A Multiplex Training Paradigm Improves Neural Radiance Field for (Almost) Free

Anonymous ICCV submission

Paper ID ****

Abstract

Recently Neural Radiance Field (NeRF) shows great success in rendering novel-view images of a certain scene by learning an implicit volumetric representation with only posed RGB images. NeRF usually optimizes a pixel-wise Mean Squared Error (MSE) loss, where one input sample corresponds one pixel. NeRF also learns and inferences pixel-wisely, as machine learning models make predictions input-wisely. It is known that the pixels of one image or one scene may form rich structural information. Unfortunately, the pixel-wise NeRF obviously failed to use the supervision of multiple pixels' structural information. In this paper, we mainly make three contributions. First, we propose the novel Stochastic Structural SIMilarity (S3IM) index which measure the similarity between two sets of pixels and capture non-local structural similarity information of stochastic sampled pixels. Second, We proposed a novel model-agnostic multiplex training paradigm for NeRF with the help of the non-local structural supervision information from S3IM. To the best of our knowledge, we are the first to formulate a multiplex training paradigm, which is computed over multiple inputs rather than a single input, for training of NeRF and relevant tasks in computer vision. Third, our extensive experiments demonstrate that the multiplex training paradigm can not only significantly improve the neural rendering performance of NeRF and relevant methods but also enhance the robustness to corrupted scenes, sparse scenes (few-shot learning), and dynamic scenes, while the extra computational cost is very limited.

1. Introduction

Synthesizing novel-view images of a 3D scene from a set of images is a long-standing task in computer vision and computer graphics [4, 6, 16, 10, 31], which may also serve as a prerequisite to recent AR and VR applications. This long-standing task has recently seen significant progress due to advances in learning-based neural rendering methods [25, 17, 19]. The learning-based neural rendering methods can represent 3D scenes from posed images towards photo-realistic novel view synthesis.

Particularly, benefited from strong representations of deep neural networks (DNNs), the recent method Neural Radiance Field (NeRF) [19] has showed impressive success on novel view synthesis of a specific scene by implicitly encoding volumetric density and color through a fully connected neural network (often referred to as a multilayer perceptron or MLP). NeRF regresses from a single 5D representation (x, y, z, θ, ϕ)—3D coordinates $\boldsymbol{x} = (x, y, z)$ plus 2D viewing directions $\boldsymbol{\theta} = (\theta, \phi)$—a single volume density σ and view-dependent RGB color $\boldsymbol{c} = (r, g, b)$, computed by fitting the model to a set of pixels (from training images). NeRF approximates this continuous 5D scene representation with an MLP network $f_\Theta : (\boldsymbol{x}; \boldsymbol{d}) \rightarrow (\boldsymbol{c}; \sigma)$ and optimize its weights Θ to map from each input 5D coordinate to its corresponding volume density and directional emitted color.

Without loss of generality, we focus on the learning module of NeRF and write the loss optimized by NeRF as

$$L(\Theta) = \frac{1}{\|\mathcal{R}\|} \sum_{\boldsymbol{r} \in \mathcal{R}} l_{\mathrm{MSE}}(\Theta, \boldsymbol{r}), \tag{1}$$

where l_{MSE} is the Mean Squared Error (MSE) loss for each pixel (/ray) $\boldsymbol{r} = (\boldsymbol{x_r}, \boldsymbol{d_r}, \boldsymbol{c_r})$ in the training data or data minibatch $\mathcal{R}$. Obviously, the MLP of NeRF learns and make inference pixel-wisely, because one data sample of the MLP corresponds to one pixel's information. However, this can be a serious but overlooked pitfall of training of NeRF.

In image quality assessment, the pixel-wise MSE-based metric, Peak Signal-to-Noise Ratio (PSNR) [7], is widely used. However, it is also well known that PSNR and MSE do not correlate well with perceived image quality [24, 37], because the images as well as the pixels may form rich structural information [15, 38, 39, 30] overlooked by previous pixel-wise metrics.

The Structural Similarity (SSIM) index not only reflects the structure of a set of pixels but also correlates with human visual systems significantly better than PSNR/MSE [38, 13]. SSIM is an important performance metric for evaluating NeRF models but not utilized for training NeRF models. The existing NeRF studies also overlooked the structural information in current training paradigm of NeRF, as the cur-

rent training paradigm only optimizes the pixel-wise MES loss but, unfortunately, fails to use the supervision of multiple pixels' structural information. This is not surprising. In conventional paradigm of machine learning, a model is considered to processes inputs independently.

