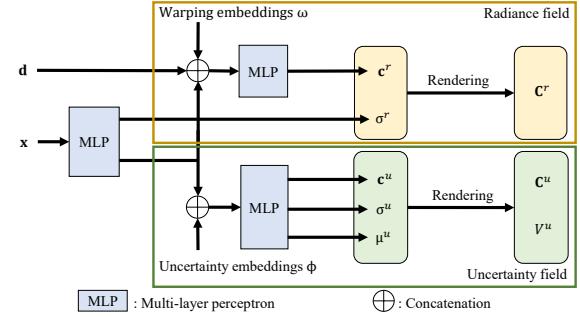


Figure 4: The architecture of our uncertainty-aware model.

are equally reliable, mip-NeRF tends to learn the uncertain colors. Consequently, the performance fluctuates wildly. Worse still, the low signal-to-noise ratio of pseudo-views sometimes leads to training collapse. To handle the challenges of uncertain pixels, we adapt mip-NeRF to be tolerant of uncertainty following Martin-Brualla *et al.* [27]. To this aim, we model the output color $\mathbf{C}^p(\mathbf{R})$ as the sum of the real color $\mathbf{C}^r(\mathbf{R})$ and uncertain color $\mathbf{C}^u(\mathbf{R})$ as follows:

$$\mathbf{C}^p(\mathbf{R}) = \mathbf{C}^r(\mathbf{R}) + \mathbf{C}^u(\mathbf{R}) \quad (4)$$

where $\mathbf{C}^r(\mathbf{R})$ and $\mathbf{C}^u(\mathbf{R})$ are learned from the radiance field and uncertainty field respectively during training. More specifically, our model adds two specialized embeddings and a branch to emit a field of uncertainty:

$$(\sigma, \mathbf{c}, \sigma^u, \mathbf{c}^u, \mu^u) = f_\theta(\gamma(\mathbf{x}), \gamma(\mathbf{d}), \omega, \phi) \quad (5)$$

where σ^u and \mathbf{c}^u are the predicted density and color of uncertainty field. μ^u is the uncertainty of the prediction. ω represents the learned warping embeddings that distinguish warped pseudo-views from predicted pseudo-views. Hence we assign the same ω to warped pseudo-views and seen views, explicitly implying that warped pseudo-views are re-projected from the seen views. Through this design, our model is expected to put more trust in the warped pseudo-views that are visible in seen views. ϕ denotes the uncertain embeddings that model per-image uncertain colors, thus each view has its own distinctive ϕ .

Owing to the uncertainty field, our model relaxes mip-NeRF's strict consistency assumption and imposes mip-NeRF to provide larger μ^u in the unobserved regions instead of collapsing to the trivial solution during the iterative process. As a result, Self-NeRF can attenuate the negative impact of uncertainty caused by warping or overfitting and gain information from adequate pseudo-views, yielding a superior image quality. The structure of our model is shown in Fig. 4.

3.5. Inference and optimization

Inference. For the query ray \mathbf{R} , we model the predicted color with an isotropic normal distribution with mean color

$\mathbf{C}^p(\mathbf{R})$ and variance $V^u(\mathbf{R})$. To get $\mathbf{C}^p(\mathbf{R})$ according to Eq. 4, we calculate $\mathbf{C}^r(\mathbf{R})$ through Eq. 2 and the learned variable colors $\mathbf{C}^u(\mathbf{R})$ can be analogously rendered with:

$$\mathbf{C}^u(\mathbf{R}) = \sum_i^N \exp\left(-\sum_j^{i-1} \sigma_j^u \delta_j^u\right) (1 - \exp(\sigma_i^u \delta_i^u)) \mathbf{c}_i^u \quad (6)$$

Similar to Eq. 6, $V^u(\mathbf{R})$ is approximated with a linear combination of sampled points:

$$V^u(\mathbf{R}) = \sum_i^N \exp\left(-\sum_j^{i-1} \sigma_j^u \delta_j^u\right) (1 - \exp(\sigma_i^u \delta_i^u)) \mu_i^u \quad (7)$$

In addition, we render the depth $d(\mathbf{R}) \in \mathbf{D}_i$:

$$d(\mathbf{R}) = \sum_i^N \exp\left(-\sum_j^{i-1} \sigma_j \delta_j\right) (1 - \exp(\sigma_i \delta_i)) t_i \quad (8)$$

$d(\mathbf{R})$ is leveraged in Eq. 3 to generate warped pseudo-views for the subsequent iteration.

