

# Enhancing Close-up Novel View Synthesis via Pseudo-labeling

Jiatong Xia\*, Libo Sun\*<sup>†</sup>, Lingqiao Liu

Australian Institute for Machine Learning, The University of Adelaide  
{jiatong.xia, libo.sun, lingqiao.liu}@adelaide.edu.au

## Abstract

Recent methods, such as Neural Radiance Fields (NeRF) and 3D Gaussian Splatting (3DGS), have demonstrated remarkable capabilities in novel view synthesis. However, despite their success in producing high-quality images for viewpoints similar to those seen during training, they struggle when generating detailed images from viewpoints that significantly deviate from the training set, particularly in close-up views. The primary challenge stems from the lack of specific training data for close-up views, leading to the inability of current methods to render these views accurately. To address this issue, we introduce a novel pseudo-label-based learning strategy. This approach leverages pseudo-labels derived from existing training data to provide targeted supervision across a wide range of close-up viewpoints. Recognizing the absence of benchmarks for this specific challenge, we also present a new dataset designed to assess the effectiveness of both current and future methods in this area. Our extensive experiments demonstrate the efficacy of our approach.

**Code** — <https://github.com/JiatongXia/Close-up-View-Pseudo-Labeling>

## Introduction

In recent years, there has been a surge in the use of radiance field approaches, such as Neural Radiance Fields (NeRF) (Mildenhall et al. 2020), for view synthesis. To extend the practical applications of radiance fields, various enhancements have been introduced, including methods aimed at increasing processing efficiency (Chen et al. 2021; Neff et al. 2021; Yu et al. 2021; Kurz et al. 2022) and enabling image manipulation capabilities (Lin et al. 2021; Zhang et al. 2021a; Wang et al. 2023; Kuang et al. 2023). However, despite the extensive development of radiance field techniques, these methods often fall short when tasked with producing high-quality images from viewpoints that significantly differ from those seen in the training data. A particularly challenging case is the generation of close-up views, which are often sought by users who wish to examine the fine details of an object from specific angles, particularly in situations where

it’s difficult to position a camera at the desired location in real-world scenarios.

In this paper, we investigate the use of radiance fields to generate close-up views from model trained on distant viewpoints. This involves training on images that capture scenes from afar, while the goal is to produce detailed close-up views of specific elements from various angles. The primary challenge lies in the lack of diverse close-up views within the training dataset, which leaves the radiance field unoptimized for rendering accurate close-up perspectives.

To address this challenge, we propose refining the training protocol for radiance fields by integrating pseudo-annotations derived from a carefully crafted pseudo-labeling approach. Our method involves generating a virtual close-up viewpoint at random during each training iteration. For each virtual viewpoint, we create wrapped images by mapping pixels from the original training images based on the rendered depth. These wrapped images are then evaluated for consistency and occlusion to determine which pixels can be effectively used as pseudo-training data. These pseudo-training data are subsequently incorporated into the general training process to enhance performance for close-up views.

Additionally, recognizing that users may prefer to achieve high-quality results for specific close-up views by slightly fine-tuning an existing model rather than training a new model, we introduce a test-time fine-tuning method that significantly improves performance for specific close-up views while requiring minimal processing time. Furthermore, we have developed a dataset specifically designed to evaluate the generation of close-up views. This dataset addresses the current lack of benchmarks for assessing the performance of existing and future methods in this domain.

## Related Work

**Neural Radiance Fields.** The original NeRF, introduced by Mildenhall et al. (Mildenhall et al. 2020), represents a scene as a continuous 5D function that maps spatial coordinates and viewing directions to radiance values. Since its introduction, NeRF-related techniques have found applications in various computer vision tasks (Zhang et al. 2021b; Chen et al. 2022; Azinović et al. 2022; Liu et al. 2024; Chen et al. 2023). In the context of enhancing human-machine interaction, several works (Wang et al. 2022; Chen et al. 2022; Bao et al. 2023) have demonstrated NeRF’s capabilities. Liu

\*These authors contributed equally.

<sup>†</sup>Corresponding author.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

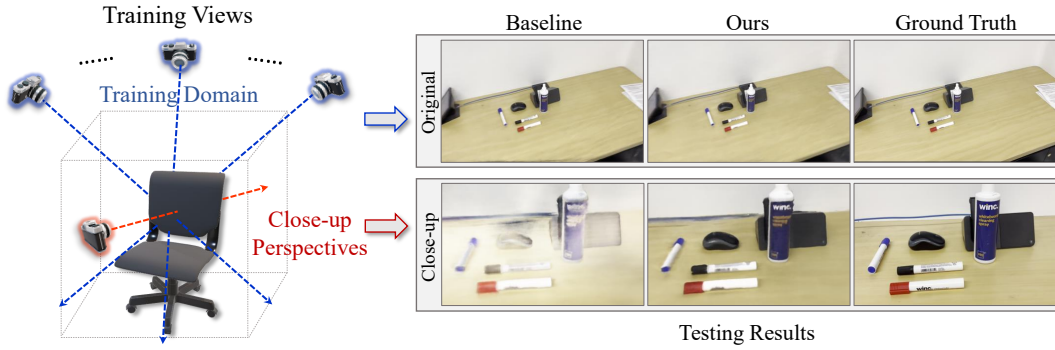


Figure 1: This work investigate novel view synthesis from viewpoints significantly different from those in the training data (blue camera), particularly focus on diverse close-up perspectives (red camera). When facing such close-up perspectives, radiance field methods often exhibit artifacts similar to those seen in the baseline results depicted above. We initially delve into this particular issue, and our proposed method in this paper significantly enhanced radiance fields in such situation.

