Pacific Graphics 2023
R. Chaine, Z. Deng, and M. H. Kim
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 7

Robust Novel View Synthesis with Color Transform Module

S. M. Kim, C. Choi, H. Heo and Y. M. Kim†

Dept. of Electrical and Computer Engineering, Seoul National University, Korea

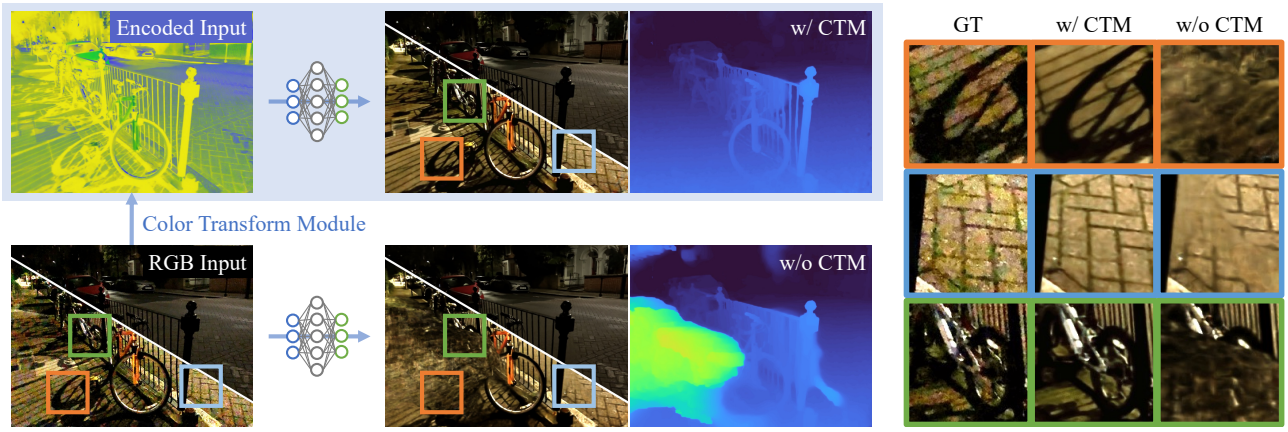


Figure 1: *In this work, we propose an easy-to-plug-in color transform module (CTM) for NeRF. We train NeRF in a transformed or encoded feature space with CTM (Blue box in the first row). By training NeRF with CTM, we can accurately reconstruct the geometry and radiance of the low-textured regions, which are known to be challenging when trained in RGB space. The dark RGB images in the lower triangle regions are enhanced for better visibility.*

Abstract

The advancements of the Neural Radiance Field (NeRF) and its variants have demonstrated remarkable capabilities in generating photo-realistic novel views from a small set of input images. While recent works suggest various techniques and model architectures that enhance speed or reconstruction quality, little attention is paid to exploring the RGB color space of input images. In this paper, we propose a universal color transform module that can maximally harness the captured evidence for the neural networks at hand. The color transform module utilizes an encoder-decoder framework that maps the RGB color space into a new latent space, enhancing the expressiveness of the input domain. We attach the encoder and the decoder at the input and output of a NeRF model of choice, respectively, and jointly optimize them to maintain the cycle consistency of the proposed transform, in addition to minimizing the reconstruction errors in the feature domain. Our comprehensive experiments demonstrate that the learned color space can significantly improve the quality of reconstructions compared to the conventional RGB representation. Its benefits are particularly pronounced in challenging scenarios characterized by low-light environments and scenes with low-textured regions. The proposed color transform pushes the boundaries of limitations in the input domain and offers a promising avenue for advancing the reconstruction capabilities of various neural representations. Source code is available at `https://github.com/sangminkim-99/ColorTransformModule`.

CCS Concepts

• ***Computing methodologies*** → *Reconstruction; Rendering;*

1. Introduction

Novel view synthesis aims to generate photo-realistic images of a scene from viewpoints that were not captured during the origi-

† Corresponding author

nal data collection process. Recently, significant progress has been made in this field with the introduction of the Neural Radiance Field (NeRF) [MST*21]. Inspired by its high-quality results, various extensions arose to model scenes that could not be handled in its original formulation, including unbounded scenes [BMV*22, ZRSK20, CKK23] or dynamic scenes [PCPMMN21, LWC*23].

The quality of reconstruction is highly dependent on the color variations of the input images. NeRF and aforementioned variants basically enforce *multi-view consistency* of color measurements and demonstrate stable performance for scenes with rich textures. When a scene contains blank walls or weak color variation due to insufficient lighting, on the other hand, the photometric consistency of input images does not provide sufficient evidence to converge to the correct geometry. Such a phenomenon, also known as shape-radiance ambiguity [ZRSK20], is inherent in the NeRF formulation, and low-textured scenes often suffer from blurry geometry. One can resolve the ambiguity with additional shape information, either from depth measurements [ALG*21] or estimated priors [RBM*22, WLR*21, GPL*22]. However, such information might not be accurate or always available.

We present a novel approach that explores alternative color domains beyond RGB. We take inspiration from existing color transforms, which are designed to separate lightness from hue information or increase expressiveness in high dynamic ranges. Instead of crafting a color space tailored to a given setting, we jointly optimize a scene-specific color transformation with NeRF reconstruction in the transformed domain. The transformation is optimized to disambiguate the low-textured area in RGB space by minimizing the reconstruction objective which enforces multi-view consistency in the feature domain. The transformed color space significantly enhances the quality of reconstruction, especially in challenging conditions where inputs exhibit a distinctive distribution in color.

The scene-specific color transformation can act in conjunction with rich recent works, improving the performance of the NeRF. The transformation is composed of an encoder-decoder framework, where the encoder maps RGB images to the feature domain, and the decoder generates RGB images from the transformed color space. The encoding and decoding transformation is parameterized as an MLP to represent any nonlinear function. We can incorporate the color transform into any method that employs reconstruction loss of individual pixel color values by prepending and appending the encoder and the decoder, respectively. We then evaluate the reconstruction loss in the transformed space instead of the conventional RGB space. In addition, we introduce a cycle consistency loss to ensure that the adapted transform preserves existing evidence.

We conducted experiments on multiple datasets, including Synthetic-NeRF [MST*21], RawNeRF [MHMB*22], and our custom datasets: Synthetic-NeRF-Dark and Low-Texture-Blender. Our extensive results show that the learned scene-specific transforms promote the integrated neural network module to better capture subtle texture variations or details in low-light and low-texture environments, and estimate faithful geometry. We demonstrate improvements in performance on a handful of NeRF settings and further integrate the transform with planar image alignment, as suggested in BARF [LMTL21]. The results indicate the robustness and applicability of our approach in computer vision tasks involving neural network modules trained with a reconstruction loss in color values.

In short, our contribution can be summarized as follows:

- We propose a latent space mapping inspired by color transforms, which can preserve delicate details of the input without extensive prior knowledge or hand-crafted operation.
- Our framework learns a low-level transform that increases the performance of reconstructing images for challenging conditions.
- We provide a dataset that allows quantitative evaluation under various measurement conditions such as low lighting or lack of detailed texture.
- The proposed transform can be easily integrated into any NeRF model, making it a versatile and adaptable solution for neural image reconstruction.

The universal performance enhancement covering various scenarios indicates that our module can be a simple yet effective addition to the powerful neural formulations for colored image reconstruction, especially for extreme conditions near to the sensor limits.

2. Related Work

NeRF volume is trained only with posed images and can be rendered into photo-realistic images for an arbitrary view with a volume rendering equation [KVH84]. With its superior novel-view synthesis results, NeRF initiated an enormous amount of subsequent works in various directions. In this section, we focus on the works that address the failure cases of NeRF. It is widely known that training NeRFs is often highly unstable because the optimization is highly non-linear and under-constrained. Even with the same configuration, the results vary on the initial random seed, and sometimes they fail to converge. As a partial remedy, the original NeRF [MST*21] employ the center cropping technique, while Barron et al. [BMT*21] utilize smoothly varying softplus activation instead of ReLU activation.

The ambiguity is pronounced when the image constraints are scarce, such as scenes with flat textures [WLR*21] or under low-light conditions [MHMB*22]. With limited image cues, the geometry remains as a cloud of "floaters", instead of high-density surface geometry. This is also referred to as shape-radiance ambiguity by [ZRSK20]. Several works employ additional regularization losses on the geometry, including TV loss [CXG*22] and distortion loss [BMV*22]. Others incorporate geometric priors of depth estimation [RBM*22, WLR*21] or Manhattan world assumption [GPL*22]. The reconstruction quality heavily relies on the accuracy of the estimated geometry.