May we propose a novel training paradigm that may capture structure information of a set of inputs/pixels other than the MSE loss of individual pixels? We can write the *multiplex loss* given by the novel training paradigm as

$$L_{\mathrm{M}}(\Theta) = \frac{1}{\|\mathcal{R}\|} \sum_{\boldsymbol{r}\in\mathcal{R}} l_{\mathrm{MSE}}(\Theta, \boldsymbol{r}) + \lambda L_{\mathrm{S3IM}}(\Theta, \mathcal{R}), \quad (2)$$

where $L_{\mathrm{S3IM}}(\Theta, \mathcal{R})$ is computed over a set of pixels (from a minibatch) and the hyperparameter λ adjusts the importance of S3IM. Different from $L_{\mathrm{S3IM}}(\Theta, \mathcal{R})$, the conventional loss $l_{\mathrm{MSE}}(\Theta, (\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{c}, \sigma))$ is computed over one individual pixel. We emphasize that $L_{\mathrm{S3IM}}(\Theta, \mathcal{R})$ cannot be expressed as the sum of another loss computed over individual pixels from $\mathcal{R}$. We call the proposed L_{S3IM} a multiplex loss, because it can help the MLP process multiple inputs as a whole multiplex input. We will discuss the exact form of S3IM in Section 3.

We summarize main contributions as follows.

1. We propose the novel Stochastic Structural SIMilarity (S3IM) index which measure the similarity between two sets of pixels and capture non-local structural similarity information of stochastic sampled pixels. In contrast, the classical Structural SIMilarity (SSIM) index measures the similarity between two images rather than two set of pixels and only captures the local structural information of only nearby pixels, while PSNR only captures the pixel-wise MSE of RGB values.

2. We proposed a novel multiplex training paradigm for NeRF with the help of the non-local structural supervision information from S3IM. The proposed multiplex method is model-agnostic and can be generally applied to various NeRF models. To the best of our knowledge, we are the first to formulate a multiplex loss, which is computed over multiple inputs rather than a single input, for training of NeRF and relevant tasks in computer vision.

3. Our extensive experiments demonstrate that the multiplex S3IM may significantly improve the performance of popular NeRF models, while the extra computational cost is very limited. The performance improvement become even more significant for those difficult scene rendering tasks, where training data is corrupted or sparse.

Structure of the paper. TBD.

2. Background

In this section, we introduce background knowledge.

2.1. NeRF

Recall that NeRF maps from a single 5D representation (x, y, z, θ, ϕ) to a single volume density σ and view-dependent RGB color $\boldsymbol{c} = (r, g, b)$ with an MLP network: $f_\Theta : (\boldsymbol{x}; \boldsymbol{d}) \to (\boldsymbol{c}; \sigma)$. For a target view with pose, a camera ray can be parameterized as $\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{d}$ with the ray origin $\boldsymbol{o}$ and ray unit direction $\boldsymbol{d}$. The expected color $\boldsymbol{C}(\boldsymbol{r})$ of camera ray $\boldsymbol{r}(t)$ with near and far bounds t_n and t_f is

$$\hat{\boldsymbol{C}}(\boldsymbol{r}) = \int_{t_n}^{t_f} T(t)\sigma(t)\boldsymbol{c}(t)dt, \quad (3)$$

where $T = \exp(-\int_{t_n}^{t} \sigma(s)ds)$ denotes the accumulated transmittance along the ray from t_n to t. For simplicity, we have ignored the coarse and fine renderings via different sampling methods.

The rendered image pixel value for camera ray $\boldsymbol{r}$ can then be compared against the corresponding ground truth pixel value $\boldsymbol{C}(\boldsymbol{r})$, for all the camera rays. The conventional NeRF rendering loss is the MSE loss

$$L(\Theta) = \frac{1}{\|\mathcal{R}\|} \sum_{\boldsymbol{r}\in\mathcal{R}} \|\hat{\boldsymbol{C}}(\boldsymbol{r}) - \boldsymbol{C}(\boldsymbol{r})\|^2. \quad (4)$$

Obviously, NeRF is a pixel-wise model. The loss computed over one pixel does not affect the loss computed over another one.

2.2. Quality Metrics: PSNR and SSIM

PSNR and SSIM are two most popular metrics of image quality assessment [38, 13]. For simplicity, we take grey-level (8 bits) images as examples. Given a test image I_a and a reference image I_b, both of size $W \times H \times C$, the PSNR can be defined as

$$\mathrm{PSNR}(I_a, I_b) = 10\log_{10}\left(\frac{255^2}{MSE(I_a, I_b)}\right), \quad (5)$$

where

$$\mathrm{MSE}(I_a, I_b) = \frac{1}{WHC}\sum_{i,j,k}(I_{b,ijk} - I_{a,ijk})^2. \quad (6)$$

It is easy to see that PSNR directly depends on MSE and overlooks the information of a set of pixels.