et al. (Liu et al. 2021) proposed a method for NeRF editing, introducing a technique to propagate coarse 2D user scribbles into 3D space for shape and color modification. Yuan et al. (Yuan et al. 2022) developed a method that enables controllable shape deformation within the implicit scene representation, allowing for scene editing without network re-training. Kerr et al. (Kerr et al. 2023) incorporated raw CLIP embeddings into NeRF to support diverse natural language queries across real-world scenes. In the context of limited data, DS-NeRF (Deng et al. 2022) leverages sparse depth data to provide additional supervision, thus improving the performance of radiance fields. Mip-NeRF (Barron et al. 2021) efficiently rendering anti-aliased conical frustums instead of rays, reduces aliasing artifacts and improves NeRF’s capability to capture fine details. To accelerate processing speed, Instant-NGP (Müller et al. 2022) uses a hash grid and an occupancy grid to accelerate computation and a smaller MLP to represent density and appearance to achieve faster training. Zip-NeRF (Barron et al. 2023) further integrates advancements from scale-aware anti-aliased NeRFs and fast grid-based NeRF training, combining their strengths to enhance performance and efficiency.

**Gaussian Splatting.** Alongside the widespread application of NeRF, 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) has emerged as a powerful technique for novel view synthesis. Compared to NeRF, 3DGS requires less training time and enables high-quality, real-time novel view synthesis at high resolution. Yuan et al. (Huang et al. 2024) further proposed a novel approach, 2D Gaussian Splatting (2DGS), which incorporates depth distortion and normal consistency to more accurately model and reconstruct geometrically precise radiance fields. More recently, Yu et al. (Yu et al. 2024) introduced Mip-Splatting, which employs two sampling filters to limit the maximum frequency of Gaussian primitives and approximate the box filter, thereby simulating the physical imaging process. While their approach is innovative in addressing aliasing issues, it diverges from our focus as it does not effectively handle viewpoints that significantly differ from those in the training domain (e.g., close-up views with varying camera orientations).

## Method

### Preliminary: Neural Radiance Fields

We would use neural radiance fields (NeRF) as a representative radiance field approach to introduce our method. NeRF utilize an MLP network to map a 3D location  $\mathbf{x} \in R^3$  and a viewing direction  $\mathbf{d} \in R^3$  to color values  $\mathbf{c} \in R^3$  and a volume density  $\sigma$ . The mapping function  $\mathcal{F}_\theta$  can be defined as  $\mathcal{F}_\theta(\mathbf{x}, \mathbf{d}) = (\mathbf{c}, \sigma)$ , where  $\theta$  represents the learnable parameters of the MLP. To generate a pixel in a novel view, a camera ray  $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$  is back-projected from the camera center  $\mathbf{o}$  in the direction of  $\mathbf{d}$ . The color of a pixel is then rendered as:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^N w_i \mathbf{c}_i, \quad (1)$$

$$\text{with } w_i = T_i(1 - \exp(-\sigma_i \delta_i)), \quad (2)$$

$$T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right), \quad (3)$$

$$\delta_i = t_{i+1} - t_i. \quad (4)$$

where  $t_i \in [t_n, t_f]$ , with  $t_n$  representing the near bound and  $t_f$  representing the far bound of a ray. To optimize the network parameters  $\theta$ , an RGB MSE loss is imposed between the rendered pixels and the pixels of training images:

$$\mathcal{L}_c(\mathbf{r}) = \left\| \hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r}) \right\|_2^2. \quad (5)$$

Previous methods only provide supervision within the training rays sample. As a result, their performance in viewpoints significantly different from those in the training data becomes uncertain, which leads to the problem that we aim to address.

### The Problem of Close-up Observations

After optimizing the radiance field through the process outlined in the previous section, any ray with camera center  $\mathbf{o}$  and ray direction  $\mathbf{d}$  similar to the training rays can be sampled through the MLP network to obtain RGB results that

closely approximate the quality of the training ground truth after rendering. This process can be conceptually understood as utilizing the optimized radiance field to interpolate the unknown rays within the training ground truth data.

However, as depicted in Fig. 1, when we move a camera pose closer and simultaneously randomly alter its orientation, the resulting rays will have camera centers  $\mathbf{o}$  and ray directions  $\mathbf{d}$  that significantly diverge from those in the training data. These rays often yield unreliable results in the final rendering output, typically manifesting as artifacts. Specifically, NeRF network is configured with a branch to output RGB values. At the initial stage of this branch, the direction of the ray  $\mathbf{d}$  is taken as input. when an untrained ray direction is input for a sampling point  $\mathbf{x}$ , the network tends to produce an unreliable RGB value  $\mathbf{c}$ . Such sampling points often lead to artifacts in the final volume rendering (Eq. 1).

In terms of density, the radiance field itself exhibits robustness in restoring geometric information (Eq. 2), therefore, the density results of sample points on rays in the untrained domain are typically robust than RGB results. Nonetheless, as we bring the camera closer, due to the absence of constraints on rays in the this distance and direction, minor density deviations in original training domain are amplified.

### Pseudo-labeling for Diverse Close-up Perspectives

Drawing from the knowledge presented above, inference rays outside the training samples exhibit erroneous density and color due to insufficient learning from training data. This deficiency leads to errors such as artifacts when rendering close-up observations across diverse perspectives. To tackle this issue, our goal is to generate reliable training labels for perspectives that involve close-up observations and use these generated labels to train and enhance the radiance field, thereby improving the performance of close-up observations. To begin with, we propose an approach to generate camera poses for diverse close-up perspectives. Then, we demonstrate how to generate reliable labels for each generated camera pose.