Several works leverage latent space embedding to overcome the limitation of input space, but they often fail to faithfully conserve the low-level details. For instance, features of CLIP embeddings [RKH*21] can be leveraged to render plausible images with limited input [JTA21]. Similarly, PixelNeRF [YYTK21] utilizes a ResNet backbone [HZRS16] that has been pre-trained on ImageNet [DDS*09] to resolve ambiguity from a sparse set of input views. Neural features of depth estimation network can assist in representing specular objects [ZYW*23]. However, pre-trained

neural features from unrelated databases cause hallucinating effects [YPW23] due to the discrepancy between the latent space and the RGB color domain.

Instead, we propose low-level transforms on the input color space. Our module is less susceptible to deviation from ground truth, quickly adapting to unknown environments or sensor settings without additional priors. Our work is similar to the work by Mildenhall et al. [MHMB*22], which successfully excavates texture from raw HDR images. However, instead of relying on the enhanced sensor input, we apply transform adaptive to the current measurement and improve the sensitivity of the training objective. Our color transform module is inspired by the alternative color spaces from color science, including HSV [WS00], to align the color distance with human perception, or the Retinex model to disentangle the illumination with inherent reflectance. The relationship between the HSV color space and the Retinex model has been extensively studied [LLH*21, QJLW22, ZDZ*21], resulting in modifications to the V channel for improved low-light image enhancement, further refined in recent works [MML*22, WWZ*22]. In the context of NeRF optimization, we propose a scene-specific color transform instead of relying on conventional handcrafted color spaces.

The color transform module is attached to the input and output of the existing network, and jointly trained to adapt to the current measurement and improve the sensitivity of the training objective. It can be applied to more advanced network architecture, as long as the input and output are color images. For example, our module is compatible with a series of works employing MLP to represent the entire scene [MST*21,BMT*21,BMV*22]. Furthermore, it can also be incorporated into methods that employ explicit representations such as feature grids [SSC22, MESK22] or factorized representations [CXG*22] to accelerate the speed. Our module can be attached to any of these recent variations and stabilize the training in challenging cases.

3. Methods

3.1. Preliminaries: Neural Radiance Fields (NeRF)

NeRFs [MST*21] represent a 3D scene as a continuous neural implicit function, which is trained to match the input RGB images of a scene from various views. NeRFs typically consist of a multilayer perceptron (MLP) that takes in a 3D spatial coordinate $\mathbf{x}$ and a 2D direction $\mathbf{d}$ as input and outputs the color c and density σ at that point. The color $C(\mathbf{r})$ of the camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ is rendered by compositing the output color c with transmittance from a near bound t_n to a far bound t_f:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))c(\mathbf{r}(t),\mathbf{d})\,dt, \tag{1}$$

where

$$T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))\,ds\right). \tag{2}$$

In practice, the color of the pixel ray is numerically approximated by inferring color values at multiple sample points along the ray, which we can denote as $\mathcal{F}_\theta(\mathbf{r}) = \hat{C}(\mathbf{r})$.

The network parameter θ is optimized by minimizing the reconstruction loss, which is just a simple mean squared error of the rendered color $\hat{C}(\mathbf{r})$ and the ground truth color $C_{gt}(\mathbf{r})$ of the pixels:

$$\mathcal{L} = \frac{1}{|\mathcal{R}|}\sum_{\mathbf{r}\in\mathcal{R}} \left\|\hat{C}(\mathbf{r}) - C_{gt}(\mathbf{r})\right\|_2^2, \tag{3}$$

where $\mathcal{R}$ is a set of ray samples in each batch.

Instead of directly taking the 3D spatial coordinate $\mathbf{x}$ and 2D direction $\mathbf{d}$ as input, NeRF utilizes sinusoidal encoding of these coordinates as input to represent high-frequency details. While neural networks inherently prefer to regress low-frequency functions, Tancik et al. [TSM*20] demonstrated that the network can stably fit data that exhibits high-frequency details by applying sinusoidal functions to the input of the network. The set of sinusoidal functions maps the inputs to a high dimensional space and is referred to as positional encoding.

3.2. Color Transform Module

Our color transform module (CTM) augments the input and output of the existing NeRF formulation, allowing the reconstruction loss in Eq. (3) to be applied in the transformed space. Our encoder $f : [0,1]^3 \to \mathbb{R}^k$ converts a pixel in an RGB space $c \in [0,1]^3$ to a latent feature $l \in \mathbb{R}^k$, and the decoder transforms the latent pixel back to the conventional RGB space $g : \mathbb{R}^k \to [0,1]^3$. We illustrate the addition of the encoder and decoder modules in Figure 2(a). The transformation operates on individual pixels in the image, converting them into a feature image of the same size.

By modifying the final MLP layer of the NeRF, we can generate an output of dimension k; $\mathcal{F}^k_\theta(\mathbf{r}) \in \mathbb{R}^k$. This modified output tensor can be considered as residing in the transformed domain. Consequently, the reconstruction loss in the encoded space can be computed by comparing the output of $\mathcal{F}^k_\theta(\mathbf{r})$ against the encoded ground truth value:

$$\mathcal{L}_{enc} = \frac{1}{|\mathcal{R}|}\sum_{\mathbf{r}\in\mathcal{R}} \left\|\mathcal{F}^k_\theta(\mathbf{r}) - f(C_{gt}(\mathbf{r}))\right\|_2^2. \tag{4}$$

The rendered output $\mathcal{F}^k_\theta(\mathbf{r})$ in the latent space is transformed to the original RGB color space through the decoder $g(l)$. In conjunction with Eq. (4), we compute the reconstruction loss on the original domain after the decoder:

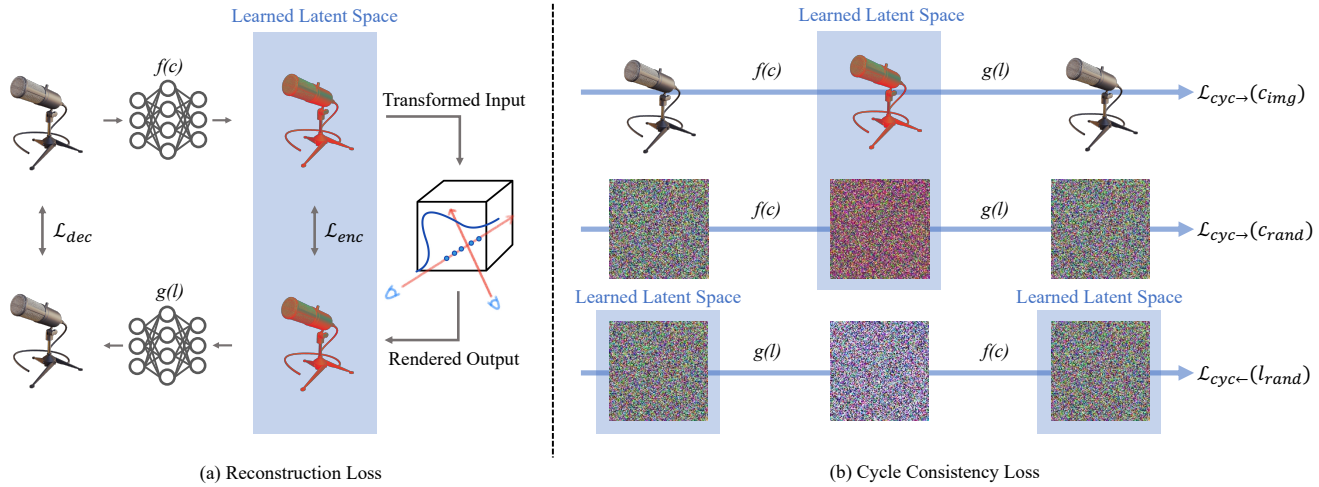
$$\mathcal{L}_{dec} = \frac{1}{|\mathcal{R}|}\sum_{\mathbf{r}\in\mathcal{R}} \left\|g(\mathcal{F}^k_\theta(\mathbf{r})) - C_{gt}(\mathbf{r})\right\|_2^2. \tag{5}$$

Therefore our reconstruction loss of the color discrepancy is

$$\mathcal{L}_{rec} = \mathcal{L}_{enc} + \mathcal{L}_{dec} \tag{6}$$

and we jointly optimize CTM with the original neural network $\mathcal{F}_\theta$. This joint optimization enables the MLP of CTM to find a scene-specific latent space, leading to enhanced scene reconstruction compared to relying solely on the RGB color space, especially for challenging scenes to reconstruct.