Optimization loss. For ray \mathbf{R} with the ground truth RGB $\mathbf{C}_{gt}(\mathbf{R})$, the RGB loss is the negative log-likelihood (NLL):

$$\mathcal{L}_r = \frac{1}{2} \log(\mu^u)^2 + \frac{(\mathbf{C}_{gt}(\mathbf{R}) - \mathbf{C}^p(\mathbf{R}))^2}{2(\mu^u)^2} \quad (9)$$

Likewise, we obtain the pseudo loss for ray \mathbf{R} with pseudo-views $\mathbf{C}_{pseudo}(\mathbf{R})$ as following:

$$\mathcal{L}_p = \frac{1}{2} \log(\mu^u)^2 + \frac{(\mathbf{C}_{pseudo}(\mathbf{R}) - \mathbf{C}^p(\mathbf{R}))^2}{2(\mu^u)^2} + \frac{\lambda_u}{N} \sum_i^N \sigma_i^u \quad (10)$$

We further regularize the cone tracing with a cone entropy loss following Shannon Entropy [41]:

$$\mathcal{L}_c = - \sum_i^N \frac{1 - \exp(\sigma_i \delta_i)}{\sum_j^N 1 - \exp(\sigma_j \delta_j)} \log \frac{1 - \exp(\sigma_i \delta_i)}{\sum_j^N 1 - \exp(\sigma_j \delta_j)} \quad (11)$$

The total loss function to optimize our model is given by:

$$\mathcal{L} = \mathcal{L}_r + \lambda_1 \mathcal{L}_p + \lambda_2 \mathcal{L}_c \quad (12)$$

where λ_1 and λ_2 denote manual parameters to balance the loss terms. In particular, λ_1 decays exponentially by a factor of 2 at every 10k steps. The slowly decreasing weight is expected to help the optimization process avoid poor local minima so that pseudo-views provide guidance without conflicting with seen views.

3.6. Convergence analysis of Self-NeRF

Lee *et al.* [22] have analyzed self-learning technique and proved that it is an effect equivalent to a version of the entropy regularization. The unlabeled data can improve generalization performance even pseudo labels are not precise.

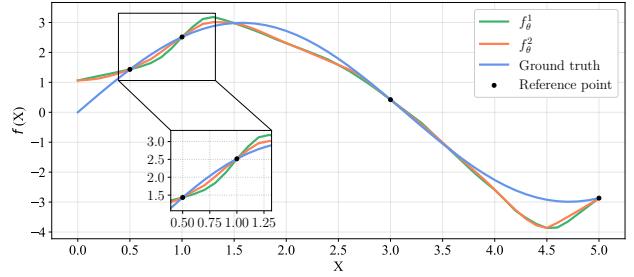


Figure 5: Comparison between the learned models f_θ^1 and f_θ^2 . We zoom in on the area in the blue box to demonstrate the improvement of f_θ^2 .

For the novel view synthesis task, our pseudo-views reduce overfitting theoretically by providing the possible solution for unseen views. In other words, the statement $P(n)$ that f_θ^n outperforms $f_\theta^{(n-1)}$ is true for any $n \in [2, \infty]$. We prove it by induction as follows.

