

---

# Active Neural 3D Reconstruction with Colorized Surface Voxel-based View Selection

---

Hyunseo Kim<sup>1</sup>, Hyeonseo Yang<sup>1</sup>, Taekyung Kim<sup>2</sup>, YoonSung Kim<sup>1</sup>,  
Jin-Hwa Kim<sup>\*2</sup>, Byoung-Tak Zhang<sup>\*1</sup>

<sup>1</sup>AI institute, Seoul National University

<sup>2</sup>Naver AI Lab

<sup>1</sup>{hskim, hsyang, yskim, btzhang}@bi.snu.ac.kr

<sup>2</sup>{jlnhwa.kim, taekyung.k}@navercorp.com

## Abstract

Active view selection in 3D scene reconstruction has been widely studied since training on informative views is critical for reconstruction. Recently, Neural Radiance Fields (NeRF) variants have shown promising results in active 3D reconstruction using uncertainty-guided view selection. They utilize uncertainties estimated with neural networks that encode scene geometry and appearance. However, the choice of uncertainty integration methods, either voxel-based or neural rendering, has conventionally depended on the types of scene uncertainty being estimated, whether geometric or appearance-related. In this paper, we introduce *Colorized Surface Voxel (CSV)*-based view selection, a new next-best view (NBV) selection method exploiting surface voxel-based measurement of uncertainty in scene appearance. CSV encapsulates the uncertainty of estimated scene appearance (*e.g.*, color uncertainty) and estimated geometric information (*e.g.*, surface). Using the geometry information, we interpret the uncertainty of scene appearance 3D-wise during the aggregation of the per-voxel uncertainty. Consequently, the uncertainty from occluded and complex regions is recognized under challenging scenarios with limited input data. Our method outperforms previous works on popular datasets, DTU and Blender, and our new dataset with imbalanced viewpoints, showing that the CSV-based view selection significantly improves performance by up to 30%.

## 1 Introduction

Active view selection in 3D reconstruction has been widely studied to deal with the cost efficiency of multi-view data selection [1]. Recent advances in neural networks for 3D reconstruction show that 3D scenes can be represented using implicit neural representations (INR) with higher output quality and smaller space usage [15, 21]. However, popular approaches such as Neural Radiance Fields (NeRF) [15] require a large number of multi-view posed images for effective generalization of neural networks [24]. To alleviate the data selection burden, active view selection with INR emerged and provided a way to select the most informative data samples for training.

Estimating uncertainty is one of the popular methods to measure informativeness in active view selection [17, 4, 22]. Uncertainties estimated with INR are divided into two types, depending on what INR encodes: scene geometry and appearance. The uncertainty in **scene geometry** is usually measured by the voxel-based integration of entropy in the scene geometry [27, 6]. However, the voxel<sup>1</sup>-based entropy measurement generally encodes the density feature in an explicit form like feature grids [6, 3, 27], not in an implicit form like neural network weights. The feature grid is

---

<sup>1</sup>A voxel is a small volume consisting of a volumetric map containing data.

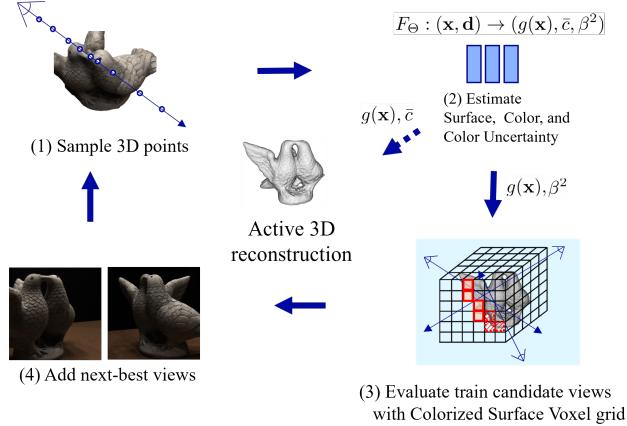


Figure 1: Active neural 3D reconstruction with colorized surface voxel (CSV)-based view selection. (1) We sample 3D points in the given views. (2) Using the points, a neural implicit surface network is trained to predict the signed distance function (SDF)  $g(\mathbf{x})$  and color uncertainty  $\beta^2$  of 3D points. (3) The uncertainty  $\beta^2$  of a 3D point is assigned to the CSV grid in the corresponding 3D position, and the CSVs are marked with the *surfaceness* computed from the SDF  $g(\mathbf{x})$ . A red grid is an inferred surface voxel of a reconstruction target object, and a grid with red diagonal lines is a surface voxel that has not yet been inferred. (4) We choose the next-best views among candidate views using 3D-interpreted color uncertainty from CSV. After adding new views, we continue the point sampling process again.

computationally inefficient since it allocates many features to areas of empty space [16]. When the feature grid is dense, it consumes more memory than the neural network weights.