In contrast, SSIM a well-known quality metric that can capture local structural similarity between images or patches. SSIM is considered to be correlated with the quality perception of the human visual system well and is widely used for evaluating NeRF [38, 13]. Suppose $\boldsymbol{a} = \{a_i | i = 1, 2, 3, \ldots, n\}$ and $\boldsymbol{b} = \{b_i | i = 1, 2, 3, \ldots, n\}$ to be two discrete non-negative signals paired with each other (e.g. two image patches extracted from the same spatial location from paired images). We denote the mean intensity

of a signal as μ (e.g. $\mu_a = \frac{1}{n}\sum_{i=1}^{n} a_i$), the standard deviation of a signal as σ^2 (e.g. $\sigma_a^2 = \frac{1}{n-1}\sum_{i=1}^{n}(a_i - \mu_a)^2$), and the covariance between two signals as σ_{ab}^2 (e.g. $\sigma_{ab}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(a_i - \mu_a)(b_i - \mu_b)$).

SSIM is expressed by the combination of three terms which are the luminance, contrast, and structure comparison metrics:

$$\mathrm{SSIM}(a, b) = l(\boldsymbol{a}, \boldsymbol{b})c(\boldsymbol{a}, \boldsymbol{b})s(\boldsymbol{a}, \boldsymbol{b}). \tag{7}$$

The luminance $l(\boldsymbol{a}, \boldsymbol{b})$, contrast $c(\boldsymbol{a}, \boldsymbol{b})$, and structure comparison $s(\boldsymbol{a}, \boldsymbol{b})$ are, respectively, written as

$$l(\boldsymbol{a}, \boldsymbol{b}) = \frac{2\mu_a\mu_b + C_1}{\mu_a^2 + \mu_b^2 + C_1}, \tag{8}$$

$$c(\boldsymbol{a}, \boldsymbol{b}) = \frac{2\sigma_a\sigma_b + C_2}{\sigma_a^2 + \sigma_b^2 + C_2}, \tag{9}$$

$$s(\boldsymbol{a}, \boldsymbol{b}) = \frac{\sigma_{ab} + C_2}{\sigma_a\sigma_b + C_3}, \tag{10}$$

where C_1, C_2, and C_3 are small constants given by

$$C_1 = (K_1 L)^2,\ C_2 = (K_2 L)^2, \text{ and } C_3 = \frac{C_2}{2}. \tag{11}$$

Following the common setting [38, 19], we set $K_1 = 0.01$ and $K_1 = 0.03$ in this paper. The data range L is 1 for pixel RGB values. The range of SSIM lies in $[-1, 1]$.

In practice of image quality assessment, people usually apply the SSIM index locally rather than globally. The local statistics μ_a, σ_a, and σ_{ab} are computed within a local $K \times K$ kernel window, which moves with a stride size s over the entire image. For example, for evaluating NeRF, people often use the kernel size 11×11, the stride size 1, and the circular symmetric Gaussian weighting function $\boldsymbol{w} = \{w_i | i = 1, 2, \ldots, n\}$, with standard deviation of 1.5 samples, normalized to unit sum ($\sum_i w_i = 1$). The local statistics are then written as

$$\mu_a = \sum_{i=1}^{n} w_i a_i, \tag{12}$$

$$\sigma_a = \left(\sum_{i=1}^{n} w_i (a_i - \mu_a)^2\right)^{\frac{1}{2}}, \tag{13}$$

$$\sigma_{ab} = \left(\sum_{i=1}^{n} w_i (a_i - \mu_a)(b_i - \mu_b)\right)^{\frac{1}{2}}. \tag{14}$$

At each step, the local statistics and SSIM index are calculated within the local window. The final SSIM metric for evaluating NeRF is actually the mean SSIM (MSSIM) which is computed by averaging the SSIM indexes over each step.

3. Methodology

In this section, we propose and a novel multiplex loss S3IM and a novel multiplex training paradigm for NeRF.

Our motivation is to let the overlooked structure information contained in the set of pixels/images supervise the learning of NeRF. Thus, we must define a multiplex-style loss over a set of pixels, which may capture structure information, rather than the conventional loss (e.g. MSE) defined over a individual pixel.

The pixels in one local patch can contain certain positional information. However, due to stochastic training of NeRF, the randomly sampled pixels in one minibatch $\mathcal{R}$ cannot form a local patch and completely lose the positional relation. While the original SSIM is differentiable, we cannot directly optimize it to improve NeRF.