Basic step. We first prove that the statement $P(n)$ is true for $n = 2$. For the sake of simplicity, let us assume that our target function is $F_{gt}(x) = \sin(x - 0.1) + \sin(x) + \sin(x + 0.1)$, which is an analog of integrating along the rays. Taking 4 pairs of labeled data as few-shot input, we tend to model $F_{gt}(x)$ with a naive network f_θ through our self-learning pipeline. f_θ comprise three fully-connected layers and has 100 neurons in the hidden layer. Following Sec. 3.3, we gather predicted pseudo-labels $f_\theta^{(i-1)}$ and assign them to unlabeled data x_u . Apart from that, we mimic warped pseudo-labels using $\frac{1}{2}(f_\theta^{(i-1)}(x_u) + F(x_u))$. Fig. 5 shows the performance of learned models $f_\theta^{(1)}$ and $f_\theta^{(2)}$. $f_\theta^{(1)}$ refers to the model trained solely with few-shot input. Besides the few-shot input, $f_\theta^{(2)}$ further uses 8 warped-labels and 12 predicted pseudo-labels. Noted that $f_\theta^{(2)}$ with a smooth curve is closer to $F_{gt}(x)$ than $f_\theta^{(1)}$. From a quantitative perspective, $f_\theta^{(2)}$ has lower mean absolute error than $f_\theta^{(1)}$. Therefore, $P(2)$ is proved to be true.

Inductive step. If the statement $P(k)$ is true where $k \in [2, \infty]$, the prediction of f_θ^k surpasses that of $f_\theta^{(k-1)}$. Hence the pseudo-labels from f_θ^k have a higher signal-noise ratio. When we train $f_\theta^{(k+1)}$ under the same setting as that in training $f_\theta^{(k)}$, the higher quality of pseudo labels ideally leads to a better model. Consequently, $f_\theta^{(k+1)}$ outperforms f_θ^k . In other words, the truth of $P(k)$ implies the truth of $P(k+1)$. Therefore, $P(n)$ is true for any $n \in [2, \infty]$. That is to say, the learned model will be improved with our iterative self-training, yielding photo-realistic novel-view synthesis results without additional priors.