**Diverse Close-up Perspectives Generation.** To efficiently fine-tune the radiance field in close-up perspectives, it is imperative to generate diverse perspectives outside the training perspectives which have images captured from far distances. Specifically, each camera pose is generated through computations involving the geometry of the radiance field and the training perspectives  $\{\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_N\}$ . Initially, for a random selected training pose  $\mathbf{P}_n$ ,

$$\mathbf{P}_n = [\mathbf{R}_n | \mathbf{t}_n] \text{ with } n \in (0, N), \quad (6)$$

its corresponding depth map  $\mathbf{D}_n$  can be rendered from the radiance field. Then, a pixel  $(u_a, v_a)$  in  $\mathbf{D}_n$  is randomly selected as the anchor point for generating a close-up perspective. For the pixel  $(u_a, v_a)$ , its corresponding 3D position  $\mathbf{X}_a = (x_a, y_a, z_a)$  in the world coordinate can be obtained as:

$$\mathbf{X}_a = \mathbf{o}_a + \mathbf{D}_n(u_a, v_a) \cdot \mathbf{d}_a, \quad (7)$$

where the camera center  $\mathbf{o}_a = \mathbf{t}_n$  and  $\mathbf{d}_a$  is the ray direction calculated from  $\mathbf{R}_n$ .

To generate a close-up perspective, we randomly select a 3D point between the camera center of  $\mathbf{P}_n$  and point  $\mathbf{X}_a$  to form a new camera position  $\mathbf{t}'_n$ :

$$\mathbf{t}'_n = \frac{((\lambda - 1) \cdot \mathbf{X}_a + \mathbf{t}_n)}{\lambda}, \quad (8)$$

where  $\lambda$  is the magnification of how closer the camera is to the anchor point.

To obtain the sample space of camera rotations, we use the rotation matrix  $\mathbf{R}_n$  of the original training pose as the reference. The conversion between Euler angles and rotation matrices is a common operation in 3D perception, and we define these two conversion as  $F_{\mathbf{R} \Rightarrow \mathbf{e}}$  and  $F_{\mathbf{e} \Rightarrow \mathbf{R}}$ . Therefore,  $\mathbf{R}_n$  can be converted into Euler angles as:

$$F_{\mathbf{R} \Rightarrow \mathbf{e}}(\mathbf{R}_n) = (\theta_{x_n}, \theta_{y_n}, \theta_{z_n}), \quad (9)$$

where  $(\theta_{x_n}, \theta_{y_n}, \theta_{z_n})$  represent the values of Euler angles. After converting  $\mathbf{R}_n$  to Euler angles, the new camera orientation  $\mathbf{e}'$  is generated as:

$$\mathbf{e}'_n = \begin{cases} \theta'_x = \theta_{x_n} + \Delta\theta_x, & \Delta\theta_x \in (\theta_{x_n} - \varepsilon, \theta_{x_n} + \varepsilon) \\ \theta'_y = \theta_{y_n} + \Delta\theta_y, & \Delta\theta_y \in (\theta_{y_n} - \varepsilon, \theta_{y_n} + \varepsilon) \\ \theta'_z = \theta_{z_n} + \Delta\theta_z, & \Delta\theta_z \in (\theta_{z_n} - \varepsilon, \theta_{z_n} + \varepsilon), \end{cases} \quad (10)$$

where  $\Delta\theta_x$ ,  $\Delta\theta_y$ , and  $\Delta\theta_z$  are three randomly generated offsets, and  $\varepsilon$  is a parameter used to ensure that the camera orientation does not change too drastically. Once the new camera orientation is obtained, its corresponding rotation matrix  $\mathbf{R}'_n$  can be obtained as:

$$\mathbf{R}'_n = F_{\mathbf{e} \Rightarrow \mathbf{R}}(\mathbf{e}'_n). \quad (11)$$

Finally, the camera pose for a randomly generated close-up perspective is written as:

$$\mathbf{P}'_n = [\mathbf{R}'_n | \mathbf{t}'_n]. \quad (12)$$

**Pseudo Labels and Masks.** With a pre-trained radiance field (i.e. a NeRF model), we can render the depth of each pixel in a given view. For a virtual pose  $\mathbf{P}'_n$  from the camera pose  $\mathbf{P}_n$ , we can render its depth map  $\mathbf{D}'_n$  and thus calculate the 3D coordinates of each pixel. Then we can find projections those coordinates on a training view  $\mathbf{I}_n$  and their corresponding pixel value. This could lead to a wrapped image  $\mathbf{I}'_n$  by extracting pixel values from  $\mathbf{I}_n$  based on the depth estimation  $\mathbf{D}'_n$ . This process can be denoted as  $\mathbf{I}'_n = W(\mathbf{I}_n, \mathbf{D}'_n, \mathbf{P}'_n, \mathbf{P}_n, \mathbf{K})$ , where  $\mathbf{K}$  is the camera intrinsic matrix.