On the other hand, the uncertainty in the appearance of a scene is usually defined as the uncertainty in color estimation [18, 8, 17]. By modeling the color as a Gaussian probability distribution, its variance is defined as the **color uncertainty**. The color uncertainty of a scene is measured by a ray-based aggregation of the color uncertainty using the classic volume rendering technique [12]. However, using volume rendering to aggregate the uncertainty can lead to less preferred view selection, especially when the views are selected in early training iterations. The density prediction is unreliable in early training iterations [16], and the volume rendering of color uncertainty with unreliable density prediction may fail to provide accurate uncertainty value for view selection. We discussed more details in Sec. 4.

To capture every measurable uncertainty in a scene cost-efficiently, we propose a colorized surface voxel (CSV) that addresses the shortcomings of previous works in three ways. First, we encode the surface information of a 3D point and its color uncertainty with a modified neural implicit surface network [21] (Fig. 1). By encoding geometric and color information of a scene with an implicit neural network, we can reduce the computation costs compared to feature grids. Second, we map color uncertainty and *surfaceness* to a CSV grid for linking the geometric and color information of a scene. A voxel in the CSV grid has a property called *surfaceness* indicating whether it belongs to a surface or not. If a voxel belongs to a surface, it is called a surface voxel. When a 3D point is located on the *visible surface* of an object, its color is observable, and the color uncertainty of that point can be lowered. We called the color uncertainty of the surface point as *reducible* color uncertainty and focused on reducing the color uncertainty on the surface to enhance the output quality. Third, we vary the integration method of uncertainty in CSV depending on the surfaceness. At early training iterations, surface estimation can be unreliable, so considering the surface voxel-wise, not point-wise, increases its robustness. Also, we integrate uncertainties from all voxels that a ray traversed when there is no definite surface voxel. More details are discussed in Sec. 5.

We evaluate the efficacy of our proposed view selection method on the DTU [7] and Blender [15] datasets. Our method outperforms other methods in image rendering and mesh reconstruction qualitatively and quantitatively. We show that the diverse CSV-based view selection leads to performance improvement. We also conduct ablation studies to show the significance of surface information in our

method. Finally, we provide qualitative evaluation results of view selection methods on scenes with occluded objects and imbalanced viewpoints.

**Contributions.** In summary, the contributions of the paper are as follows: (1) We propose *colorized surface voxel* (CSV), a voxel grid that encapsulates color uncertainty and surfaceness. It works as a basic component for 3D interpretations of color uncertainty. (2) We suggest modifying a neural implicit surface network to estimate color uncertainty and surface information. The simultaneous derivation of surface information and color uncertainty enables linking the two types of information in CSV. (3) We present CSV-based view selection, which provides a proper aggregation method using surface information for 3D-interpreted color uncertainty. The computed uncertainty of an image accurately reflects the reducible color uncertainty visible in the view. (4) We present our new dataset with imbalanced viewpoints, which reflect the real-world condition of the data collection environment.

## 2 Related work

### 2.1 Geometric uncertainty in volumetric representations

In prior works of active 3D reconstruction, new view positions are evaluated by examining volumetric representations encoding geometry: camera ray-traversed voxels or boundaries of estimated surfaces (meshes) [6, 3]. The views with high uncertainty are selected, and the uncertainty concerning scene geometry is defined as the expected information regarding voxel occupancy [6]. For example, when a robot or a sensor observes a part of the scene, the occupancy probability of the corresponding voxel approaches 0 or 1. Then, the uncertainty of scene geometry reduces because, after an observation, the expected information within the volumetric map is reduced.

The ActiveRMAP [27] also conducted active 3D reconstruction using the voxel-based uncertainty measurement. The ActiveRMAP is based on DVGO architecture [19] that encodes scene geometry with an explicit feature grid and selects views with an additional volumetric representation that stores the entropy of occupancy probability. However, the ActiveRMAP is two-stage optimized to compensate for the computation inefficiency in an explicit feature grid since it allocates many features to areas of empty space. This work estimates uncertainties with an implicit neural network, which efficiently encodes the scene.

### 2.2 Neural network uncertainty in active 3D reconstruction

As an INR is increasingly applied in the active reconstruction field, various forms of uncertainty are utilized for NBV selections [17, 11, 18, 8]. Lee *et al.* [11] measured uncertainty from scene geometry by using the entropy of weight distribution of 3D points along a camera ray. However, they require initializing neural network weights with camera views from a 360-degree perspective, making them challenging to apply in common scenarios. Also, there are works that measure uncertainty in scene appearance [18, 8, 17], primarily by modeling the color as a Gaussian probability distribution. For scene uncertainty, the color uncertainty is conventionally integrated using volume rendering [12]. However, the density estimation used in volume rendering is unreliable in the early training stage [16]. The uncertainty integrated with errors may misdirect the view selection process. In this work, we infer a surface in scene geometry using neural networks and vary integration strategies depending on the surface to deal with unreliable estimation.