Our formulation is generally applicable to any network that imposes losses on the color differences. The architecture of color transform is ignorant of the formulation of the NeRF module $\mathcal{F}_\theta(\mathbf{r})$, and regards it as a black-box module. Indeed we can replace



(a) Reconstruction Loss (b) Cycle Consistency Loss

Figure 2: *Method overview: Our color transform module comprises an encoder $f(c)$ and a decoder $g(l)$. (a) With $f(c)$, we transform the colors of input images into latent values and train the radiance fields using reconstruction losses in both the transformed and color domains. (b) To ensure meaningful values in the latent domain, we incorporate cycle consistency loss as a regularization technique.*

NeRF with any other neural mapping function which produces an estimated color given a spatial coordinate, and increase the color sensitivity of the underlying neural network.

Sinusoidal color embedding. In conjunction with the shallow MLP layers, we augment sinusoidal functions of different frequencies to the input color, similar to the positional encoding in the spatial coordinates. Specifically, we augment the color value in k dimension with P frequency basis, resulting in $\gamma(\cdot) \in \mathbb{R}^{k(1+2P)}$:

$$\begin{aligned}\gamma(c) = [I, &\sin(2^0\pi), \sin(2^1\pi), \ldots, \sin(2^{P-1}\pi),\\ &\cos(2^0\pi), \cos(2^1\pi), \ldots \cos(2^{P-1}\pi)],\end{aligned} \tag{7}$$

where I refers to the identity function. The sinusoidal color embedding is applied for the inputs of both the encoder and decoder and further increases the flexibility of the mapping. The effect of the embedding is further investigated in the experiments.

3.3. Regularization with Cycle Consistency Loss

Naïvely optimizing over the reconstruction loss can significantly distort the color space in extreme cases. We ensure that the resulting transform conserves useful information by an indirect regularization with a cycle consistency loss, namely $(f \circ g) = (g \circ f) = \mathbf{I}$.

We define both forward and backward cycle consistency losses for the encoder and the decoder. The forward cycle consistency loss is computed as:

$$\mathcal{L}_{cyc\rightarrow} = \frac{1}{N} \sum_{c \in \mathbb{R}^3} \left\| c - g(f(c)) \right\|_2^2, \tag{8}$$

where N is the number of samples in each batch. Similarly, the backward cycle consistency loss is calculated as:

$$\mathcal{L}_{cyc\leftarrow} = \frac{1}{N} \sum_{l \in \mathbb{R}^k} \left\| l - f(g(l)) \right\|_2^2. \tag{9}$$

We incorporate the two-way cycle consistency for random images to assist the forward consistency of the input image as shown in Figure 2(b):

$$\mathcal{L}_{cyc} = \mathcal{L}_{cyc\rightarrow}(c_{img}) + \mathcal{L}_{cyc\rightarrow}(c_{rand}) + \mathcal{L}_{cyc\leftarrow}(l_{rand}), \tag{10}$$

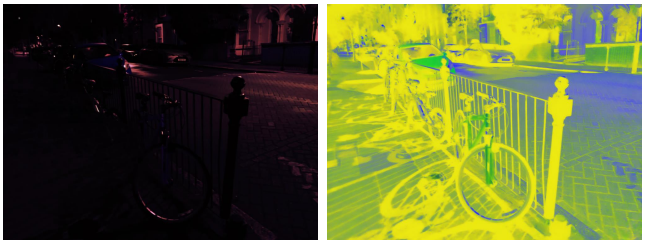
where c_{rand} and l_{rand} are randomly sampled from a uniform distribution, and c_{img} denotes the color values of the training image. Learning two-way conservative mapping with random pixels regularizes the overall color transformation regardless of the imbalance in the input images.

The key component of the cycle consistency loss $\mathcal{L}_{cyc}$ is $\mathcal{L}_{cyc\rightarrow}(c_{img})$, which encourages a mapping that preserves the pixel colors in the original images. The color transform is designed to ease the training of the neural network by manipulating distances of the current colors existing in the scene. Focusing solely on the reconstruction loss in Eq. (6) can concentrate the encoded values to be within proximity when the original color distribution is clustered, ignoring imperceptible subtle details. $\mathcal{L}_{cyc\rightarrow}(c_{img})$ serves as a regularization term, encouraging the subtle differences to be separated within the latent space. The loss is especially beneficial when the input images suffer from extreme light conditions as shown in Figure 3.

Usually the forward cycle consistency loss in Eq. (8) is sufficient to preserve the information in the training images, and adding the backward cycle consistency loss may sometimes lead to a slight performance decrease. However, the backward cycle consistency is crucial to avoid fitting the latent space to a small set of dominant colors. For example, previous works stabilize the training by using a random background color [MESK22, SSC22]. Combined with our color transform, the decoder loss $\mathcal{L}_{dec}$ encourages the decoder to map the latent values to match the random background colors during initial iterations. Our backward cycle loss in Eq. (9) remedies the singularity in the mapping, especially in extremely dark regions.

Methods	Synthetic-NeRF [MST*21]			RawNeRF [MHMB*22]			Synthetic-NeRF-Dark			Low-Texture-Blender		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
DVGO [SSC22]	31.47	0.9566	0.0280	31.67	0.8261	0.1994	30.35	0.7935	0.2645	36.34	0.9610	0.0503
DVGO w/ CTM	31.12	0.9549	0.0289	33.38	0.8752	0.1585	34.95	0.9181	0.1396	36.99	0.9652	0.0395
NeRF [MST*21]	28.60	0.9261	0.0601	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	37.94	0.9628	0.0458
NeRF w/ CTM	28.32	0.9259	0.0606	33.60	0.8736	0.1539	38.86	0.9616	0.0494	38.13	0.9630	0.0457

Table 1: *Quantitative results of NeRF optimization. N.C. indicates that the method does not converge for that particular dataset.*



(a) w/o $\mathcal{L}_{cyc\rightarrow}(c_{img})$ (b) w/ $\mathcal{L}_{cyc\rightarrow}(c_{img})$

Figure 3: *Volume rendering results in encoded space. (a) Without $\mathcal{L}_{cyc\rightarrow}(c_{img})$, the optimization process is dominated by the reconstruction loss, resulting in a shrinking encoding space. (b) With $\mathcal{L}_{cyc\rightarrow}(c_{img})$, the encoding space expands, allowing the neural network to capture subtle details more effectively.*

In summary, the final loss term for optimization is

$$\mathcal{L} = \mathcal{L}_{rec} + \mathcal{L}_{cyc}. \tag{11}$$

4. Experiments

We first demonstrate the effectiveness of the learned color transform for optimizing NeRF (Section 4.1) in various NeRF models [MST*21,SSC22]. Additionally, we apply the same module to a slightly different problem of 2D planar image alignment introduced by BARF [LMTL21] to show the versatility of our module beyond NeRF (Section 4.2).

Implementation details. We maintain the dimensionality of the original RGB color space by setting the latent dimension to be $k = 3$ and treat our transform as a color transform applied to individual pixel values. We deliberately keep the MLP architecture as shallow as possible for efficient inference. Specifically, we utilize a single-layer MLP with 64 channels, using float32 datatype. To capture subtle variations in the color mapping, we use sinusoidal embedding with four frequency bases inspired by a positional encoding for NeRF. The network activation function is chosen as ReLU, and a sigmoid activation is applied to the output to ensure that both the encoder and decoder operate within a bounded $\mathbb{R}^3$ space. To ensure the modularity of our method, we utilize a separate Adam optimizer [KB15] for our module instead of integrating it with the NeRF optimizer. We set the learning rate to 5×10^{-4} for NeRF optimization and 5×10^{-3} for the planar image alignment. We train all NeRF models with 100k iterations. We preserve the setting of their original implementations for the other hyperparameters and training details [MST*21, SSC22, LMTL21].

4.1. Optimizing NeRF

We demonstrate the effectiveness of our module for robust novel view synthesis through experiments on various NeRF models. NeRF networks use a set of posed images as input and train a volume that outputs the scene geometry (density) and radiance. The color transform module, consisting of the color encoder and decoder, is integrated into the colors of input and output images to enable optimization in the transformed color space. While our module can be attached to any works that are trained with the color discrepancy, we demonstrate the results on two representative NeRF networks, the original NeRF architecture [MST*21] utilizing a single MLP for a scene, and DVGO [SSC22] employing a feature-grid structure for a faster network.

We do not compare against RawNeRF [MHMB*22] due to the difference in input modality: our inputs are LDR images, while RawNeRF uses HDR images. Additionally, our enhancement solely is based on the image measurements, distinguishing it from NeRF variants [YYTK21, ZYW*23] using feature-space conversions. Extracting features incorporates priors from training data and we deliberately avoid such dataset biases.