Since we do not add priors during iterative training, there

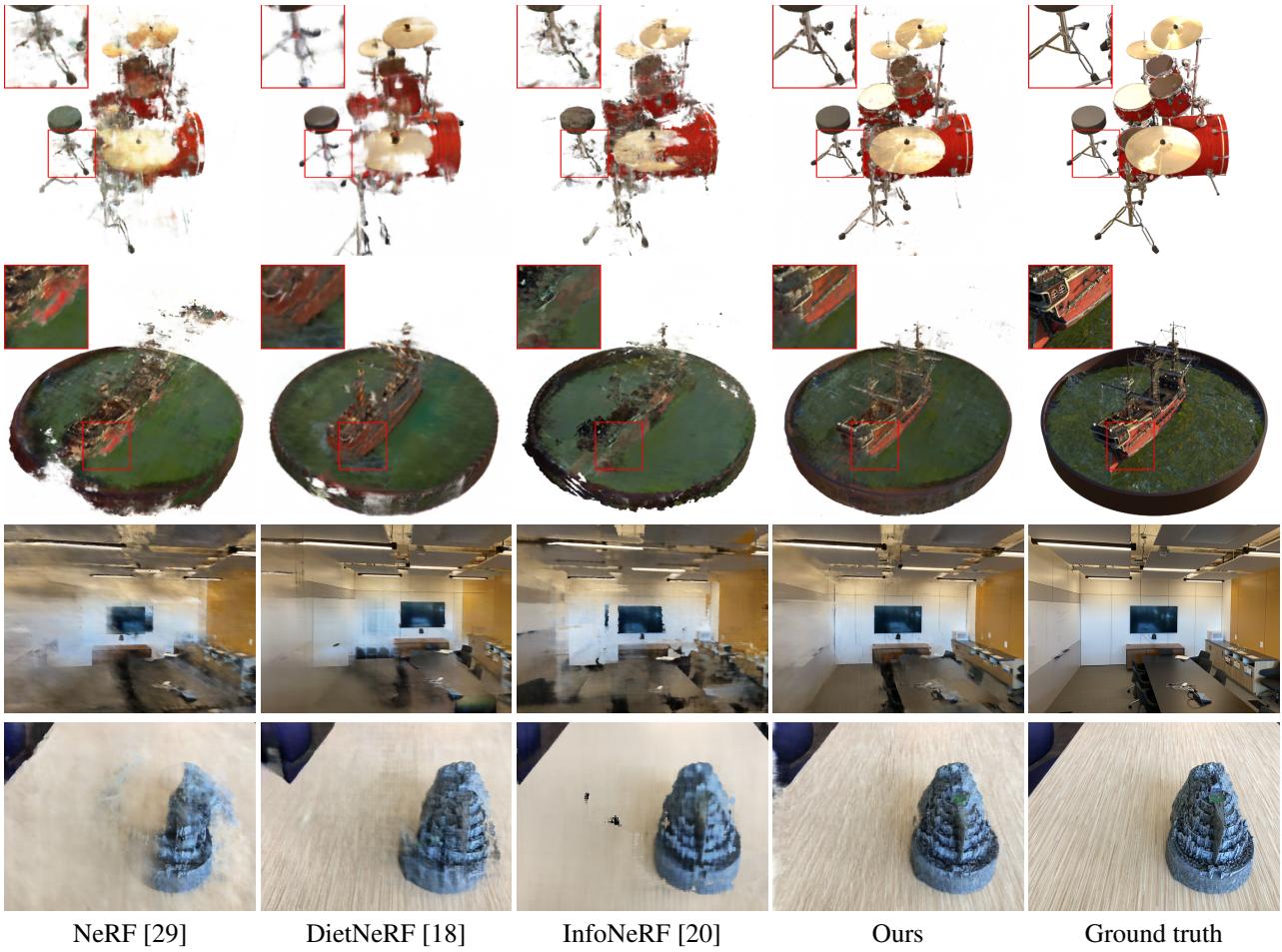


Figure 6: Qualitative comparison on the NeRF synthetic dataset [29] (Row 1-2) in 4-view settings and LLFF dataset [29] (Row 3-4) in 2-view settings.

exists the upper bound of Self-NeRF. To determine whether Self-NeRF has reached the upper bound, we sample some unseen views as validation dataset. We think that Self-NeRF has converged when the performance of the unseen views does no improve.

4. Experiments

4.1. Experimental settings

Baseline. We compare our method with baseline NeRF [29] and two state-of-the-art models for few-shot NeRF, including DietNeRF [18] and InfoNeRF [20].

Dataset. We demonstrate our approach on NeRF synthetic dataset [29] and LLFF dataset [28]. In NeRF synthetic dataset, we randomly sample 4 viewpoints as few-shot input for each scene. We run this experiment three times and compute the average scores on 200 testing images for evaluation. For LLFF dataset, we take one out of every eight

images from the collection of images for evaluation and randomly select 2 views from the remaining images as training images.

Evaluation Metrics. We measure the rendered image quality with several quantitative metrics, including the peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) and learned perceptual image patch similarity (LPIPS). PSNR and SSIM are popular metrics for evaluating reconstruction quality, while LPIPS can reflect the perception of humans more precisely.

Implement details. We implement Self-NeRF with PyTorch [34] while other approaches run on their own official codes. The Adam optimizer [21] is used with an initial learning rate of 0.00005 for optimization. We train other methods for 50K steps (about 15 hours) with their default settings and ensure they have converged. For a fair comparison, we train Self-NeRF in 2 iterations within 15 hours. All these experiments are conducted on a Tesla V100 GPU.

Table 1: Quantitative evaluation of our method against NeRF [29], DietNeRF [18] and InfoNeRF [20] on the NeRF synthetic dataset [29] and LLFF dataset [28].