## 3 Preliminaries

### 3.1 Neural implicit surfaces

NeuS [21] utilizes a neural signed distance function (SDF) to encode a scene geometry. Given a 3D point’s position  $\mathbf{x}$  and a viewing direction  $\mathbf{d}$ , the neural implicit surface network (NeuS) interprets the scene geometry into SDF  $g(\mathbf{x})$  and color  $\mathbf{c}$  ( $F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (g(\mathbf{x}), \mathbf{c})$ ). The zero-level set of SDF represents the surface  $\mathcal{S}$  of an object:

$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^3 | g(\mathbf{x}) = 0\}. \quad (1)$$

NeuS introduces the S-density field  $\phi_s(g(\mathbf{x}))$  to integrate the SDF with the volume rendering [12]. The  $\phi_s(x)$  denotes the derivative of the Sigmoid function  $\Phi_s(x) = (1 + e^{-sx})^{-1}$ , a zero-centered

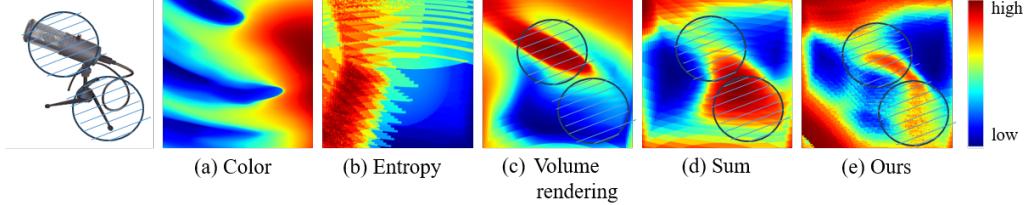


Figure 2: Pixel-wise visualization of information gain (IG) in NBV selection methods. The leftmost image is the ground-truth image of the fixed viewpoint used in the visualizations. The IG images are visualized when the reconstruction of a microphone is 5% progressed (5 views collected). The head and the cord of the microphone are marked with slashed circles. Note that the reconstructions are based on the neural implicit surface network, not NeRF. (a) and (b) do not show geometric characteristics. (c) has high IG at the head of the microphone and low IG at the cord of the microphone, while (d) shows the opposite situation. (e) shows moderately high IG value at both positions.

unimodal density distribution. Note that the standard deviation of  $\phi_s$  is  $1/s$ , which approaches zero as training converges with a learnable parameter  $s$ . The value  $1/s$  is also interpreted as the point sampling step size, which means the network infers a more precise and thinner surface as training progresses. To deal with general volume rendering cases, NeuS introduces opaque density  $\rho$ , analogous to NeRF’s [15] volume density  $\sigma$ , and the discrete opacity value  $\alpha$ :

$$\alpha_j = 1 - \exp\left(-\int_{t_j}^{t_{j+1}} \rho(t) dt\right), \quad \rho(t_j) = \max\left(\frac{-\frac{d\Phi_s}{dt}(g(\mathbf{r}_i(t_j)))}{\Phi_s(g(\mathbf{r}_i(t_j)))}, 0\right) \quad (2)$$

where  $N$  points are sampled in an  $i$ -th ray  $\mathbf{r}_i(t_j)$ . The final loss function consists of color and Eikonal [5] regularization losses. The color loss in Eq. (3) is averaged over  $M$  camera rays, where  $C(\mathbf{r})$  and  $\hat{C}(\mathbf{r})$  are the ground truth and the predicted colors for the ray  $\mathbf{r}$ , respectively. The Eikonal loss encourages the gradients of SDF  $\nabla g(\mathbf{r}_i(t_j))$  to be of unit 2-norm, and with a parameter  $\lambda$ , the average of Eikonal loss is taken to the number of sample points  $N$ :

$$\mathcal{L}_s = \frac{1}{M} \sum_{i=1}^M \left[ \|(\hat{C}(\mathbf{r}_i) - C(\mathbf{r}_i))\|_1 + \frac{\lambda}{N} \sum_{j=1}^N (\|\nabla g(\mathbf{r}_i(t_j))\|_2 - 1)^2 \right] \quad (3)$$