4.1.1. Dataset

We extensively evaluate our approach in four datasets. Two of them (Synthetic-NeRF and RawNeRF) are widely-used datasets to benchmark the performance of view synthesis. We also contribute two newly collected datasets that present new challenges that were not covered by existing datasets.

- Synthetic-NeRF [MST*21]: This widely-used synthetic dataset contains small objects inside a bounding box with rich textures and various material properties.
- RawNeRF [MHMB*22]: This dataset comprises high-resolution real images with raw data. We selected nine scenes captured in low-light conditions for our experiments.
- Synthetic-NeRF-Dark: This dataset is a low-light version of the Synthetic-NeRF dataset. We re-rendered each scene of Synthetic-NeRF amidst low-light conditions.
- Low-Texture-Blender: We also introduce a new dataset with low-textured scenes. It consists of seven scenes, each containing 49 images. We use one-eighth of the images as test images, while we use the remaining images to train models.

In our evaluation, we largely adhere to the original setup of each method. However, in the RawNeRF dataset, we use a random background instead of the original black background. We need to avoid black background for dark scenes as the network can introduce holes in the trained volume. Similarly, in the Synthetic-NeRF-Dark dataset, we replace the black background with a random background to suppress artifacts in extremely dark scenes.

4.1.2. Results

The quantitative results of novel view synthesis are presented in Table 1, and sample qualitative results are contained in Figure 4. We observe that CTM enhances the performance in most datasets, especially for challenging environment settings. We also perform on par with other works on the Synthetic-NeRF dataset. When we measure the variation of results trained with different initial seeds, we observe that the performance gaps for all metrics are within 10% of the standard deviation, indicating the results are almost the same for the ordinary setting (results are included in the appendix). Note that CTM also cannot take advantage of the pre-defined background color. NeRF [MST*21] additionally suppresses density for regions matching the selected background color, and the original NeRF in the RGB space can easily remove the background density. On the other hand, our color transform adapts the color into the flexible encoding space and does not rely on the synthetic background setup.

The effectiveness of our CTM is prominent in dark scenes (RawNeRF, Synthetic-NeRF-Dark). With a limited color range in the input image, the vanilla NeRF even does not converge. Our module not only stabilizes the convergence but also automatically finds an adequate mapping to increase the performance. We also noticed that the feature grid of DVGO can sometimes fail to converge despite its fast convergence. NeRF models the entire scene with a single network, and benefits from the inherent smooth interpolation to fill the ambiguous volumes. However, individual feature grid points observe local neighborhoods and can significantly deteriorate, as shown in Figure 4.

While the quantitative results demonstrate subtle performance enhancement for the Low-Texture-Blender, the qualitative results in Figure 4 shows that CTM can capture more accurate geometry, especially with DVGO. Low-texture-Blender contains walls with almost flat textures, and the resulting color differences might not be significant. Even in the worst-case scenario where the MLP emits only one color, the difference in performance metric is not substantial. As a result, incorrect geometry does not necessarily lead to incorrect colors as long as the network predicts the dominant color. CTM, however, correctly interprets weak signals in color and shows remarkable improvement in geometry (further details are included in the appendix). CTM, therefore, holds great potential for various downstream tasks that utilize geometric context.

Comparison between other color spaces. We compare our scene-specific color transformation against existing handcrafted color maps. We replace our method with transformations based on HLS, HSV, Lab, and Luv color spaces [WS00]. Additionally, we apply a warping technique to transform the cylindrical positions of the Hue channel into Cartesian coordinates (denoted as $^{\text{cart}}$). Since the Hue channel of HLS and HSV is defined as an angle in a cylinder, there is a discontinuity at zero, which results in artifacts, as shown in Figure 5. Our Cartesian variants effectively transform the discontinuous angular space into a continuous domain where the neural network can be efficiently trained. In Table 2, we present the quantitative results of NeRF using various color transformations on the RawNeRF dataset and verify that our method yields the best performance in a challenging scenario. Interestingly, our experiments reveal that color spaces such as HLS and HSV outperform RGB. This observation supports our initial motivation that there might exist a more suitable color space for optimizing NeRF models compared to the traditional RGB color space.

Color Space	PSNR	SSIM	LPIPS
RGB	31.67	0.8261	0.1994
HLS	31.85	0.8303	<u>0.1610</u>
HLS$^{\text{cart}}$	32.63	0.8515	0.1645
HSV	32.31	0.8427	0.1615
HSV$^{\text{cart}}$	<u>32.86</u>	<u>0.8579</u>	0.1698
Lab	28.34	0.7359	0.2574
Luv	28.50	0.7346	0.2564
Ours	**33.38**	**0.8752**	**0.1585**

Table 2: *Quantitative results in various color spaces for RawNeRF dataset. HLS^{cart} and HSV^{cart} denote the warped color space into Cartesian coordinate.*

Methods	warping error (↓)	patch PSNR (↑)
No PE	0.3559	21.87
No PE w/ CTM	**0.2156**	22.47
Naïve PE	0.3981	19.89
Naïve PE w/ CTM	0.3721	20.13
BARF [LMTL21]	0.2922	26.22
BARF w/ CTM	0.2357	**29.24**

Table 3: *Quantitative results of planar image alignment. We report the mean of 10 random seeds.*

4.2. Planar Image Alignment

As another problem to apply the color transformation, we present results on planar image alignment, as suggested in BARF [LMTL21]. Given five patches of an image cropped with random homography parameters, the problem is to optimize for the warping homography, and the patch is initialized from the center position. The image is represented with a neural network that maps pixel coordinates to corresponding color values. BARF presented improved alignment results by incorporating higher frequencies of positional encoding for the later stages of optimization. The results are compared against no positional encoding, naïve positional encoding, and the scheduled set of frequencies of BARF.

Similar to the NeRF setting, we augment the encoder and decoder to map the RGB color and jointly optimize for the homography parameters and the color transform. Table 3 demonstrates that CTM enhances the performance of all the approaches, especially in

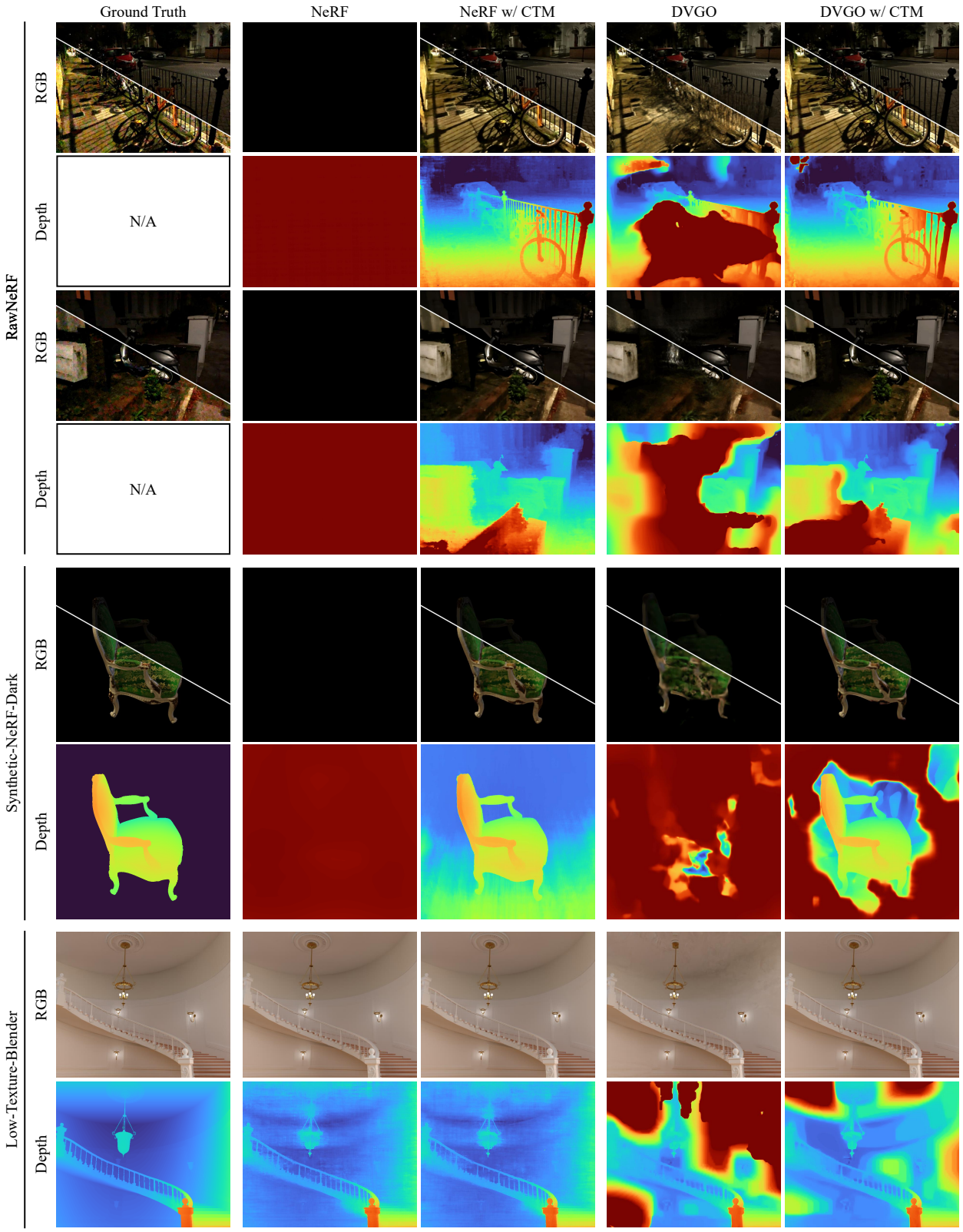


Figure 4: *Qualitative results of optimizing NeRF in challenging environments, including bikes, scooter, chair, and staircase. The first row showcases the rendered images of each scene, with the lower triangle enhanced for better visibility in dark regions. The second row presents the corresponding depth images.*