Dataset	Metric	[29]	[18]	[20]	Ours
Synthetic	PSNR \uparrow	16.08	15.89	18.59	20.66
	SSIM \uparrow	0.795	0.719	0.810	0.840
	LPIPS \downarrow	0.284	0.330	0.225	0.179
LLFF	PSNR \uparrow	13.47	12.62	14.38	15.22
	SSIM \uparrow	0.280	0.253	0.319	0.372
	LPIPS \downarrow	0.607	0.574	0.559	0.443

4.2. Comparisons

Quantitative comparisons. Tab. 1 shows quantitative comparisons of our approach against NeRF, DietNeRF [18] and InfoNeRF [20]. Please refer to the supplemental materials for the detailed experimental results from individual scenes. Due to uncertainty, the blurry renderings produced by NeRF can outperform the visually appealing but incorrect renderings of DietNeRF on average error metrics like PSNR. However, the quantitative results of DietNeRF are still comparable to those of NeRF. InfoNeRF significantly reduces artifacts in the rendering, resulting in improved quantitative results. Self-NeRF outperforms all other methods in comparison in terms of all the evaluation metrics.

Qualitative comparisons. Fig. 6 depicts the rendering images synthesized by different methods. Compared to other methods, Self-NeRF achieves more realistic rendering in the novel views. Specifically, NeRF struggles to accurately reconstruct the scene and often produces blurry and cloudy artifacts. DietNeRF attempts to improve upon this by incorporating priors into the model, resulting in more reasonable and appealing renderings in some cases, such as the front of a ship. However, their use of low-dimensional CLIP [37] embeddings hinders the model’s ability to learn high-frequency details. InfoNeRF yields better results with fewer artifacts by imposing the sparsity on the scene. Despite this, their renderings of novel views still exhibit flaws and lack clear details, which makes the results look unrealistic at first glance. By contrast, Self-NeRF preserves the best geometry while generating realistic details. For example, our method successfully reconstructs the front of the ship and the texture of the table.

4.3. Ablation study

We validate our design choices by performing an ablation study on two scenes from NeRF synthetic dataset.

Design for networks. We study the effectiveness of our uncertainty-aware NeRF by replacing it with mip-NeRF [2] and NeRF-W [27]. Quantitative and qualitative results are given in Tab. 2 and Fig. 7, respectively. We observe

Table 2: Quantitative comparison with different models. Here “Base” represents the model in the previous iteration. On this basis, we replace our model with mip-NeRF [2] and NeRF-W [27] for the current iteration.

Scene	Metric	Base	[2]	[27]	Ours
Mic	PSNR \uparrow	19.38	21.42	21.40	23.75
	SSIM \uparrow	0.895	0.916	0.896	0.931
	LPIPS \downarrow	0.180	0.139	0.171	0.101
Ship	PSNR \uparrow	19.96	19.36	17.64	21.26
	SSIM \uparrow	0.719	0.711	0.719	0.757
	LPIPS \downarrow	0.318	0.364	0.388	0.263

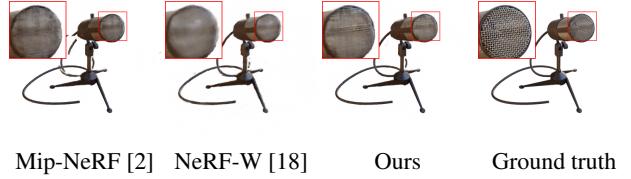


Figure 7: Validate the effectiveness of our uncertainty-aware NeRF in Self-NeRF.

that mip-NeRF utilizes cone tracing to capture fine details. However, the absence of uncertainty fields destabilizes the mip-NeRF, resulting in divergent behaviours such as the wire in the mic scene. On the contrary, NeRF-W is an uncertainty-aware model without cone tracing. Thus it alleviates the degradation due to uncertainty but produces blur contents. Our model combines the advantages of both and achieves favorable overall performance.

Choice of pseudo-views. We conduct an ablation study on different categories of pseudo-views. We report quantitative results in Tab. 3 and show qualitative results in Fig. 8. As discussed in Sec. 3.3, the predicted pseudo-views are incapable of accurately reconstructing colors for certain regions, resulting in color deviations when exclusively training with them. In contrast, pixels in the warped pseudo-views are more reliable yet may lack fine-grained details. From the results, it is evident that training with these pseudo-views simultaneously leads to optimal results.