### 3.2 Information gain in voxel grid

In NBV selection, the information within a voxel grid is defined as the entropy  $I(x)$  of voxel occupancy [6]. When the voxel grid is initialized, the occupancy probability  $p(x)$  of a voxel  $x$  is initialized at 0.5 since the occupancy of a voxel has not yet been determined. Therefore, the initialized voxel has the highest entropy (expected information).  $I(x)$  is formulated as:

$$I(x) = -p(x) \log p(x) - (1 - p(x)) \log(1 - p(x)). \quad (4)$$

The information gain (IG) in a voxel grid is defined as the sum of expected information enclosed in voxels that are visible from a particular view [20]. Let  $\mathcal{R}_v$  be the set of camera rays traversing the voxel grid on camera view  $v$ , and  $\mathcal{X}$  be the set of voxels that the rays traverse through. The information gain  $G(v)$  of a camera view  $v$  is defined as:

$$G(v) = \frac{1}{n} \sum_{\forall r \in \mathcal{R}_v} \sum_{\forall x \in \mathcal{X}_r} I(x). \quad (5)$$

where  $n$  is the total number of traversed voxels.

## 4 Analysis on the formulation of information gain

We compared IG measured by various NBV selection methods in Fig. 2. *Color* is an NBV selection method that uses IG formulation of ActiveNeRF [17], which is the difference in color variance

between a 3D point's prior distribution and its predicted posterior distribution after a candidate view is incorporated. However, as shown in Fig. 2 (a), the difference in color variance does not reflect the geometric characteristic of the scene. *Entropy* is an NBV selection method using IG formulation from Isler *et al.* [6]. The IG of *Entropy* (Eq. (5)) does not show the geometric characteristic when it has geometric property in its IG formulation due to the difficulty in converging geometry estimation in the microphone scene in Fig. 2. Since the *Entropy* selects NBV with IG measured from unreliable geometry estimation in early training iterations, it can select less informative views, affecting the training process afterward. Therefore, the IG measurement may lose geometric characteristics if the *Entropy* selects less informative views and cannot be converged.

In Fig. 2 (c,d,e), we formulate the IG using CSV. We vary the IG formulations of CSV to identify which formulation captures the information of a scene well. First, we used a conventional volume rendering technique [12] to integrate color uncertainties of a view (Fig. 2 (c)). The volume rendering emphasizes the color uncertainty of a 3D point with high density. Generally, the points on the surface of an object have high density when the surface estimation is converged. However, surface estimation is unreliable in early training iterations, and the neural network encoding surface tends to estimate a lump of low-density points where the convergence of estimation has not yet been completed. Therefore, the volume rendering cannot capture the color uncertainty of the low-density points. In Fig. 2 (c), the color uncertainty estimated on the cord of the microphone is diminished.

Second, we substitute  $H(x)$  for  $I(x)$  in Eq. (5) to integrate the color uncertainty of a view (Fig. 2 (d)). The conversion process of color uncertainty to an entropy form is described in Sec. 5. However,  $G(v)$  with color entropy integrates the irreducible uncertainty, which is the color uncertainty behind the surface of an object. Therefore, the thick part of an object, where the number of voxels that the camera ray traverses is large, has high uncertainty regardless of the training progression. As shown in Fig. 2 (d), the IG from the thick lump of low-density points is so high that the IG from other parts is not recognized. Finally, to compensate for each shortcoming of (c) and (d), we modify  $G(v)$  and vary the IG formulation of CSV depending on the surfaceness (Fig. 2 (e)). When a camera ray that traverses voxels does not hit a surface voxel, we sum up the color entropy of traversed voxels. When a camera ray hits a surface voxel, we use the color entropy of the surface voxel as IG. In Fig. 2 (e), IG from the head and cord of the microphone are both recognizable, which capture the information of the scene well.

## 5 Method

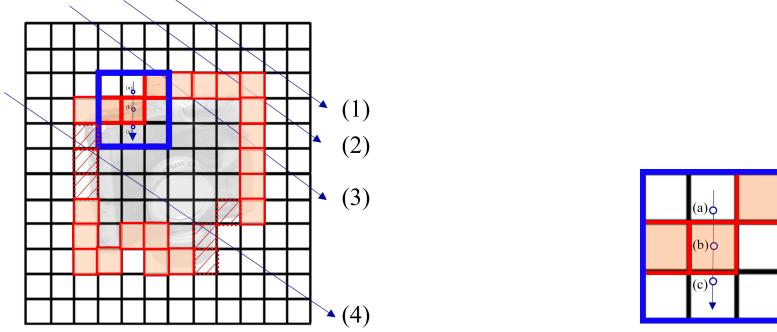
In this section, we propose a new information gain (IG) formulation for selecting the next-best view (NBV). Our IG formulation focuses on *reducible color uncertainty*, defined as the color uncertainty of a 3D point on the surface of an object. To measure reducible color uncertainty, we suggest modifying a neural implicit surface network to estimate color uncertainty along with surface information (Sec. 5.1). For a fast and efficient IG computation, we defined a colorized surface voxel (CSV), a voxel grid with corresponding color uncertainty assigned and surfaceness marked. We vary the IG formulation of CSV depending on the surfaceness to capture color uncertainty with robustness, as discussed in Sec. 4. In Sec. 5.2, we introduce the conversion of color uncertainty to color entropy, which works as a basic component for our IG formulation of CSV.