Alternatively, we can render the depth of each pixel in  $\mathbf{I}_n$  and find their correspond projections in  $\mathbf{I}'_n$ . One can also copy the pixel value from  $\mathbf{I}_n$  to the corresponded pixel in  $\mathbf{I}'_n$  to create a wrapped image  $\mathbf{I}^*_n$ . It important to note that not every pixel in  $\mathbf{I}_n$  can find correspondence in view  $\mathbf{I}'_n$ , the wrapped image should contain some undefined pixels. This process can be denoted as  $\mathbf{I}^*_n = W(\mathbf{I}_n, \mathbf{D}_n, \mathbf{P}'_n, \mathbf{P}_n, \mathbf{K})$ .

Ideally, if the depth estimation is accurate, those two wrapped images should have same RGB values at the corresponding pixels. However, due to the error of depth estimation in  $\mathbf{D}$  and  $\mathbf{D}'$ , those values may not be consistent. If their pixel value difference is small enough, we will reasonably assume that their corresponding depth estimation is correct.

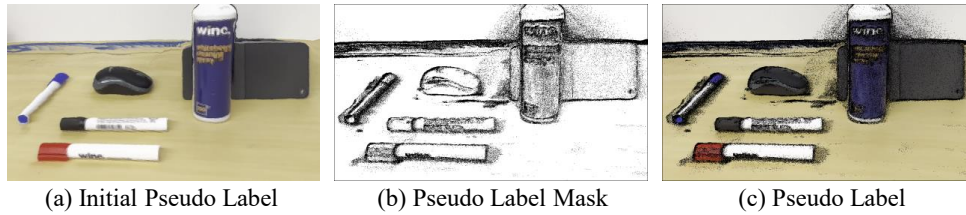


Figure 2: The generated pseudo labels. We show the intermediate outputs in the pseudo label generation process: (a) The initial pseudo label extracted from training images (i.e.,  $\mathbf{I}'_n$ ), (b) The pseudo label mask as described in Eq. 13, and (c) The final pseudo-label obtained after applying the mask.

The wrapped image  $\mathbf{I}_n^*$  for pose  $\mathbf{P}'_n$  can be obtained from multiple training view images  $\{\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_N\}$ , each resulting in a wrapped image. In other words, for a given pixel  $(u, v)$ , there might be  $\mathbf{I}_n^*(u, v)$ ,  $n \in \{1, \dots, N\}$ ,  $N$  possible values. We aggregate them by always retaining the pixel value corresponding to the minimum depth. This is because occlusions need to be considered, where nearby points can block distant points. By warping all the training view images and checking for occlusions, we can update  $\mathbf{I}_n^*$  to  $\bar{\mathbf{I}}_n^*$ , which contains projected pixels from all the training view images while considering occlusions.

The consistency between  $\mathbf{I}'_n$  and  $\bar{\mathbf{I}}_n^*$  can be checked to remove errors contained in  $\mathbf{I}'_n$  for composing the pseudo label used in training. For each pixel  $(u, v)$  in image  $\mathbf{I}'_n$ , the mask  $\mathbf{M}_n$  which decides if  $\mathbf{I}'_n(u, v)$  will be used as a pseudo label for fine-tuning is defined as:

$$\mathbf{M}_n(u, v) = \begin{cases} \text{True} & \text{if } |\mathbf{I}'_n(u, v) - \bar{\mathbf{I}}_n^*(u, v)| < \epsilon \\ \text{False} & \text{otherwise,} \end{cases} \quad (13)$$

where  $\epsilon$  denotes the threshold for determining whether two RGB values can be considered as matched, and we set  $\epsilon$  to 0.05 in this paper. An example of a generated pseudo-label is shown in Fig. 2. As we can observe from this figure, our method can generate accurate pseudo-labels and remove areas that contain errors.

### Training on Diverse Close-up Perspectives

In order to include rays from a wide variety of diverse perspectives during training, we adopt a strategy of generating a new virtual camera pose along with its corresponding pseudo labels at each fine-tuning iteration. We randomly select the value of  $\lambda$  in Eq. 8 from the range of  $(2, 8)$  for each iteration. This implies that the virtual camera pose in each iteration can be randomly positioned closer from  $\frac{1}{2}$  to  $\frac{1}{8}$  of the distance between the original training camera location and the anchor point. And set the value of  $\varepsilon$  in Eq. 10 to  $\frac{\pi}{4}$ . We set the fine-tuning process for 10k iterations, resulting in a total of 10k random virtual camera poses used for fine-tuning, significantly covers the untrained regions.

**Batchify Random Rays for NeRF.** When fine-tuning NeRF, rendering the entire depth map for each virtual pose at every iteration can be extremely time-consuming. Therefore, we propose a batchify random virtual perspectives training.

In each training iteration, instead of rendering the entire depth map for  $\mathbf{P}'_n$ , we randomly choose a set of 2D coordinates  $\{(u_1, v_1), (u_2, v_2), \dots, (u_B, v_B)\}$ , where  $B$  represent the batch size. Using these 2D coordinates, we can sample a training batch-size collection of rays  $\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_B\}$  from virtual camera pose  $\mathbf{P}'_n$ , then render the depth values  $\{z_1, z_2, \dots, z_B\}$  for this set of rays through radiance field and derive pseudo labels  $\{\mathbf{I}'_n(u_k, v_k)\}_{k=1}^B$  for this set of rays.