4.4. Analysis

Robustness to the number of views. We report the variation curve of quantitative results under different numbers of training views for Lego scene in Fig. 9. Our method exhibits gradual performance improvement with an increasing number of training views. While Self-NeRF outperforms NeRF in all metrics, our method’s advantage reaches saturation point when using 16 training images. It is partly because

Table 3: Ablation study on different choices of the pseudo-labels. Here "Base" refers to the basic model in the previous iteration. "Warped" refers to training solely with the warped pseudo-views from the basic model while "Predicted" refers to training with the predicted pseudo-views.

Scene	Metric	Base	Warped	Predicted	Ours
Mic	PSNR \uparrow	19.38	21.61	23.47	23.75
	SSIM \uparrow	0.895	0.919	0.929	0.931
	LPIPS \downarrow	0.180	0.128	0.103	0.101
Ship	PSNR \uparrow	19.96	20.34	20.67	21.26
	SSIM \uparrow	0.719	0.744	0.749	0.757
	LPIPS \downarrow	0.318	0.290	0.270	0.263

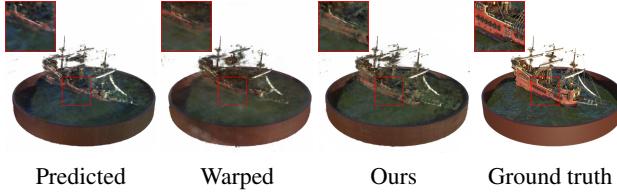


Figure 8: Validate the influence of different categories of pseudo-views.

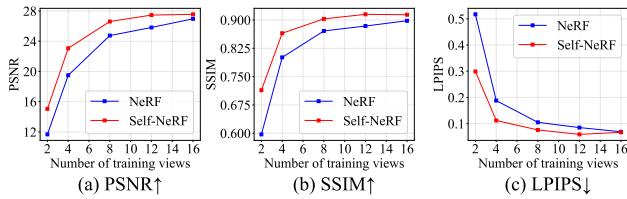


Figure 9: Quantitative results with different numbers of training views.

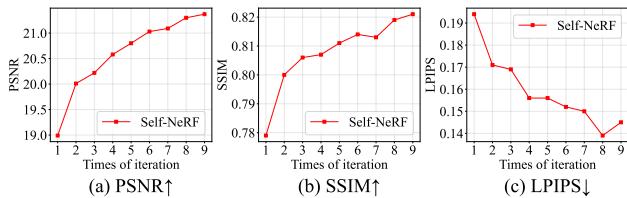


Figure 10: Quantitative results of different iterations.

the unseen regions decrease as the number of training views increases, thereby limiting the improvement from pseudo-views.

Improvement in iterative training. We report the quantitative results of various iterations in Fig. 10 and depict the outputs of Self-NeRF in Fig. 11. Self-NeRF has converged