### 5.1 Estimation of Surface and Color Uncertainty

To measure *reducible* color uncertainty, we first define the color uncertainty of a scene for a neural implicit surface network. We model the color of a 3D point with a Gaussian distribution (Eq. (6)) to estimate the color uncertainty of a scene as the color variance of the 3D point.

$$c(\mathbf{r}(t)) \sim \mathcal{N}(\bar{c}(\mathbf{r}(t)), \beta^2(\mathbf{r}(t))) \quad (6)$$

The modified neural implicit surfaces network with the color in the Gaussian distribution is defined as:  $F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (g(\mathbf{x}), \bar{c}, \beta^2)$ , where  $\bar{c}$  and  $\beta^2$  represent the mean and variance of the color  $c$ , respectively. The mean and variance of a ray,  $\bar{c}(\mathbf{r}), \beta^2(\mathbf{r})$  are rendered through the volume rendering [12]. Note that the color variance  $\beta^2$ , which reflects the inconsistency in color when the viewpoint changes, does not have view dependency and does not take view direction  $\mathbf{d}$  as input. Finally, the negative log-likelihood of the probability distribution of the predicted color is incorporated as the



(a) Four types of camera rays traversing CSV grid

(b) Surface inference in a grid structure

Figure 3: The illustration of colorized surface voxel (CSV) encapsulating reconstruction target object representation. A part of CSV is highlighted with the blue box, which is enlarged to explain surface voxel inference in (b). The voxels in red are the inferred surface voxels, and those with diagonal red lines represent the surface voxels but not yet inferred. **(a)**: Camera rays (1) and (4) do not hit surface voxels while traversing the CSV grid. Camera rays (2) and (3) hit surface voxels while traversing the CSV grid. **(b)**: It demonstrates how we infer the surface voxel using estimated SDF. 3D points (a), (b), and (c) are points on the same camera ray. The distance between (a) and (c) is the step size  $1/s$  inferred from the neural implicit surface network, and (b) is the middle point.

uncertainty loss  $\mathcal{L}_u$  to optimize the mean and variance of a ray defined as follows:

$$\mathcal{L}_u = \frac{1}{M} \sum_{i=1}^M \left( \frac{\|\bar{C}(\mathbf{r}_i) - C(\mathbf{r}_i)\|_2^2}{2\mathcal{B}^2(\mathbf{r}_i)} + \frac{\log \mathcal{B}^2(\mathbf{r}_i)}{2} \right) \quad (7)$$

where  $C(\mathbf{r}_i)$  is the ground truth color from the training image and  $M$  is the number of camera rays. To derive the color uncertainty along with the surface information (Eq. (3)), the final loss function is formulated as  $\mathcal{L} = \mathcal{L}_s + \omega \mathcal{L}_u$ , with a hyperparameter  $\omega$ .

## 5.2 Information Gain using Colorized Surface Voxel

We defined a colorized surface voxel (CSV) as a voxel grid that encapsulates color uncertainty and surface information. For an efficient IG computation with CSV, we formulate IG based on  $G(v)$  in Eq. (5). We convert the color uncertainty to the color entropy to apply the formulation of  $G(v)$  as the entropy of the voxel grid defines it. As we define the color with a Gaussian probability distribution, the entropy  $H$  of the color is defined with the variance  $\beta^2$  of the color [2], as follows:

$$H(c(\mathbf{r}(t))) = -\mathbb{E}[\log \mathcal{N}(\bar{c}(\mathbf{r}(t)), \beta^2(\mathbf{r}(t)))] \quad (8)$$

$$= \frac{1}{2} \log(2\pi\beta^2) + \frac{1}{2} \quad (9)$$

Empirically,  $\beta^2(\mathbf{r}(t))$  has a value in the range of 0 to 1, so we set the initial color uncertainty value of CSV as 1 to make the entropy highest at the initialization. When we update the color uncertainty of CSV, we choose the minimum of previous uncertainty with a growth rate of 1.05 and newly predicted uncertainties. Since we estimate color uncertainty with an implicit neural network, the estimated value can change as the network weights change with training progression [14]. So, to stabilize the uncertainty value assigned to CSV, we strategically update the CSV.