In this paper, we propose a stochastic variant of SSIM, namely S3IM, for stochastic training of NeRF. The idea is concise. Suppose we have B (e.g. 1024) pixels per minibatch and choose the kernel size and the stride size of S3IM as $K \times K$(e.g. 4×4) and $s = K$. We choose the stride size equal to the kernel size, because the stochastic patches from one minibatch will be independent without overlapped pixel in this case. We summarize S3IM as three steps: **(1)** we let B rays/pixels from a dataset/minibatch $\mathcal{R}$ randomly form a rendered patch $\mathcal{P}(\hat{\mathcal{C}})$ and the corresponding ground-truth image patch $\mathcal{P}(\mathcal{C})$, where $\hat{\mathcal{C}} = \{\hat{\boldsymbol{C}}(\boldsymbol{r}) | \boldsymbol{r} \in \mathcal{R}\}$ and $\mathcal{C} = \{\boldsymbol{C}(\boldsymbol{r}) | \boldsymbol{r} \in \mathcal{R}\}$; **(2)** we compute SSIM with the kernel size $K \times K$ and the stride size K over the rendered patch and the corresponding ground-truth patch, which is exactly an estimator of the proposed S3IM over the paired stochastic patches; **(3)** due to stochasticity of $\mathcal{P}(\cdot)$, we may repeat steps **(1)** and **(2)** M times and average the M estimated SSIM values to obtain the final S3IM over the rendered RGB values $\hat{\boldsymbol{C}}(\mathcal{R})$ and the ground-truth RGB values $\boldsymbol{C}(\mathcal{R})$ as

$$\mathrm{S3IM}(\hat{\mathcal{R}}, \mathcal{R}) = \frac{1}{M}\sum_{m=1}^{M} \mathrm{SSIM}(\mathcal{P}^{(m)}(\hat{\mathcal{C}}), \mathcal{P}^{(m)}(\mathcal{C})), \tag{15}$$

where SSIM needs to apply the kernel size $K \times K$ and the stride size K. We note that Step **(3)** can be well vectorized and still requires only one time back-propagation. Thus, the extra computational cost of our multiplex training is very limited.

As S3IM lies in $[-1, 1]$ and positively correlated with image quality, we define the S3IM-based loss L_{S3IM} as

$$\begin{aligned} L_{\mathrm{S3IM}}(\Theta, \mathcal{R}) &= 1 - \mathrm{S3IM}(\hat{\mathcal{R}}, \mathcal{R}), \\ &= 1 - \frac{1}{M}\sum_{m=1}^{M} \mathrm{SSIM}(\mathcal{P}^{(m)}(\hat{\mathcal{C}}), \mathcal{P}^{(m)}(\mathcal{C})). \end{aligned} \tag{16}$$

We illustrate the overview of multiplex training in Figure XXX. We present the pseudocode in Algorithm 1; for

Algorithm 1: Multiplex Training via S3IM (for NeRF or other Neural Fields)

Input: The pixel/ray dataloader $\mathcal{D}$, the batch size as B, the hyperparameters $\{\lambda, K, M\}$ for S3IM
Output: model $f_\Theta(\cdot)$

Let $\mathcal{A}$ be an SGD-like training algorithm;
while *no stopping criterion has been met* **do**
  Sample a data minibatch of rays $\mathcal{R}$ from $\mathcal{D}$;
  Obtain the ground-truth pixels $\mathcal{C} = \{\boldsymbol{C}(\boldsymbol{r})|\boldsymbol{r} \in \mathcal{R}\}$;
  Compute the rendered pixels $\hat{\mathcal{C}} = \{\hat{\boldsymbol{C}}(\boldsymbol{r})|\boldsymbol{r} \in \mathcal{R}\}$;
  for $m = 1$ **to** M **do**
    Initialize the stochastic patch generation function $\mathcal{P}^{(m)}$;
    Transform the rendered pixels into the rendered stochastic patch $\mathcal{P}^{(m)}(\hat{\mathcal{C}})$;
    Transform the ground-truth pixels into the ground-truth stochastic patch $\mathcal{P}^{(m)}(\mathcal{C})$;
    Compute $\mathrm{SSIM}(\mathcal{P}^{(m)}(\hat{\mathcal{C}}, \mathcal{P}^{(m)}(\mathcal{C}))$ with the given kernel $K \times K$ and the stride size K;
  end
  Obtain the S3IM loss $L_{\mathrm{S3IM}}(\Theta) = 1 - \frac{1}{M}\sum_{m=1}^{M} \mathrm{SSIM}(\mathcal{P}^{(m)}(\hat{\boldsymbol{C}}(\mathcal{R})), \mathcal{P}^{(m)}(\boldsymbol{C}(\mathcal{R})))$;
  Obtain the conventional MSE loss $L(\Theta) = \frac{1}{\|\mathcal{R}\|}\sum_{\boldsymbol{r}\in\mathcal{R}} \|\hat{\boldsymbol{C}}(\boldsymbol{r}) - \boldsymbol{C}(\boldsymbol{r})\|^2$;
  $L_M(\Theta) = L(\Theta) + \lambda L_{\mathrm{S3IM}}(\Theta)$;
  Compute the gradient $\nabla L_{\mathrm{M}}(\Theta)$;
  Update the model parameters Θ by $\mathcal{A}$;
end