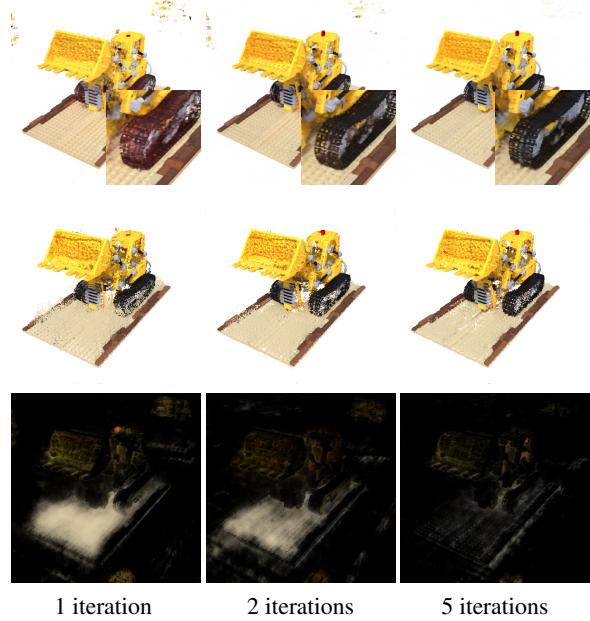


Figure 11: Visualization of the prediction of Self-NeRF (top row), the warped pseudo-views (middle row) and the corresponding uncertainty map (bottom row) in different iterations. With the increasing number of iterations, Self-NeRF corrects the color shifts on the wheel and reduces artifacts. The improved performance results in better pseudo-views with less uncertainty, which in turn benefits the training.

since the LPIPS deteriorates in the 9th iteration. Note that our uncertainty-aware NeRF is capable of detecting uncertain pixels and leverages pseudo-views to their full potential. It gradually reduces the artifacts and effectively mitigates color shifts as the number of iterations increases. Hence, our iterative process leads to continuous improvement of the overall quality of predictions.

5. Conclusion

In this paper, we propose Self-NeRF to synthesize novel views given few-shot images. Inspired by self-training, Self-NeRF iteratively generates pseudo-views and trains the model with seen views and pseudo-views jointly. In the iteration, we generate two categories of pseudo-views: predicted pseudo-views from the previous iteration and warped pseudo-views which are reprojected from seen views using depth-based forward warping. These pseudo-views are shown to have a stabilizing effect and alleviate the color shifts. To avoid the negative impact of uncertain pixels in pseudo-views, we propose an uncertainty-aware NeRF with specialized embeddings. We also utilize techniques such as cone entropy regularization to reconstruct fine details and facilitate optimization. Our experiments further demonstrate our method's competitiveness compared with

state-of-the-art models for few-shot novel view synthesis.

References

- [1] Young Chun Ahn, Seokhwan Jang, Sungheon Park, Ji-Yeon Kim, and Nahyup Kang. Panerf: Pseudo-view augmentation for improved neural radiance fields based on few-shot inputs. *ArXiv*, abs/2211.12758, 2022.
- [2] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5835–5844, 2021.
- [3] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5460–5469, 2021.
- [4] Matteo Bortolon, Alessio Del Bue, and Fabio Poiesi. Data augmentation for nerf: a geometric consistent solution based on view morphing. *ArXiv*, abs/2210.04214, 2022.
- [5] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14104–14113, 2021.
- [6] Shenchang Eric Chen and Lance R. Williams. View interpolation for image synthesis. *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, 1993.
- [7] Zhengjie Chen, Chen Wang, Yuanchen Guo, and Song-Hai Zhang. Structnerf: Neural radiance fields for indoor scenes with structural hints. *ArXiv*, abs/2209.05277, 2022.
- [8] Francois Darmon, B. Basile, Jean-Clement Devaux, Pascal Monasse, and Mathieu Aubry. Improving neural implicit surfaces geometry with patch warping. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6250–6259, 2021.
- [9] Paul E. Debevec, Camillo Jose Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996.
- [10] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12872–12881, 2021.
- [11] Zhiwen Fan, Yifan Jiang, Peihao Wang, Xinyu Gong, Dejia Xu, and Zhangyang Wang. Unified implicit neural stylization. In *European Conference on Computer Vision*, 2022.
- [12] John Flynn, Michael Broxton, Paul E. Debevec, Matthew DuVall, Graham Fyffe, Ryan S. Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2362–2371, 2019.
- [13] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deep stereo: Learning to predict new views from the world’s imagery. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5515–5524, 2015.
- [14] Qiancheng Fu, Qingshan Xu, Y. Ong, and Wenbing Tao. Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. *ArXiv*, abs/2205.15848, 2022.
- [15] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996.
- [16] Andrew Harley and Andrew Zisserman. Multiple view geometry in computer vision (2. ed.). 2003.
- [17] Dominic Jack, Jhony Kaesemodel Pontes, Sridha Sridharan, Clinton Fookes, Sareh Abolahrari Shirazi, Frédéric Maire, and Anders P. Eriksson. Learning free-form deformations for 3d object reconstruction. In *Asian Conference on Computer Vision*, 2018.
- [18] Ajay Jain, Matthew Tancik, and P. Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5865–5874, 2021.
- [19] Nima Khademi Kalantari, Tingxian Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (TOG)*, 35:1 – 10, 2016.
- [20] Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12902–12911, 2021.
- [21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- [22] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. 2013.
- [23] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’96*, page 31–42, New York, NY, USA, 1996. Association for Computing Machinery.
- [24] J. Li, Shuo Shang, and Ling Shao. Metanerf: Named entity recognition with meta-learning. *Proceedings of The Web Conference 2020*, 2020.
- [25] Kai-En Lin, Lei Xiao, Feng Liu, Guowei Yang, and Ravi Ramamoorthi. Deep 3d mask volume for view synthesis of dynamic scenes. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1729–1738, 2021.
- [26] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7814–7823, 2021.
- [27] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. *2021 IEEE/CVF Conference*