After successfully assigning color uncertainty to CSV, we mark CSV with the surfaceness to vary the IG formulation depending on the surfaceness. In Fig. 3b, the 3D center point (b) of a voxel is sampled with random noise and adjacent points (a) and (c) are sampled. If the voxel containing (b) is a surface voxel, the sign of the SDF value of points (a) and (c) should be different so we can identify the surface voxel by examining the sign of the product of the SDF of two adjacent points. Note that despite the randomness in sampling point (b), the surfaceness of CSV is robust because of a suitable parameterized threshold and a strategic update. Details are described in Appendix A.3.

Finally, as depicted in Fig. 3a, we vary the IG formulation of CSV depending on the surfaceness. The simple integration of color uncertainty in CSV can integrate the irreducible color uncertainty behind

Table 1: Evaluation of image rendering (PSNR, SSIM, and LPIPS) and mesh reconstruction (Accuracy, Completeness, and Chamfer distance) on DTU. In the 2/4-image setting, 2/4 NBVs are selected at each view-selection iteration. A total of 10 and 20 images are selected with NBV selection methods in 2 and 4-image settings, respectively.

Methods	2-image setting						4-image setting					
	PSNR↑	SSIM↑	LPIPS↓	Acc.↓	Comp.↓	Chamfer↓	PSNR↑	SSIM↑	LPIPS↓	Acc.↓	Comp.↓	Chamfer↓
<b>Density (except Ours)</b>												
Random	21.17	0.769	0.238	5.166	8.680	6.923	28.14	0.896	0.101	5.084	5.757	5.420
FVS	18.48	0.714	0.287	8.861	8.043	8.452	28.48	0.896	0.105	8.677	6.223	7.450
Entropy	16.86	0.684	0.299	4.760	6.886	5.823	26.09	0.866	0.127	5.230	5.580	5.405
Color	21.78	0.786	0.229	5.862	9.955	7.909	29.56	0.911	0.090	6.345	8.269	7.307
Ours	28.19	0.867	0.168	1.829	2.176	2.002	30.10	0.910	0.116	1.715	1.967	1.864
<b>Surface</b>												
Random	27.69	0.864	0.170	2.368	3.472	2.920	29.34	0.907	0.111	1.765	2.340	2.053
FVS	26.52	0.839	0.194	3.619	4.184	3.902	29.38	0.899	0.124	2.957	3.249	3.103
Entropy	24.21	0.810	0.218	2.861	4.426	3.644	27.36	0.872	0.147	2.206	3.198	2.702
Color	26.30	0.852	0.175	2.026	2.763	2.395	28.25	0.873	0.137	2.702	2.182	2.021
Ours	28.19	0.867	0.168	1.829	2.176	2.002	30.10	0.910	0.116	1.715	1.967	1.864

the surface. Therefore, when a camera ray that traverses voxels does not hit a surface voxel (camera rays (1) and (4) in Fig. 3a), we sum up the color entropy (Eq. (9)) of traversed voxels. When a camera ray hits surface voxels (camera rays (2) and (3) in Fig. 3a), we use the color entropy of the surface voxel as IG. Our IG formulation  $G_s(v)$  is:

$$G_s(v) = \frac{1}{n} \sum_{\forall r \in \mathcal{R}_v} \sum_{\forall x \in \mathcal{X}_r \cap \mathcal{S}} H(c(x)) \quad (10)$$

Here,  $n$  represents  $|\mathcal{X}_r \cap \mathcal{S}|$  when a camera ray  $r$  hits a surface, and if it does not,  $n$  represents  $|\mathcal{X}_r|$ .