We combine this batch of pseudo labels with the same batch size of original training samples, then fine-tune the radiation field to encompass diverse perspectives across both trained and untrained domains. The supervision provided by pseudo labels here can be represented as:

$$\mathcal{L}_{pl} = \left\| \hat{\mathbf{C}} - \mathbf{I}'_n \right\|_2^2. \quad (14)$$

And the overall loss function  $\mathcal{L}$  can be expressed as follows:

$$\mathcal{L} = \frac{1}{2} (\mathcal{L}_c + \mathcal{L}_{pl}). \quad (15)$$

**Fine-tuning on Gaussian Splatting.** Instead of per pixel rendering, the tile-based fast rasterizer of Gaussian Splatting could render an entire image and its depth map at a time. This allows us to directly produce the entire pseudo label  $\mathbf{I}'_n$  and its corresponding mask  $\mathbf{M}_n$  for generated close-up pose  $\mathbf{P}'_n$ , thus we can apply  $\mathbf{I}'_n$  to provide supervision for Gaussian Splatting fine-tuning after filtering with  $\mathbf{M}_n$ . We choose 2DGS (Huang et al. 2024) as our baseline method of Gaussian Splatting in this paper due to its robust capability to generate reliable depth maps for each synthesized view.

### Test-time Fine-tuning

In practical scenarios, users may occasionally have specific perspectives in mind that they wish to use. For instance, they might want to observe a cup on a table from particular viewpoints. In such cases, fine-tuning the radiance field on these predetermined perspectives can significantly reduce the iteration times required for fine-tuning with randomly generated camera poses and accelerate the process.

We consider test-time fine-tuning as optional and as an additional benefit. Specifically, creating virtual camera poses in an extensive untrained domain and refining the radiance field by applying pseudo labels with masks can enhance the overall rendering quality for diverse perspectives. And directly generating pseudo-labels and masks for the camera poses that need to be tested in the untrained domain, and



Figure 3: Typical existing view synthesis benchmarks. The test images are positioned at the similar distance as the training images and share highly similar view directions.



Figure 4: Our dataset with each row as the example of a scene, where the training images on the left and the testing images on the right. Training images in each scene are in the same domain of the example, moving forward facing or moving around the objects in a similar distance, while testing images are much closer to the objects and are significantly divergent from the training views.

applying a ‘test-time fine-tuning’ on these testing poses, is more targeted and can lead to rapid convergence, while this setup aligns perfectly with the Gaussian Splatting methods due to their quick training time and the ability for the tile-based rasterizer to render an entire view during optimization. Typically, only 5 iterations are needed to complete the fine-tuning on one specific test view (less than 3 seconds on our test images).

Unlike the generated training perspectives, where imaged are chosen based on a randomly selected anchor to obtain pseudo labels, we introduce a selection strategy to determine which image will be used to obtain pseudo labels. We first project the pixels of all training images  $\{\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_N\}$  onto this specific test pose  $\mathbf{P}''_n$  using their respective training depths  $\{\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_N\}$ , and count the number of pixels from each training image that can be projected onto this test pose, recorded as  $\{\tau_1, \tau_2, \dots, \tau_N\}$ . By assessing  $\{\tau_1, \tau_2, \dots, \tau_N\}$ , we can identify the training image with the highest content that can be projected onto this test pose. Subsequently, we select this training perspective as the projection target. Following the approach outlined earlier, we can proceed to derive pseudo label for  $\mathbf{P}''_n$ . We generate corresponding pseudo labels for all test poses  $\{\mathbf{P}''_1, \mathbf{P}''_2, \dots, \mathbf{P}''_Q\}$ , where  $Q$  represents the number of known test poses, then combine the training batch from these samples with the original training batch. The radiance field can be fine-tuned as the same optimization way in the previous section.

### Dataset with Diverse Close-up Perspectives

As shown in Fig. 3, existing benchmarks for evaluating novel view synthesis methods typically only include test im-

ages within the training domain. However, since there is no available data to evaluate close-up view synthesis, we introduce a new dataset for evaluating the performance of current and future methods from close-up perspectives.

Our dataset comprises diverse scenes, each one is extracted as a subset of frames from a captured video, with 50 to 100 training images and 10 to 20 testing images for each scene, images are captured at a resolution of  $960 \times 540$ . As shown in Fig. 4, our dataset has diverse scene types, contains both indoor and outdoor scenes. And the training images are closely resemble those in existing NeRF benchmarks like LLFF (Mildenhall et al. 2019) and Tanks & Temples (Knapitsch et al. 2017) (as in Fig. 3), while the testing images concentrate on scenarios where camera poses brought much closer to objects and significantly differ from training views. In experiments section, we show the results of typical radiance field methods on our dataset, and the results reveal that our dataset effectively showcases the issues present in various methods within this specific setting.

We follow the main evaluation metrics used in novel view synthesis methods to evaluate the performance on our dataset. Specifically, we use PSNR, SSIM (Wang et al. 2004) and LPIPS (Zhang et al. 2018) to measure the quality of synthesized RGB novel views by comparing them with the ground truth images.

## Experiments

### Implementation Details

We conducted experiments following the implementation and settings of our baseline methods, NeRF (Mildenhall et al. 2020) and 2DGS (Huang et al. 2024). The experiments

Initial Pseudo Label	Pseudo Label Mask	Diverse Close-up Perspective Generation	Novel View Synthesis		
			PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
-	-	-	14.48	0.657	0.552
$\checkmark$	-	-	16.61	0.683	0.548
$\checkmark$	$\checkmark$	-	18.17	0.700	0.520
$\checkmark$	$\checkmark$	$\checkmark$	<b>18.92</b>	<b>0.710</b>	<b>0.515</b>

Table 1: Ablation studies of components in training on diverse close-up perspectives on our proposed dataset.

are conducted on NVIDIA 3090 GPUs and the Adam optimizer (Kingma and Ba 2015) is employed to optimize the radiance field. For both baseline methods, we first trained the radiance field following the implementation and settings of vanilla NeRF and 2DGS. After that, for NeRF as baseline method, we load the weights of the pre-trained NeRF model and optimized for 10K iterations per scene with a ray batch of original training samples and generated samples set to 2048 (1024 for each). For 2DGS as baseline method, we optimized the pre-trained 2DGS model for 500 iterations per scene with each iteration combine supervision from both training image and masked pseudo label. For test-time fine-tuning, we applied our method on both NeRF and 2DGS. Specifically, we optimized the pre-trained 2DGS model with original training samples and generated samples together for iterations of 5 times for each test view. And for NeRF, we optimized the pre-trained model for 200 iterations on test poses, with a batch size of 1024 for each samples.