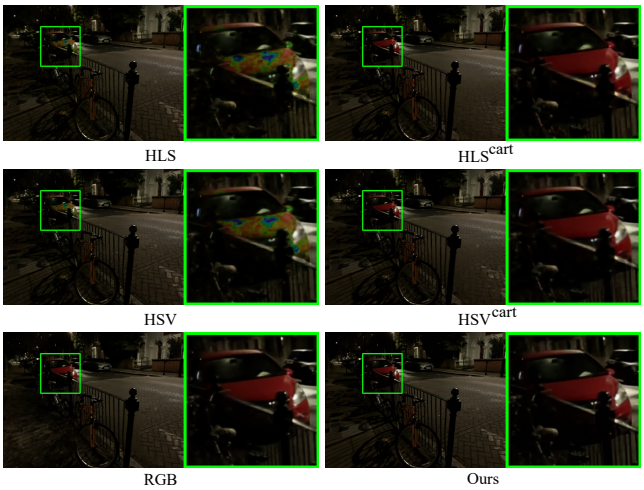


Figure 5: *Qualitative results in various color spaces for RawNeRF dataset. The warping technique effectively reduces artifacts in the red car, denoted as* ^{cart}.

BARF. One may attribute the improvement to the increased number of parameters with our module. However, the deeper layers of the module effectively encode higher-frequency positional information [ZRLL22], and it has been reported that higher-level positional encoding can lead to erroneous warping homography [LMTL21]. Nonetheless, CTM consistently improves the performance in terms of both PSNR and the warping error. The results indicate that the latent embedding of CTM adapts the images for more distinctive features, thereby aiding in accurate warping estimation. In Figure 6, we present the qualitative results for various images. The correct warping with CTM generates crisp images on the second row. CTM plays a crucial role in refining the warping process, particularly in cases where the original warping exhibits slight misalignment, as shown with BARF on the cat and squirrel image.

4.3. Ablation Study

We conduct ablation studies to validate our design choices. First, we analyze the impact of individual loss terms in Table 4. Excluding the $\mathcal{L}_{dec}$ results in a decrease in performance, highlighting the importance of incorporating the photometric loss in both domains to fully extract valuable information. It is worth noting that even without the loss in the RGB domain, our approach still outperforms the naïve RGB approach, emphasizing the effectiveness of the applied color transformation. Additionally, we find a significant performance drop when excluding the $\mathcal{L}_{cyc\rightarrow}(c_{img})$. As shown in Figure 3, this term plays a crucial role in achieving a more scene-specific color transformation by enhancing subtle details in the RGB space. We also observe a decrease of performance without $\mathcal{L}_{cyc\leftarrow}(l_{rand})$. This supports the effect of regularization terms as illustrated in Section 3.3.

Although the inclusion of $\mathcal{L}_{cyc\rightarrow}(c_{rand})$ results in a slight decrease in performance, the findings presented in Figure 7 underscore the significance of this term. In complex scenes with thin handrails or fog-like lighting, the volume rendering process of NeRF leads to the blending of foreground and background features. These blended latent features may not be represented in the encoded space of the training images, resulting in inaccurate color values during decoding. By incorporating $\mathcal{L}_{cyc\rightarrow}(c_{rand})$, we ensure the preservation of cycle consistency for these novel colors, effectively mitigating artifacts in the rendered images. In Figure 7, we can see how the edge of a traffic sign turns black and the handrail, along with the gap between them, exhibits a reddish color while blending with the background.

	PSNR	SSIM	LPIPS
w/o CTM	31.67	0.8261	0.1994
w/o $\mathcal{L}_{dec}$	32.96	0.8689	0.1634
w/o $\mathcal{L}_{cyc\rightarrow}(c_{img})$	31.48	0.7664	0.1935
w/o $\mathcal{L}_{cyc\rightarrow}(c_{rand})$	**33.40**	0.8751	**0.1567**
w/o $\mathcal{L}_{cyc\leftarrow}(l_{rand})$	33.31	0.8721	0.1574
Ours (full)	33.38	**0.8752**	0.1585

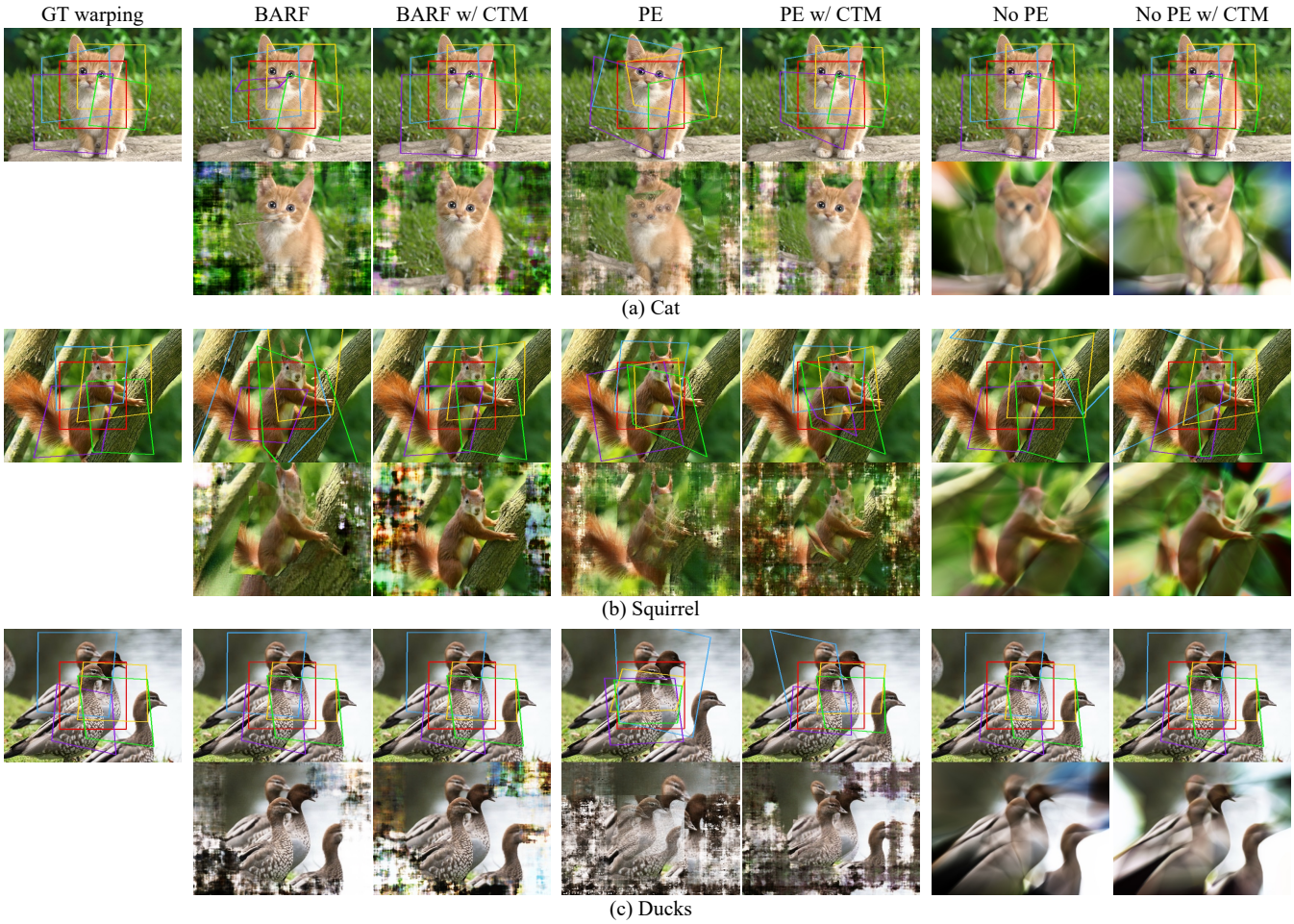
Table 4: *Ablation study on the loss term in the RawNeRF dataset.*