- on Computer Vision and Pattern Recognition (CVPR)*, pages 7206–7215, 2020.
- [28] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *arXiv: Computer Vision and Pattern Recognition*, 2019.
- [29] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 2020.
- [30] Subhabrata Mukherjee and Ahmed Hassan Awadallah. Uncertainty-aware self-training for few-shot text classification. In *Neural Information Processing Systems*, 2020.
- [31] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)*, 41:1 – 15, 2022.
- [32] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5470–5480, 2021.
- [33] Xuran Pan, Zihang Lai, Shiji Song, and Gao Huang. Activenerf: Learning where to see with uncertainty estimation. In *European Conference on Computer Vision*, 2022.
- [34] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.
- [35] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9050–9059, 2020.
- [36] C. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2016.
- [37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.
- [38] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14315–14325, 2021.
- [39] Barbara Roessler, Jonathan T. Barron, Ben Mildenhall, Pratul P. Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12882–12891, 2021.
- [40] Jonathan Shade, Steven J. Gortler, Li wei He, and Richard Szeliski. Layered depth images. *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 1998.
- [41] Claude E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:623–656, 1948.
- [42] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3d photography using context-aware layered depth inpainting. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8025–8035, 2020.
- [43] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. Deepvoxels: Learning persistent 3d feature embeddings. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2441, 2019.
- [44] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5449–5459, 2021.
- [45] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3825–3834, 2021.
- [46] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P. Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas A. Funkhouser. Ibrenet: Learning multi-view image-based rendering. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4688–4697, 2021.
- [47] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Loddon Yuille, and Fan Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10852–10861, 2021.
- [48] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7465–7475, 2019.
- [49] Tianhao Wu, Fangcheng Zhong, Andrea Tagliasacchi, Forrester Cole, and Cengiz Öztireli. D2nerf: Self-supervised decoupling of dynamic and static objects from a monocular video. *ArXiv*, abs/2205.15838, 2022.
- [50] Qizhe Xie, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. Self-training with noisy student improves imagenet classification. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2019.
- [51] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. Sinnerf: Training neural radiance fields on complex scenes from a single image. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, page 736–753, Berlin, Heidelberg, 2022. Springer-Verlag.
- [52] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *CoRR*, abs/2103.00550, 2021.

- [53] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5491–5500, 2021.
- [54] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4576–4585, 2020.
- [55] Jingbo Zhang, Xiaoyu Li, Ziyu Wan, Can Wang, and Jing Liao. Fdnerf: Few-shot dynamic neural radiance fields for face reconstruction and expression editing. *SIGGRAPH Asia 2022 Conference Papers*, 2022.
- [56] Kai Zhang, Nicholas I. Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. Arf: Artistic radiance fields. In *European Conference on Computer Vision*, 2022.
- [57] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.
- [58] Yang Zou, Zhiding Yu, Xiaofeng Liu, B. V. K. Vijaya Kumar, and Jinsong Wang. Confidence regularized self-training. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5981–5990, 2019.