simplicity and generality, we focus on the neural network module and ignore the details of sampling in Equation (3). The multiplex trianing paradigm via S3IM brings in three hyperparameters λ, K, and M. If we let M approach $+\infty$, we will eliminate the stochasticity of S3IM and obtained its expected value. However, simply choosing $K = 4$ and $M = 10$ is fine in practice, and only the loss weight λ requires fine-tuning in practice.

Why not optimize SSIM directly? Even if we use a pixel dataloader, which yields a minibatch of pixels per iteration, and a local-patch dataloader, which yields a minibatch of local patches per iteration, our ablation study in Section 5 support that optimizing S3IM (over stochastic patches) significantly outperforms optimizing SSIM (over conventional local patches). This is not surprising. SSIM over local patches can only capture structural information carried by nearby pixels from one image, while S3IM over stochastic patches can capture non-local structural information carried by distant pixels from various images. The additional dataloader not only brings in extra coding and computational costs but also hurts the performance of NeRF.

4. Related Work

In this section, we review representative related works and discuss their relations to our method.

Neural Fields and Neural Radiance Fields Fields can continuously parameterize an underlying physical quantity of an object or scene over space and time. Since a long time ago, fields have been used to describe physical phenomena [28], compute image gradients [29], simulate collisions [23]. Recent advances showed increased interest in employing coordinate-based neural networks to parameterize some physical quantities over space and time, such as a neural network that maps a 3D spatial coordinate to a flow field in fluid dynamics, or a colour and density field in 3D scene representation. Such networks are often referred to as neural fields [41]. The application of neural fields in visual computing has make remarkable progress on various computer vision problems such as 3D scene reconstruction and generative modelling.

Among them, NeRF [19] is one of the most representative neural field method. We let NeRF serve as the main baseline method for evaluating the proposed S3IM and the resulted multiplex training paradigm. The line of NeRF has developed a number of useful NeRF variants. NeRF++ [45] helped resolve the shape-radiance ambiguity of NeRF. Mip-NeRF [1] adopted a multiscale representation method and significantly improved the quality of representating fine details. Mip-NeRF 360 [2], an extension of Mip-NeRF, furthe uses a non-linear scene parameterization, online distillation, and a novel distortion-based regularizer to overcome the challenges presented by unbounded scenes. D-NeRF [26] extends neural radiance fields to modeling dynamical scences. Some works, such as Pixel-NeRF [44] and Reg-NeRF [21], focused on view synthesis from sparse inputs. NeRF−− [40] performs view synthesis by estimating approximate camera poses rather than known camera poses. As the common NeRF methods suffers a lot from slow training and inference, some works, inluding DVGO [34], Fast-NeRF [8], Kilo-NeRF [27], aimed at accelerating training and inference of NeRF. The proposed multiplex training paradigm via S3IM, a powerful alternative to the existing pixel-wise training paradigm, is model-agnostic and orthogonal to these NeRF variants.

Image Quality Metrics As we mentioned above, PSNR and SSIM are two most popular image quality metrics. Beyond them, we have also seen other useful metrics. Multiscale SSIM [39] is a variant of SSIM, which incorporates image details at different resolutions. However, Multiscale SSIM still can only capture local structural information car-

Table 1. Evaluation on neural rendering of scenes in T&T Datset.

Model	Training	M60			Playground			Train			Truck		
		PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)
DVGO	Standard	17.49	0.647	0.501	22.74	0.669	0.467	17.19	0.566	0.533	22.01	0.704	0.386
DVGO	Multiplex	**17.71**	**0.652**	**0.483**	**22.75**	**0.681**	**0.444**	**18.24**	**0.581**	**0.491**	**22.44**	**0.730**	**0.338**
Mip-NeRF	Standard	-	-	-	-	-	-	-	-	-	-	-	-
Mip-NeRF	Multiplex	-	-	-	-	-	-	-	-	-	-	-	-

ried by nearby pixels.

Deep features learned by DNNs has unreasonable effectiveness as a perceptual metric for measuring the similarity between two sets of perceptual features [46]. Thus, the Learned Perceptual Image Patch Similarity (LPIPS) metric [46] serves as the third rendering quality metric in NeRF and related studies, because it agrees surprisingly well with humans. The Fréchet inception distance (FID) score [11] is another metric which measure the distance-based similarity of perceptive features, but it is more widely used for evaluating generative models, such as Generative Adversarial Networks [5, 9] (GAN) and Diffusion Models [32, 33, 12]. Both LPIPS and FID metrics inevitably have certain stochasticity, because deep features are learned from stochastic training of DNNs. However, the stochasticity does not affect the usefulness and popularity of LPIPS and FID as image quality metrics.