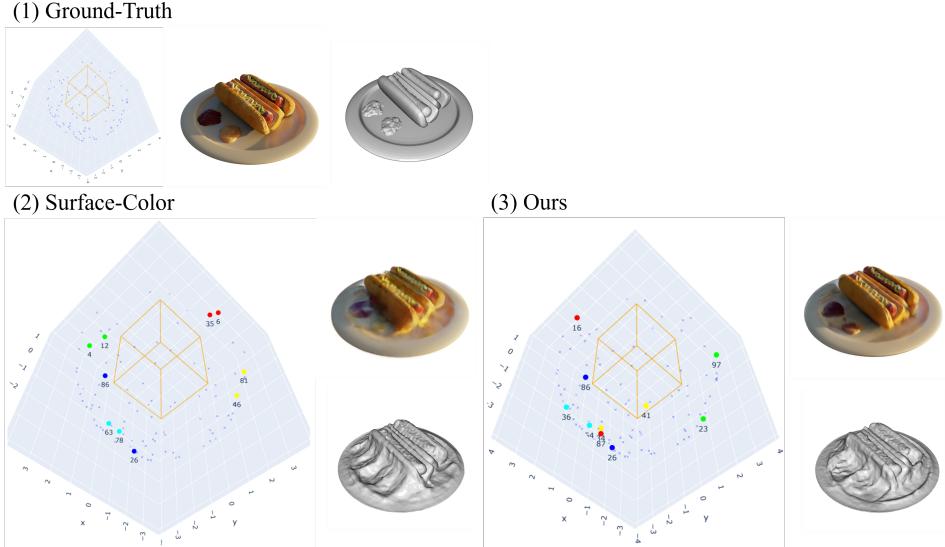
## 6 Experiment

### 6.1 Experimental Setup

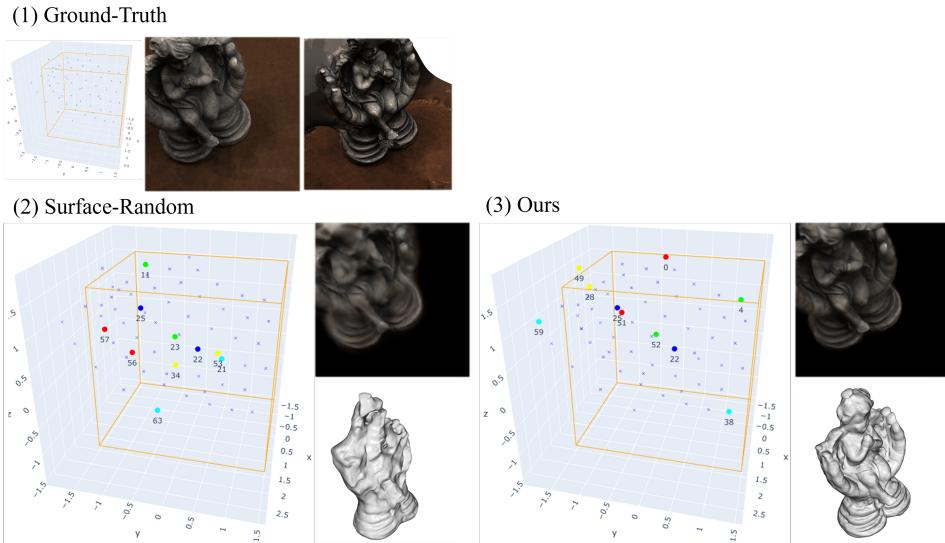
**Datasets and metrics.** We evaluate our CSV-based view selection and other NBV selection methods on three datasets using active learning schemes: 15 scenes from the DTU dataset [7], 8 scenes from the NeRF Blender Synthetic dataset (Blender) [15], and 2 scenes from our new dataset with imbalanced viewpoints. For DTU, each scene contains 49 or 64 images with an image resolution of  $384 \times 384$ , and 10 images are reserved for the test. For Blender, each scene contains 100 train images and 200 test images with an image resolution of  $800 \times 800$ , but we used 25 evenly sampled images for the test. We used the DTU and Blender dataset in the format of SDFStudio [25]. Our dataset has 80 train images and 10 test images with an image resolution of  $800 \times 800$ . We divide the train images into a train set and a train candidate set for the experiment. The train candidate set refers to all train images except those included in the train set. Image rendering qualities are evaluated using PSNR, SSIM, and LPIPS [28] scores, following the metrics used in the NeRF [15]. The reconstructed meshes are evaluated using three metrics (accuracy, completeness, and Chamfer distance) used in the ActiveRMAP [27].

**Implementation details.** We evaluate the reconstruction performance in two settings: 2-image and 4-image settings. In the 2 or 4-image setting, training begins with a train set consisting of 2 or 4 pre-designated images. After a predefined number of iterations, the model selects 2 or 4 NBV images from the train candidate set to add to the train set and removes them from the candidate set. This process is repeated four times. Afterward, view selection is stopped, and training continues until the specified number of iterations is reached. Therefore, 10 or 20 images are used in total for training in 2 or 4-image settings, respectively. Detailed experiment settings are described in Appendix A.2. We use the Adam optimizer [10] with a learning rate of 0.0005. The network is implemented in PyTorch and trained with a single NVIDIA RTX 2080 GPU for 10 hours until convergence. Detailed hyperparameter settings are described in Appendix Tab. 3.

**Comparing methods.** We test our method against four types of NBV selection methods: Random, furthest view sampling (FVS), *Entropy*, and *Color*. *Entropy* selects NBV with IG formulation  $G(v)$  (Eq. (5)). *Color* selects NBV with IG formulation defined in ActiveNeRF [17]. Since the original IG formulation of *Entropy* and *Color* are defined with uncertainty in density representation, we



(a) View selections, rendered images, and reconstructed meshes in a 2-image setting on Blender.



(b) View selections, rendered images, and reconstructed meshes in a 2-image setting on DTU.