## Ablation Studies

In this section, we perform comprehensive ablation studies to evaluate the effectiveness of our method, including the effectiveness of training on diverse close-up perspectives and the effectiveness of test-time fine-tuning.

**The Effectiveness of Components in Training on Diverse Close-up Perspectives.** For the initial pseudo labels in Tab. 1, we directly use the warped result  $I'_n$  to generate batch samples in the fine-tuning process. For pseudo label mask, we use mask  $M_n$  to select out the reliable areas in  $I'_n$  for the fine-tuning process. For the fine-tuning without diverse close-up perspective generation, we manually defined four anchor points for each training image and fixed the distance to create four consistent generated poses for each training viewpoint. As shown in Tab. 1, directly employing the warped results as pseudo labels for fine-tuning brings a relatively limited improvement over the baseline, increasing from 14.48 to 16.61 in PSNR. This is because the warped pseudo labels contain inaccuracies, and using them directly for training partially address the issue of artifacts, but simultaneously introduce new errors. Therefore, applying pseudo label masks to filter out erroneous regions can lead to substantial improvements in results, highlighting the crucial function of pseudo label masks in enhancing fine-tuning effectiveness. As we can observe from Tab. 1, applying pseudo label masks improves PSNR from 16.61 to 18.17. The effectiveness of randomly generating diverse close-up perspectives is evaluated in the final row of Tab. 1, it further improves PSNR from 18.17 to 18.92 by maintaining pose diversity, which results in a variety of rays for training.

**The Effectiveness of Test-time Fine-tuning.** If users know which close-up perspectives they need to enhance performance for, we can directly fine-tune an existing model by generating and applying pseudo-labels for those specific perspectives. To demonstrate this, we conducted comparison experiments using NeRF and 2DGS. Regarding processing speed, our experiments revealed that fine-tuning a NeRF model using the proposed test-time fine-tuning method requires only 200 iterations for around 10 testing views, while fine-tuning a 2DGS model is also very fast, requiring just 5 iterations per testing view. In terms of accuracy, as shown in Tab. 2, our test-time fine-tuning method on 2DGS achieves a PSNR of 20.95, compared to the baseline 2DGS PSNR of 19.99. This approach significantly speeds up processing while achieving outstanding performance, even comparable to our general training method. The primary reason for this is that general close-up perspectives training aims to cover diverse perspectives, requiring more iterations for learning. We argue that the proposed test-time fine-tuning does not conflict with the general training method, as it is designed for specific cases. For a general scenario where testing poses are unknown before training, the proposed general training method remains applicable.

**Computational Efficiency.** Training a vanilla NeRF model takes 4 hours on an NVIDIA 3090 GPU, while the additional time introduced by our method—including label generation and optimizations—for fine-tuning this pre-trained NeRF model is 40 minutes. Training a 2DGS model takes 15 minutes, with approximately 7 additional minutes of training time introduced by our method. This additional time is only incurred during training; the inference speed of our method remains the same as that of NeRF and 2DGS.

## Comparisons with Other Methods

In this section, detailed comparisons between our general fine-tuning method and other approaches are presented.

**Comparisons of Numerical Accuracy.** As shown in Tab. 3, we can observe that our method significantly outperforms other methods in terms of all the metrics. For

Method	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
NeRF (Mildenhall et al. 2020)	14.48	0.657	0.552
2DGS (Huang et al. 2024)	19.99	0.768	0.386
Ours-NeRF	18.89 (+4.41)	0.708	0.520
Ours-2DGS	<b>20.95 (+0.96)</b>	<b>0.785</b>	<b>0.383</b>

Table 2: Ablation studies of test-time fine-tuning. We directly use test poses of each scene in our dataset to fine-tune the radiance field, and also test on those poses.