S3IM can also be considered as a image quality metric, while we only use S3IM as a differentiable training objective in this paper. More specifically, S3IM measures the structural similarity of two paired sets of pixels(/signals), which may or may not form images. By analyzing the stochastic patch consists of random pixels, S3IM can capture non-local structural information carried by nearby/distant pixels. S3IM also inevitably have certain stochasticity like LPIPS and FID, while S3IM does not depend on deep features given by DNNs.

Multiplexing in Machine Learning The idea of multiplexing inputs has only been touched by very few machine learning studies recently. The most relevant work may be DataMux [20] which enables DNNs to process multiple inputs (e.g. images and sentences) simultaneously using a single compact representation by employing a multiplexing encoding layer and a demuliplexing decoding layer after the neural network architecture. DataMux which uses an additional layer is a kind of novel model architecture design rather than our multiplex training paradigm which is model-agnostic.

5. Empirical Analysis and Discussion

In this section, we conduct extensive experiments to demonstrate that multiplex training via S3IM significantly outperform the conventional training paradigm for NeRF and its variants.

The main principle of our experimental setting is to fairly compare multiplex training and standard training for NeRF and relevant neural field methods. We let the experimental settings follow original papers to reproduce the baselines, unless we specify otherwise. Fortunately, most of our baseline results are even slightly stronger than the results reported in original papers. This make our empirical evidence more convincing. We present the experimental settings in details in Appendix A.

Evaluation on Benchmark Datasets We first study how multiplex training via S3IM improve generalization of NeRF on two benchmark datasets, Tanks and Temples (T&T) [14] and LLFF [18]. We mainly use two popular NeRF variants, an accelerated variant DVGO and an multiscale-representation variant Mip-NeRF. Our experimental results in Table 1 suggest that multiplex training via S3IM consistently improve all three common quality metrics of NeRF. We also notice that the improvements of SSIM and LPIPS, which correlate with human perception better, are usually more significant. As PSNR directly depends on MSE, it is surprising to see the improvement of PSNR given the fact that S3IM distract the training objective from the original MSE loss. It means that S3IM significantly improve the generalization of NeRF.

We visualize the qualitative comparisons on the challenging parts in Figure XXX.

Neural rendering of dynamic scenes Rendering novel photo-realistic views of dynamic scenes is a more difficult task. A NeRF variant, D-NeRF, has been proposed to model dynamic scenes. We also study how multiplex training via S3IM improves D-NeRF on dynamic-scene datasets [26]. While the baseline in our work is even stronger than the original paper, multiplex training via S3IM still significantly improve the performance of D-NeRF on dynamic scenes.

Neural rendering from sparse or few images Learning with sparse or very few examples is a hot topic in machine learning [36]. In practice, it is possible that we cannot collect many images of a scene. Does multiplex training still work well when we have only sparse or even few images? To answer the question, we train DVGO on the sparse version of the Truck, called Sparse Truck, where we randomly remove a portion of original training images. We choose the fast NeRF variant DVGO because DVGO is more environment-

Table 2. Evaluation on neural rendering of dynamic scenes, Mutant and LEGO [26].

Model	Training	LEGO PSNR(↑)	LEGO SSIM(↑)	LEGO LPIPS(↓)	Mutant PSNR(↑)	Mutant SSIM(↑)	Mutant LPIPS(↓)
D-NeRF	Standard	21.29	0.821	0.0634	32.14	0.972	0.0181
D-NeRF	Multiplex	**23.36**	**0.900**	**0.0482**	**32.76**	**0.976**	**0.0151**

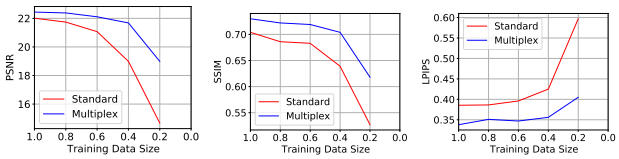

Figure 1. We plot the curves of PSNR, SSIM, and LPIPS with respect to the training data size. The improvement of multiplex training can be even more significant when the training data size decreases. Model: DVGO. Dataset: T&T-Truck.

Table 3. Evaluation on few-shot learning. We only keep eight training images from the Truck Scene, T&T Dataset.