		PSNR	SSIM	LPIPS
Sinusoidal embedding level P	0	33.32	0.8699	0.1573
	2	33.37	0.8750	**0.1568**
	4 (Ours)	**33.38**	**0.8752**	0.1585
	6	31.53	0.8307	0.1875
	8	30.86	0.8172	0.2007
	10	26.93	0.7036	0.2747
Latent dimension k	1	32.75	0.8676	0.1610
	2	33.37	0.8735	**0.1570**
	3 (Ours)	**33.38**	**0.8752**	0.1585
	4	33.27	0.8733	0.1597
	5	33.07	0.8678	0.1632

Table 5: *Ablation study on the sinusoidal color embedding level and latent dimension in the RawNeRF dataset.*

Table 5 outlines additional experiments on our parameter choices. We explore different values for the level of sinusoidal color embedding P and the latent dimension k in our feature domain. We find that when P exceeds 4, the rendering results experience a significant degradation. We use $P = 4$ to balance the complexity of mapping and avoid significant distortion. Regarding the latent space dimension k, we observe no significant difference except a slight performance decrease for $k = 1$. However, when k is less than 3, the rendered outputs exhibit noticeable color discrepancies compared to the ground truth images. This can be clearly seen in Figure 8, where the rendered images near the tree appear to be achromatic, lacking the expected color fidelity. Despite of the reduced latent space dimension, however, CTM can reconstruct high-fidelity depth maps. It indicates that CTM effectively finds the embedding that captures important variations within the scene.

Table 6 demonstrates the impact of varying loss functions. While changing the loss function into L1 can enhance the performance of



(a) Cat

(b) Squirrel

(c) Ducks

Figure 6: *Qualitative results of planar image alignment. The first row of each scene shows the estimated warping for each patch. The optimized 2D neural image is presented in the second row.*

(a) Ground Truth (b) w/o $\mathcal{L}_{cyc\rightarrow}(c_{rand})$ (c) w/ $\mathcal{L}_{cyc\rightarrow}(c_{rand})$

Figure 7: *Rendering output on highly blended region with the $\mathcal{L}_{cyc\rightarrow}(c_{rand})$ ablated. The top image exhibits artifacts at the top edge of the traffic sign, while the bottom image highlights reddish artifacts near the thin handrail.*

	PSNR	SSIM	LPIPS
MSE w/o CTM	31.67	0.8261	0.1994
L1 w/o CTM	32.21	0.8548	0.1701
MSE w/ CTM	33.38	0.8752	0.1585
L1 w/ CTM	33.10	0.8720	0.1540

Table 6: *Ablation study on the different loss functions in the RawNeRF dataset.*

NeRF, it can also be combined with our CTM and further improve quality.

5. Conclusion

We propose a scene-specific color transformation attached to existing neural representations processing images. By incorporating the color encoder and decoder within the NeRF pipeline, we were able to enhance the reconstruction quality in scenes with low textures

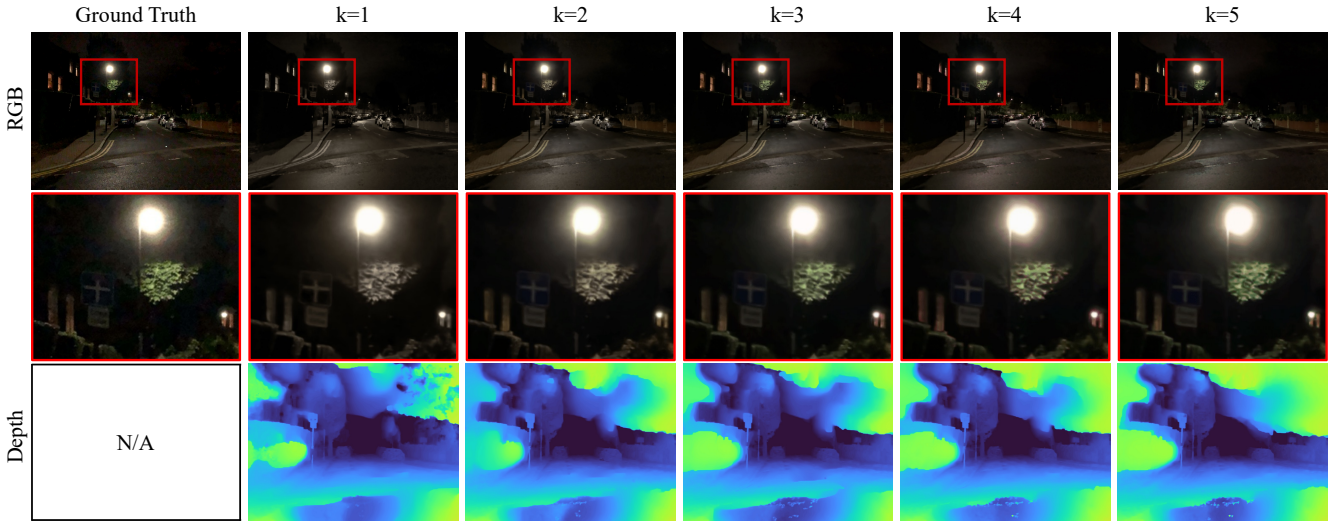


Figure 8: *Qualitative results of ablation study on the latent dimension k. Rendered output images and corresponding depth maps are presented. When $k < 3$, the rendered image, particularly the tree under the lamp, appears achromatic.*

and low-light environments. The scene-specific color transformation is jointly trained to resolve ambiguities and subtle variations in the measurements. Our extensive results show that it can improve the reconstruction quality in challenging environments. The module is versatile and can be combined to enhance existing image synthesis techniques, offering a valuable tool to overcome the challenges associated with the input domain. Future research directions include integrating it with other computer vision tasks or other datasets. We anticipate it can readily be adopted to enhance the performance of various vision-based tasks, for example, indoor visual SLAM, where low-textured scenes are prevalent.

Acknowledgements This work was partly supported by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (P0012746, HRD Program for Industrial Innovation), and the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2023.

References

[ALG*21] ATTAL B., LAIDLAW E., GOKASLAN A., KIM C., RICHARDT C., TOMPKIN J., O'TOOLE M.: Törf: Time-of-flight radiance fields for dynamic scene view synthesis. *Advances in neural information processing systems 34* (2021), 26289–26301. 2

[BMT*21] BARRON J. T., MILDENHALL B., TANCIK M., HEDMAN P., MARTIN-BRUALLA R., SRINIVASAN P. P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 5855–5864. 2, 3

[BMV*22] BARRON J. T., MILDENHALL B., VERBIN D., SRINIVASAN P. P., HEDMAN P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 5470–5479. 2, 3

[CKK23] CHOI C., KIM S. M., KIM Y. M.: Balanced spherical grid for egocentric view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 16590–16599. 2

[CXG*22] CHEN A., XU Z., GEIGER A., YU J., SU H.: Tensorf: Tensorial radiance fields. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII* (2022), Springer, pp. 333–350. 2, 3

[DDS*09] DENG J., DONG W., SOCHER R., LI L.-J., LI K., FEI-FEI L.: Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (2009), Ieee, pp. 248–255. 2

[GPL*22] GUO H., PENG S., LIN H., WANG Q., ZHANG G., BAO H., ZHOU X.: Neural 3d scene reconstruction with the manhattan-world assumption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 5511–5520. 2

[HZRS16] HE K., ZHANG X., REN S., SUN J.: Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 770–778. 2

[JTA21] JAIN A., TANCIK M., ABBEEL P.: Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 5885–5894. 2

[KB15] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, Conference Track Proceedings* (2015). 5

[KVH84] KAJIYA J., VON HERZEN B.: Ray tracing volume densities. *ACM SIGGRAPH Computer Graphics 18* (1984), 165–174. 2

[LLH*21] LIU S., LONG W., HE L., LI Y., DING W.: Retinex-based fast algorithm for low-light image enhancement. *Entropy 23*, 6 (2021), 746. 3

[LMTL21] LIN C.-H., MA W.-C., TORRALBA A., LUCEY S.: Barf: Bundle-adjusting neural radiance fields. In *IEEE International Conference on Computer Vision* (2021). 2, 5, 6, 8