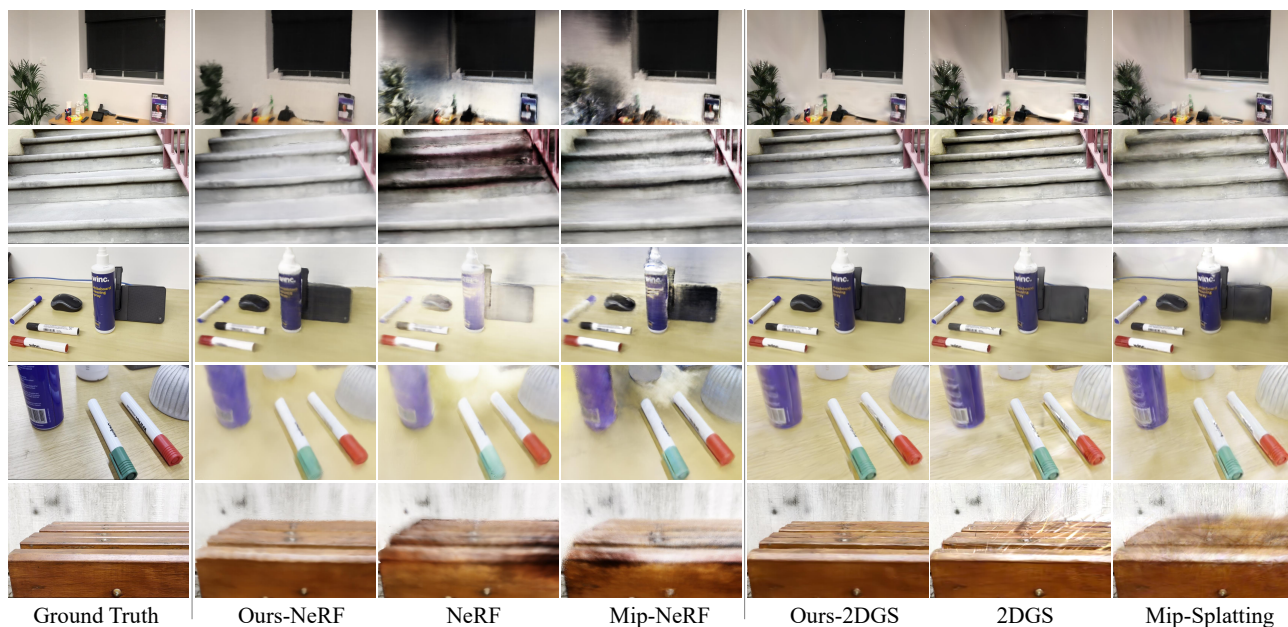


Figure 5: Quantitative comparisons with other methods. We visualize the synthesized images from our method, and compare them with Mip-NeRF, Mip-Splatting and the baseline methods.

Method	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
Instant-NGP (Müller et al. 2022)	14.23	0.615	0.616
TensoRF (Chen et al. 2022)	14.59	0.651	0.613
NeRF (Mildenhall et al. 2020)	14.48	0.657	0.552
Zip-NeRF (Barron et al. 2023)	15.31	0.637	0.510
DS-NeRF (Deng et al. 2022)	16.48	0.676	0.558
Mip-NeRF (Barron et al. 2021)	17.05	0.665	0.554
3DGS (Kerbl et al. 2023)	19.16	0.747	0.430
Mip-Splatting (Yu et al. 2024)	19.55	0.765	0.407
2DGS (Huang et al. 2024)	19.99	0.768	0.386
Ours-NeRF	<b>18.92</b>	<b>0.710</b>	<b>0.515</b>
Ours-2DGS	<b>20.88</b>	<b>0.784</b>	<b>0.383</b>

Table 3: Qualitative comparisons of diverse close-up fine-tuning with other methods.

NeRF-based methods, our method achieves PSNR of 18.92, SSIM of 0.710 and LPIPS of 0.515, demonstrates a significant performance improvement in PSNR (+4.44) compared to our baseline, NeRF. Our method also shows a substantial improvement compared to Instant-NGP, TensoRF and Zip-NeRF. Mip-NeRF demonstrates some robustness when bringing the camera closer compared to NeRF. However, in comparison to our method, it shows difference in PSNR (-1.87), SSIM (-4.5%) and LPIPS. DS-NeRF benefits from its utilization of depth constraints, shows good performance in reconstructing geometric information and exhibits robustness when close-up the viewpoint, but still has significant performance gap from our method. For Gaussian Splatting methods, our method achieves PSNR of 20.88, SSIM of 0.784 and LPIPS of 0.383, effectively improving the baseline method 2DGS across all metrics, significantly outperforming 3DGS in PSNR (+1.72), SSIM (+3.7%) and LPIPS

(-4.7%). Mip-Splatting was proposed to address alias-free rendering, however, it fails to handle the close-up case, achieving only a PSNR of 19.55 (-1.33). These comparisons clearly highlight the remarkable improvement of our method when close-up observations are required.

**Comparisons of Visualized Results.** The visualization results on our benchmark in Fig 5 clearly demonstrate the problem that other conventional methods face under the settings proposed in this paper. Especially in the last row of comparisons, each method’s respective issues are most pronounced. We can observed that the results of NeRF are affected by the issue of artifacts, as we analyzed in our paper. Mip-NeRF, 2DGS and Mip-Splatting also shows similar issues, demonstrating that these challenges are prevalent across radiance field methods. Our method clearly addresses the issue of artifacts compared to our baseline methods. The visualization results demonstrate a notable improvement in the rendering quality, indicating a significant resolution to the problems outlined in our paper.

## Conclusion

This work dig into the complexities of generating reliable images from viewpoints that diverge considerably from the training data, especially for close-up perspectives. We propose a novel learning strategy utilizes pseudo-labels derived from the available training data to offer targeted guidance for various close-up viewpoints to tackle this challenge. We also developed a dataset tailored to evaluate the synthesis of close-up views, addressing the lack of a benchmark for assessing the effectiveness of existing and upcoming methods in this particular task.