Model	Training	PSNR(↑)	SSIM(↑)	LPIPS(↓)
DVGO	Standard	11.37	0.343	0.704
DVGO	Multiplex	**13.43**	**0.372**	**0.610**

friendly and can significantly reduce the energy cost and carbon emission of our experiments. The experimental results in Figure 1 suggest that the performance improvement of multiplex training can be very significant when we have sparse training data. For example, when only 20% of the original training data set is available, the improvement of PSNR can be surprisingly up to 4.32!

We further evaluate multiplex training on a few-shot learning task, where we only keep eight training images. The few-shot learning experimental result in Table 5 also support the significant advantage of multiplex training via S3IM.

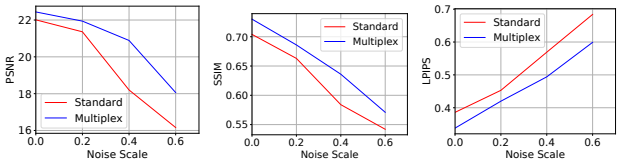

Figure 2. We plot the curves of PSNR, SSIM, and LPIPS with respect to the image noise scale std. The improvement of multiplex training can be even more significant when training image is corruppted by random noise. Model: DVGO. Dataset: T&T-Truck.

Robustness to image corruption While common benchmark datasets in NeRF studies are relatively clear, the training images in practice may be corrupted or noisy due to the realistic limitation of data collection. Gaussian image noise in digital images often arise during acquisition due to the inherent noise in the sensors [3]. The robustness to image corruption can be an important performance metric of NeRF but unfortunately overlooked by most existing NeRF studies.

We use the Truck scene to make an image-corrupted dataset, called Corrupted Truck, where we inject Gaussian noise with the standard deviation as std into original Truck images (each RGB value lies in $[0, 1]$) and obtain the corrupted version. We evaluate DVGO on the Corrupted Truck scence.

The experimental results in Figure 2 show that multiplex training via S3IM can signficantly improve the robustness to image corruption.

Ablation Study

Neural Surface Reconstruction In previous experiments, we mainly focus on NeRF family. We further report that it is very easy and effective to generalize multiplex training into other neural field methods. Reconstructing surfaces from multi-view images is also a fundamental problem in computer vision. Recent neural surface reconstruction methods [43, 42, 22, 35] are another kind of neural field method orthogonal to NeRF. To evaluate the universal effectiveness of the proposed method, we empirically study multiplex training for a classical neural surface reconstruction, called NeuS [35], which can render both RGB images and surface information. We choose reconstructing Advanced Scenes of T&T as the benchmark tasks. We add a new metric, Chamfer Distance, which can measure rendering quality of surface by computing the distance of two sets of points clouds.

Our experimental results in Table 4 suggest that multiplex training via S3IM can significantly improve neural surface reconstruction methods in terms of all four quality metrics. Multiplex training shows even more significant advantage on neural surface reconstruction. This shows the surprisingly effectiveness and universality of multiplex training in more fields.

Discussion We have demonstrated the advantage of multiplex training over standard training in extensive experiments, which are even beyond common benchmark NeRF tasks. We also does not notice negative effects of multiplex training in these experiments. Moreover, the extra computational cost of multiplex training is very limited and less than 5% in our experiments. It indicates the potential wide applications of multiplex training in practice.

Table 4. Evaluation on neural surface reconstruction of Advanced Scenes [14] .

Model	Training	Scene 1				Scene 2				Scene 3				Scene 4			
		PSNR(↑)	SSIM(↑)	LPIPS(↓)	Chamfer(↓))	PSNR(↑)	SSIM(↑)	LPIPS(↓)	Chamfer(↓))	PSNR(↑)	SSIM(↑)	LPIPS(↓)	Chamfer(↓))	PSNR(↑)	SSIM(↑)	LPIPS(↓)	Chamfer(↓))
NeuS	Standard	20.73	0.636	0.393	-	21.26	0.739	0.434	-	17.58	0.551	0.428	-	20.32	0.554	0.398	-
NeuS	Multiplex	**21.64**	**0.668**	**0.342**	-	**21.76**	**0.744**	**0.391**	-	**18.36**	**0.589**	**0.375**	-	**21.14**	**0.579**	**0.373**	-

6. Conclusion

Recently, NeRF and its variants have made great empirical success in the long-standing problem, novel-view rendering of 3D scenes. However, the current training paradigm only uses pixel-wise supervision information for training DNNs and overlook rich structural information contained in the set of pixels. In this paper, we proposed a novel multiplex training paradigm to push the performance limit of NeRF and its variants by using the non-local structural information of sets of pixels from S3IM. Our extensive experiments demonstrate that multiplex training via S3IM can significantly improve the performance of popular NeRF models, while the extra computational cost is very limited. The performance improvement become even more significant for those difficult scene rendering tasks, where training data is corrupted, sparse, or dynamic. The proposed method not only is model-agnostic to NeRF but also generally improves other neural field methods (e.g. NeuS). We will explore generalized applications and theoretical mechanism of multiplex training via S3IM in future.