[LWC*23] LI Z., WANG Q., COLE F., TUCKER R., SNAVELY N.: Dynibar: Neural dynamic image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 4273–4284. 2

[MESK22] MÜLLER T., EVANS A., SCHIED C., KELLER A.: Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG) 41*, 4 (2022), 1–15. 3, 4

[MHMB*22] MILDENHALL B., HEDMAN P., MARTIN-BRUALLA R., SRINIVASAN P. P., BARRON J. T.: Nerf in the dark: High dynamic range view synthesis from noisy raw images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 16190–16199. 2, 3, 5

[MML*22] MA L., MA T., LIU R., FAN X., LUO Z.: Toward fast, flexible, and robust low-light image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 5637–5646. 3

[MSOC*19] MILDENHALL B., SRINIVASAN P. P., ORTIZ-CAYON R., KALANTARI N. K., RAMAMOORTHI R., NG R., KAR A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)* (2019). 11

[MST*21] MILDENHALL B., SRINIVASAN P. P., TANCIK M., BARRON J. T., RAMAMOORTHI R., NG R.: Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM 65*, 1 (2021), 99–106. 2, 3, 5, 6, 11

[PCPMMN21] PUMAROLA A., CORONA E., PONS-MOLL G., MORENO-NOGUER F.: D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 10318–10327. 2

[QJLW22] QUAN J., JIN H., LI Z., WEN Z.: Low illumination image enhancement algorithm based on hsv-rnet. In *2022 7th International Conference on Image, Vision and Computing* (2022), pp. 531–536. 3

[RBM*22] ROESSLE B., BARRON J. T., MILDENHALL B., SRINIVASAN P. P., NIESSNER M.: Dense depth priors for neural radiance fields from sparse input views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 12892–12901. 2

[RKH*21] RADFORD A., KIM J. W., HALLACY C., RAMESH A., GOH G., AGARWAL S., SASTRY G., ASKELL A., MISHKIN P., CLARK J., ET AL.: Learning transferable visual models from natural language supervision. In *International conference on machine learning* (2021), PMLR, pp. 8748–8763. 2

[SSC22] SUN C., SUN M., CHEN H.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022). 3, 4, 5, 12

[TSM*20] TANCIK M., SRINIVASAN P., MILDENHALL B., FRIDOVICH-KEIL S., RAGHAVAN N., SINGHAL U., RAMAMOORTHI R., BARRON J., NG R.: Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems 33* (2020), 7537–7547. 3

[WLR*21] WEI Y., LIU S., RAO Y., ZHAO W., LU J., ZHOU J.: Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021). 2

[WS00] WYSZECKI G., STILES W. S.: *Color science: concepts and methods, quantitative data and formulae*, vol. 40. John wiley & sons, 2000. 3, 6

[WWZ*22] WU W., WENG J., ZHANG P., WANG X., YANG W., JIANG J.: Uretinex-net: Retinex-based deep unfolding network for low-light image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 5901–5910. 3

[YPW23] YANG J., PAVONE M., WANG Y.: Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023). 3

[YYTK21] YU A., YE V., TANCIK M., KANAZAWA A.: pixelNeRF: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021). 2, 5

[ZDZ*21] ZHANG Y., DI X., ZHANG B., JI R., WANG C.: Better than reference in low-light image enhancement: conditional re-enhancement network. *IEEE Transactions on Image Processing 31* (2021), 759–772. 3

[ZRLL22] ZHENG J., RAMASINGHE S., LI X., LUCEY S.: Trading positional complexity vs deepness in coordinate networks. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII* (2022), Springer, pp. 144–160. 8

[ZRSK20] ZHANG K., RIEGLER G., SNAVELY N., KOLTUN V.: Nerf++: Analyzing and improving neural radiance fields. *arXiv:2010.07492* (2020). 2

[ZYW*23] ZHU B., YANG Y., WANG X., ZHENG Y., GUIBAS L.: Vdn-nerf: Resolving shape-radiance ambiguity via view-dependence normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 35–45. 2, 5

Appendix A: Dataset and Implementation Details

Dataset

We introduce two newly collected datasets, namely Synthetic-NeRF-Dark and Low-Texture-Blender. Synthetic-NeRF-Dark and Low-Texture-Blender are challenging datasets that contain low-light environments and large-portion of low-textured areas, respectively. For the Synthetic-NeRF-Dark dataset, we render the identical 3D models from Synthetic-NeRF [MST*21] in the weak light environment. We render images from the same 100 and 200 camera viewpoints on the hemisphere for trainset and testset in the original Synthetic-NeRF dataset. For the Low-Texture-Blender dataset, we collect open-sourced scenes which contain low-textured areas (e.g. white walls). We render 49 images from forward-facing camera trajectory following LLFF [MSOC*19]. Samples from the proposed datasets are demonstrated in Figure 9. The sources used to generate the Low-Texture-Blender dataset and the test images for the planar image alignment are provided below.

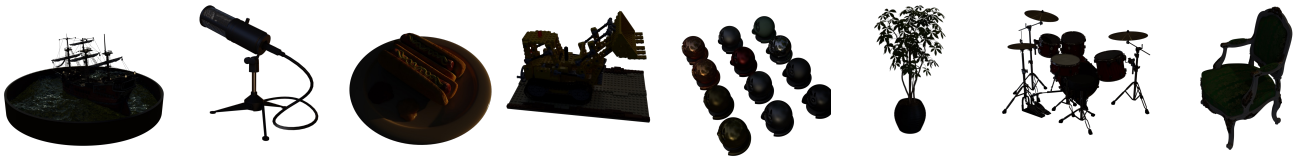
```
# Low-Texture-Blender
- Bathroom by bobal57 (CC-0)
http://www.blendswap.com/blends/view/21307
- Bathroom-2 by nacimus (CC-BY)
https://blendswap.com/blend/12584
- Bathroom-3 by imperfectino (CC-0)
https://blendswap.com/blend/29175
- Dark-Kitchen by Vladoffsky (CC-0)
https://blendswap.com/blend/19116
- Island-Kitchen by Matteo Pascale (RF)
https://www.blenderkit.com/asset-gallery-detail/c6d4517a-853d-4a2b
      -a312-55cb36a3a826/
- Low-poly-deer by Spine69 (CC-0)
https://blendswap.com/blend/26863
- Staircase by blenderjunky (CC-0)
https://blendswap.com/blend/11660

# Planar Image Alignment
- Squirrel by Andhoj via Pixabay
https://pixabay.com/photos/nature-rodent-squirrel-mammal-7997402/
- Ducks by pen_ash via Pixabay
https://pixabay.com/photos/australian-wood-ducks-ducks-birds
      -8012590/
```

Implementation Details

For the optimization of NeRF, we set the following values for the hyperparameters: $w_{enc} = 1$, $w_{dec} = 1$, $w_{cyc\rightarrow}(c_{img}) = 10000$, $w_{cyc\rightarrow}(c_{rand}) = 10$, and $w_{cyc\leftarrow}(l_{rand}) = 100$. For planar image alignment, we use $w_{enc} = 1$, $w_{dec} = 0.1$, $w_{cyc\rightarrow}(c_{img}) = 100$, $w_{cyc\rightarrow}(c_{rand}) = 0$, and $w_{cyc\leftarrow}(l_{rand}) = 0$.

During our experiments on the Synthetic-NeRF-Dark dataset,



(a) Synthetic-NeRF-Dark

(b) Low-Texture-Blender

Figure 9: *We demonstrate samples from our newly introduced (a) Synthetic-NeRF-Dark and (b) Low-Texture-Blender dataset.*

Methods	PSNR mean±std	SSIM mean±std	LPIPS mean±std
DVGO	31.80 ± 3.39	0.9554 ± 0.0345	0.0364 ± 0.0351
DVGO w/ CTM	31.59 ± 3.36	0.9547 ± 0.0343	0.0365 ± 0.0342

Table 7: *Quantitative results of DVGO on the Synthetic-NeRF dataset with varying random seeds.*

we observed that DVGO [SSC22] encounters difficulties with "floaters" during the coarse optimization stage. These floaters cause significant challenges in reconstructing the scene accurately. As a result, we decided to skip the coarse stage and instead focus on training the fine stage for a total of $30k$ steps similar to RawNeRF setting. This adjustment allowed us to mitigate the issues caused by floaters and achieve more satisfactory results in reconstructing the scene.

Appendix B: Performance of CTM in Synthetic-NeRF Dataset

We conducted an additional experiment on the Synthetic-NeRF dataset to investigate the impact of CTM on performance in high-textured scenarios. We utilized DVGO instead of NeRF due to its faster convergence properties. By varying the random seed from 0 to 3, we calculated the mean and standard deviation for different metrics, as shown in Table 7. Notably, the differences in all metrics were found to be smaller than 10% of their corresponding standard deviations, indicating that the application of CTM does not degrade the performance of Novel View Synthesis in Synthetic-NeRF dataset.