## References

- Azinović, D.; Martin-Brualla, R.; Goldman, D. B.; Nießner, M.; and Thies, J. 2022. Neural RGB-D surface reconstruction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 6290–6301.
- Bao, C.; Zhang, Y.; Yang, B.; Fan, T.; Yang, Z.; Bao, H.; Zhang, G.; and Cui, Z. 2023. SINE: Semantic-driven image-based NeRF editing with prior-guided editing field. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 20919–20929.
- Barron, J. T.; Mildenhall, B.; Tancik, M.; Hedman, P.; Martin-Brualla, R.; and Srinivasan, P. P. 2021. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Int. Conf. Comput. Vis.*, 5855–5864.
- Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; and Hedman, P. 2023. Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields. *Int. Conf. Comput. Vis.*
- Chen, A.; Xu, Z.; Geiger, A.; Yu, J.; and Su, H. 2022. Tensorf: Tensorial radiance fields. In *Eur. Conf. Comput. Vis.*, 333–350. Springer.
- Chen, A.; Xu, Z.; Zhao, F.; Zhang, X.; Xiang, F.; Yu, J.; and Su, H. 2021. MVNeRF: Fast generalizable radiance field reconstruction from multi-view stereo. In *Int. Conf. Comput. Vis.*, 14124–14133.
- Chen, H.; Gu, J.; Chen, A.; Tian, W.; Tu, Z.; Liu, L.; and Su, H. 2023. Single-Stage Diffusion NeRF: A Unified Approach to 3D Generation and Reconstruction. In *Int. Conf. Comput. Vis.*
- Deng, K.; Liu, A.; Zhu, J.-Y.; and Ramanan, D. 2022. Depth-supervised NeRF: Fewer views and faster training for free. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 12882–12891.
- Huang, B.; Yu, Z.; Chen, A.; Geiger, A.; and Gao, S. 2024. 2D Gaussian Splatting for Geometrically Accurate Radiance Fields. In *SIGGRAPH 2024 Conference Papers*.
- Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.*, 42(4).
- Kerr, J.; Kim, C. M.; Goldberg, K.; Kanazawa, A.; and Tancik, M. 2023. LERF: Language embedded radiance fields. In *Int. Conf. Comput. Vis.*, 19729–19739.
- Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. *Int. Conf. Learn. Represent.*
- Knapitsch, A.; Park, J.; Zhou, Q.-Y.; and Koltun, V. 2017. Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction. *ACM Trans. Graph.*, 36(4).
- Kuang, Z.; Luan, F.; Bi, S.; Shu, Z.; Wetzstein, G.; and Sunkavalli, K. 2023. PaletteNeRF: Palette-based appearance editing of neural radiance fields. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 20691–20700.
- Kurz, A.; Neff, T.; Lv, Z.; Zollhöfer, M.; and Steinberger, M. 2022. AdaNeRF: Adaptive sampling for real-time rendering of neural radiance fields. In *Eur. Conf. Comput. Vis.*, 254–270. Springer.
- Lin, J.; Zhang, R.; Ganz, F.; Han, S.; and Zhu, J.-Y. 2021. Anycost gans for interactive image synthesis and editing. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 14986–14996.
- Liu, M.; Xu, C.; Jin, H.; Chen, L.; Varma, T. M.; Xu, Z.; and Su, H. 2024. One-2-3-45: Any single image to 3D mesh in 45 seconds without per-shape optimization. *Adv. Neural Inform. Process. Syst.*, 36.
- Liu, S.; Zhang, X.; Zhang, Z.; Zhang, R.; Zhu, J.-Y.; and Russell, B. 2021. Editing conditional radiance fields. In *Int. Conf. Comput. Vis.*, 5773–5783.
- Mildenhall, B.; Srinivasan, P. P.; Ortiz-Cayon, R.; Kalantari, N. K.; Ramamoorthi, R.; Ng, R.; and Kar, A. 2019. Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines. *ACM Trans. Graph.*
- Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Eur. Conf. Comput. Vis.*
- Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Trans. Graph.*, 41(4): 102:1–102:15.
- Neff, T.; Stadlbauer, P.; Parger, M.; Kurz, A.; Mueller, J. H.; Chaitanya, C. R. A.; Kaplanyan, A.; and Steinberger, M. 2021. DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks. In *Comput. Graph. Forum*, volume 40, 45–59.
- Wang, C.; Chai, M.; He, M.; Chen, D.; and Liao, J. 2022. CLIP-NeRF: Text-and-image driven manipulation of neural radiance fields. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 3835–3844.
- Wang, X.; Zhu, J.; Ye, Q.; Huo, Y.; Ran, Y.; Zhong, Z.; and Chen, J. 2023. Seal-3D: Interactive Pixel-Level Editing for Neural Radiance Fields. In *Int. Conf. Comput. Vis.*, 17683–17693.
- Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.*, 13(4): 600–612.
- Yu, A.; Li, R.; Tancik, M.; Li, H.; Ng, R.; and Kanazawa, A. 2021. PlenOctrees for real-time rendering of neural radiance fields. In *Int. Conf. Comput. Vis.*, 5752–5761.
- Yu, Z.; Chen, A.; Huang, B.; Sattler, T.; and Geiger, A. 2024. Mip-Splatting: Alias-free 3D Gaussian Splatting. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 19447–19456.
- Yuan, Y.-J.; Sun, Y.-T.; Lai, Y.-K.; Ma, Y.; Jia, R.; and Gao, L. 2022. NeRF-Editing: geometry editing of neural radiance fields. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 18353–18364.
- Zhang, J.; Li, K.; Lai, Y.-K.; and Yang, J. 2021a. Pise: Person image synthesis and editing with decoupled gan. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 7982–7990.
- Zhang, J.; Yang, G.; Tulsiani, S.; and Ramanan, D. 2021b. NeRS: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild. *Adv. Neural Inform. Process. Syst.*, 34: 29835–29847.
- Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 586–595.