References

[1] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 4

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 4

[3] Ajay Kumar Boyat and Brijendra Kumar Joshi. A review paper: Noise models in digital image processing. *Signal & Image Processing*, 6(2):63, 2015. 6

[4] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 279–288, 1993. 1

[5] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018. 5

[6] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 11–20, 1996. 1

[7] Ahmet M Eskicioglu and Paul S Fisher. Image quality measures and their performance. *IEEE Transactions on communications*, 43(12):2959–2965, 1995. 1

[8] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14346–14355, 2021. 4

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 5

[10] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, 1996. 1

[11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 5

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 5

[13] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010. 1, 2

[14] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017. 5, 7

[15] Jan J Koenderink. The structure of images. *Biological cybernetics*, 50(5):363–370, 1984. 1

[16] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, 1996. 1

[17] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 1

[18] 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. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 5

[19] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1, 3, 4

[20] Vishvak Murahari, Carlos E Jimenez, Runzhe Yang, and Karthik R Narasimhan. Datamux: Data multiplexing for neural networks. In *Advances in Neural Information Processing Systems*. 5

[21] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022. 4

[22] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021. 6

[23] Stanley Osher, Ronald Fedkiw, and K Piechor. Level set methods and dynamic implicit surfaces. *Appl. Mech. Rev.*, 57(3):B15–B15, 2004. 4

[24] Thrasyvoulos N Pappas, Robert J Safranek, and Junqing Chen. Perceptual criteria for image quality evaluation. *Handbook of image and video processing*, 110, 2000. 1

[25] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 1

[26] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 4, 5, 6

[27] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14335–14345, 2021. 4

[28] Paolo Sabella. A rendering algorithm for visualizing 3d scalar fields. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 51–58, 1988. 4

[29] Karsten Schlüns and Reinhard Klette. *Local and global integration of discrete vector fields*. Springer, 1997. 4

[30] Hamid R Sheikh and Alan C Bovik. Image information and visual quality. *IEEE Transactions on image processing*, 15(2):430–444, 2006. 1

[31] Harry Shum and Sing Bing Kang. Review of image-based rendering techniques. In *Visual Communications and Image Processing 2000*, volume 4067, pages 2–13. SPIE, 2000. 1

[32] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 5

[33] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 5

[34] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022. 4

[35] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *Advances in Neural Information Processing Systems*, 34:27171–27183, 2021. 6

[36] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020. 5

[37] Zhou Wang and Alan C Bovik. A universal image quality index. *IEEE signal processing letters*, 9(3):81–84, 2002. 1

[38] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 1, 2, 3

[39] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003. 1, 4

[40] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 4

[41] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, volume 41, pages 641–676. Wiley Online Library, 2022. 4

[42] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 6

[43] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33:2492–2502, 2020. 6

[44] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 4

[45] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 4

[46] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 5

A. Experimental Settings and Details

In this section, we present the experimental settings and details for reproducing the results. The main principle of our experimental setting is to fairly compare multiplex training and standard training for NeRF and the variants. Our experimental settings follows original papers to reproduce the baseline, unless we specify otherwise.

General Training Settings

DVGO

D-NeRF

Mip-NeRF

NeuS

B. Supplementary Experimental Results

We present the experimental results of neural rendering from sparse training images in Table 5.

Table 5. Evaluation on neural rendering of the sparse T&T-Truck scene.

Model	Training	20%			40%			60%			80%			100%		
		PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)
DVGO	Standard	14.67	0.527	0.597	18.99	0.639	0.425	21.07	0.683	0.396	21.74	0.686	0.386	22.01	0.704	0.386
DVGO	Multiplex	**18.99**	**0.618**	**0.405**	**21.68**	**0.704**	**0.356**	**22.11**	**0.719**	**0.347**	**22.38**	**0.722**	**0.351**	**22.44**	**0.730**	**0.338**

We present the experimental results of neural rendering from LLFF scenes in Table 6.

Table 6. Evaluation on neural rendering of the LLFF scenes.

Model	Training	Flower			Fortress			Horns			Room			Trex		
		PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)
DVGO	Standard	27.52	0.853	0.119	29.89	0.867	0.100	27.02	0.854	0.142	30.54	0.940	0.100	26.17	0.894	0.0978
DVGO	Multiplex	**27.57**	**0.856**	**0.114**	**30.14**	**0.879**	**0.0847**	**27.12**	**0.859**	**0.127**	**30.75**	**0.941**	**0.0960**	**26.36**	**0.896**	**0.0935**