Appendix C: Performance Boost in the Low-Textured Scene

The performance improvement in low-textured scenes follows the same underlying principle with low-light scenarios. In essence, CTM amplifies subtle details that are already present but lack distinction in the original color space. We demonstrate this effect using an image from the *Staircase* scene (the leftmost image in Figure 9). Upon applying our CTM, the total image variance increased from 859.6 to 871.7. Additionally, when handpicking a 100×100 patch from a low-textured region, the variance increased from 256.5 to 346.2. These results indicate how our CTM enhances variations in low-textured scenes, thereby improving the quality of NeRF reconstructions.

Appendix D: Additional Results

In this section, we provide per-scene results of our experiments to further support the effectiveness of CTM in challenging scenarios. Table 11-19 demonstrate the performance increase applying CTM in challenging scenarios.

Methods	chair	drums	ficus	hotdog	lego	materials	mic	ship	mean
NeRF	29.90	23.84	27.84	33.56	29.54	27.25	30.18	26.65	28.60
NeRF w/ CTM	29.62	23.74	27.03	33.38	29.16	27.54	29.50	26.54	28.32
DVGO	33.89	25.05	32.48	35.88	33.87	29.12	32.95	28.51	31.47
DVGO w/ CTM	33.19	24.91	31.51	35.73	33.14	28.96	32.86	28.63	31.12

Table 8: *PSNR*(↑) *on Synthetic-NeRF dataset*

Methods	chair	drums	ficus	hotdog	lego	materials	mic	ship	mean
NeRF	0.9434	0.9057	0.9452	0.9623	0.9359	0.9292	0.9651	0.8221	0.9261
NeRF w/ CTM	0.9410	0.9069	0.9384	0.9630	0.9351	0.9339	0.9641	0.8249	0.9259
DVGO	0.9780	0.9294	0.9789	0.9800	0.9765	0.9507	0.9825	0.8765	0.9566
DVGO w/ CTM	0.9759	0.9278	0.9753	0.9799	0.9739	0.9493	0.9824	0.8751	0.9549

Table 9: *SSIM*(↑) *on Synthetic-NeRF dataset*

Methods	chair	drums	ficus	hotdog	lego	materials	mic	ship	mean
NeRF	0.0552	0.0849	0.0346	0.0369	0.0371	0.0409	0.0364	0.1544	0.0601
NeRF w/ CTM	0.0564	0.0817	0.0422	0.0370	0.0399	0.0368	0.0388	0.1515	0.0606
DVGO	0.0135	0.0528	0.0121	0.0135	0.0118	0.0200	0.0114	0.0886	0.0280
DVGO w/ CTM	0.0163	0.0535	0.0144	0.0135	0.0131	0.0214	0.0115	0.0871	0.0289

Table 10: *LPIPS*(↓) *on Synthetic-NeRF dataset*

Methods	bikes	candlefiat	livingroom	morningkitchen	nightstreet	notchbush	parkstatue	scooter	streetcorner	mean
NeRF	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.
NeRF w/ CTM	31.95	36.13	32.54	31.64	33.51	31.53	34.32	38.07	32.67	33.60
DVGO	27.90	35.77	32.50	31.39	31.05	30.77	33.43	32.15	30.02	31.67
DVGO w/ CTM	31.98	36.09	32.65	31.43	32.33	31.14	33.88	38.24	32.71	33.38

Table 11: *PSNR*(↑) *on RawNeRF dataset*

Methods	bikes	candlefiat	livingroom	morningkitchen	nightstreet	notchbush	parkstatue	scooter	streetcorner	mean
NeRF	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.
NeRF w/ CTM	0.8317	0.8992	0.9064	0.8589	0.8868	0.8023	0.8982	0.9177	0.8615	0.8736
DVGO	0.7182	0.8872	0.9003	0.8555	0.8509	0.7783	0.8636	0.7940	0.7873	0.8261
DVGO w/ CTM	0.8448	0.8999	0.9152	0.8708	0.8789	0.7932	0.8923	0.9143	0.8678	0.8752

Table 12: *SSIM*(↑) *on RawNeRF dataset*

Methods	bikes	candlefiat	livingroom	morningkitchen	nightstreet	notchbush	parkstatue	scooter	streetcorner	mean
NeRF	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.
NeRF w/ CTM	0.1768	0.1360	0.0982	0.1781	0.1339	0.2634	0.1051	0.1656	0.1280	0.1539
DVGO	0.2754	0.1473	0.1056	0.1867	0.1770	0.2832	0.1793	0.2333	0.2065	0.1994
DVGO w/ CTM	0.1821	0.1358	0.0940	0.1675	0.1458	0.2681	0.1366	0.1638	0.1328	0.1585

Table 13: *LPIPS*(↓) *on RawNeRF dataset*

Methods	chair	drums	ficus	hotdog	lego	materials	mic	ship	mean
NeRF	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.
NeRF w/ CTM	43.56	35.63	41.24	37.54	41.17	38.16	38.03	35.58	38.86
DVGO	32.09	29.20	36.14	30.07	31.92	26.52	29.74	27.14	30.35
DVGO w/ CTM	37.18	33.23	39.05	33.18	36.26	31.58	35.54	33.56	34.95

Table 14: *PSNR*(↑) *on Synthetic-NeRF-Dark dataset*

Methods	chair	drums	ficus	hotdog	lego	materials	mic	ship	mean
NeRF	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.
NeRF w/ CTM	0.9775	0.9398	0.9594	0.9801	0.9705	0.9676	0.9704	0.9277	0.9616
DVGO	0.8043	0.7530	0.8588	0.8929	0.8460	0.6889	0.7927	0.7113	0.7935
DVGO w/ CTM	0.9394	0.8972	0.9248	0.9492	0.9394	0.8896	0.9468	0.8584	0.9181

Table 15: *SSIM*(↑) *on Synthetic-NeRF-Dark dataset*

Methods	chair	drums	ficus	hotdog	lego	materials	mic	ship	mean
NeRF	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.
NeRF w/ CTM	0.0306	0.0801	0.0392	0.0279	0.0298	0.0412	0.0370	0.1090	0.0494
DVGO	0.2442	0.3244	0.1972	0.1380	0.2111	0.3415	0.2727	0.3872	0.2645
DVGO w/ CTM	0.1175	0.1998	0.1255	0.0783	0.1088	0.1525	0.1141	0.2203	0.1396

Table 16: *LPIPS*(↓) *on Synthetic-NeRF-Dark dataset*

Methods	bathroom	bathroom-2	bathroom-3	dark-kitchen	island-kitchen	low-poly-deer	staircase	mean
NeRF	34.95	35.23	36.70	37.84	36.16	42.97	41.71	37.94
NeRF w/ CTM	35.14	35.33	36.70	38.21	36.45	43.13	41.93	38.13
DVGO	35.07	35.63	35.09	37.15	36.15	39.68	35.61	36.34
DVGO w/ CTM	34.48	36.08	35.33	37.34	36.77	40.40	38.50	36.99

Table 17: *PSNR*(↑) *on Low-Texture-Blender dataset*

Methods	bathroom	bathroom-2	bathroom-3	dark-kitchen	island-kitchen	low-poly-deer	staircase	mean
NeRF	0.9516	0.9548	0.9516	0.9592	0.9508	0.9875	0.9840	0.9628
NeRF w/ CTM	0.9507	0.9547	0.9519	0.9603	0.9522	0.9871	0.9842	0.9630
DVGO	0.9617	0.9645	0.9494	0.9560	0.9541	0.9814	0.9600	0.9610
DVGO w/ CTM	0.9601	0.9677	0.9519	0.9579	0.9607	0.9829	0.9753	0.9652

Table 18: *SSIM*(↑) *on Low-Texture-Blender dataset*

Methods	bathroom	bathroom-2	bathroom-3	dark-kitchen	island-kitchen	low-poly-deer	staircase	mean
NeRF	0.0487	0.0511	0.0570	0.0569	0.0840	0.0068	0.0161	0.0458
NeRF w/ CTM	0.0498	0.0501	0.0567	0.0564	0.0839	0.0073	0.0156	0.0457
DVGO	0.0364	0.0313	0.0670	0.0593	0.0634	0.0189	0.0760	0.0503
DVGO w/ CTM	0.0388	0.0282	0.0623	0.0542	0.0450	0.0156	0.0327	0.0395

Table 19: *LPIPS*(↓) *on Low-Texture-Blender dataset*