Figure 4: View selections, rendered images, and reconstructed meshes of the two best-performing algorithms in 2-image settings are compared. In the 2-image setting, two NBVs are selected at each view-selection iteration. The results of the view selection are shown in the order of blue-cyan-green-yellow-red. The initial 2 views (shown in blue) are fixed in all NBV selection methods.

implemented the methods in two types of representations: density and surface. Further details are described in Appendix A.1.

## 6.2 Comparison

**DTU dataset.** Our method outperforms other NBV selection methods both in mesh reconstruction and image rendering, quantitatively (Tab. 1) and qualitatively. In Tab. 1, our method performs well in all criteria in both representations. Interestingly, the second-best method in density representation is *Color* in image rendering criteria and *Entropy* in mesh reconstruction criteria. Those methods perform well in the criteria that their IG formulation focuses on but perform poorly on the other criteria. As shown in Fig. 4b, Surface-Random selection can cover a wide viewpoint range but cannot

Table 2: Ablation studies on CSV-based view selection in a 2-image setting. The column  $S$  denotes how the surfaceness of CSV is utilized in IG computation (Eq. (10)).  $Y$  means the NBV selection methods use the same IG formulation as  $G_s(v)$  and  $N$  means the methods change the integration target to  $\forall x \in \mathcal{X}_r$  in  $G_s(v)$ , which the surfaceness is not considered.

$S$	Blender			DTU					
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	Acc.↓	Comp.↓	Chamfer↓
Y	<b>21.22</b>	<b>0.854</b>	<b>0.170</b>	<b>28.19</b>	<b>0.867</b>	<b>0.168</b>	<b>1.829</b>	<b>2.176</b>	<b>2.002</b>
N	20.13	0.844	0.181	27.18	0.855	0.180	2.060	2.441	2.251

focus on incompletely reconstructed parts. As a result, Surface-Random cannot reconstruct the upper part of the reconstruction target. Note that we trained all NBV methods with object masks applied to the DTU dataset for a fair evaluation. If we evaluate methods on DTU without masks, surface-based methods should handle the uncertainties from two different networks (NeRF and NeuS) because the surface-based methods additionally need a background model (*e.g.*, NeRF) for background training [21]. Therefore, this paper does not cover the method for combining and utilizing uncertainties from different networks, which would be a great future work.

**Blender dataset.** We evaluated the effectiveness of image selection on image rendering quality and mesh reconstruction quality in Fig. 4a. The result of Surface-Color shows that some parts of objects are cloudy, which occurs when the model does not have enough color information on those parts because of the lack of diverse viewpoints. The quantitative result and qualitative results on other settings can be found in Appendix A.4.

**Time comparison.** We compared the time to select the next-best views (NBV) in the 2-image setting in Blender. The NBV selection times are averaged over eight scenes (Fig. 5). Our method (0.8 s) has significantly improved time consumption compared to *Color* (10 s) and has a similar time consumption to *Entropy* (0.5 s).

**Imbalanced viewpoint dataset.** We collected a 3D reconstruction dataset of complex and occluded objects with imbalanced viewpoints. In the real world, it is hard to collect views for 3D reconstruction evenly from all directions. Some images from certain viewpoints have a larger portion in the entire dataset than other viewpoints. In this situation, NBV selection methods like Random tend to select views from certain viewpoints a lot. We construct the imbalanced viewpoint dataset that has a limited number of views that can observe all geometry and color information about the scene. We compared NBV selection methods in our new dataset. We show the result of the shelf scene in Fig. 6. The only method that renders 4 objects with the right color is Ours. Also, Ours shows a higher PSNR value evaluated on the test set than other methods. More results and detailed explanations about our new dataset are described in Appendix A.6.

### 6.3 Ablation studies

We analyze the effects of considering surfaceness in IG computation on NBV selection. We evaluate the image rendering and mesh reconstruction performance. In Tab. 2, the NBV selection methods that consider the surfaceness of CSV outperforms the methods that do not consider the surface in every metric. This result shows that considering the surface helps in the selection of more informative views. The qualitative results are shown in Appendix A.4.

## 7 Conclusion

We have presented CSV-based view selection, an effective next-best view selection method that utilizes color uncertainty of 3D points with a surface voxel-based approach. The implicit nature of neural network-estimated color uncertainty makes assigning its value to a grid-like data structure challenging. Nonetheless, we address such a challenge using the strategic grid update. Also, CSV-

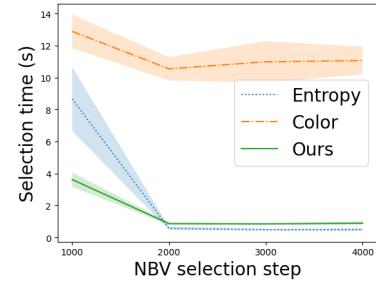


Figure 5: NBV select. time

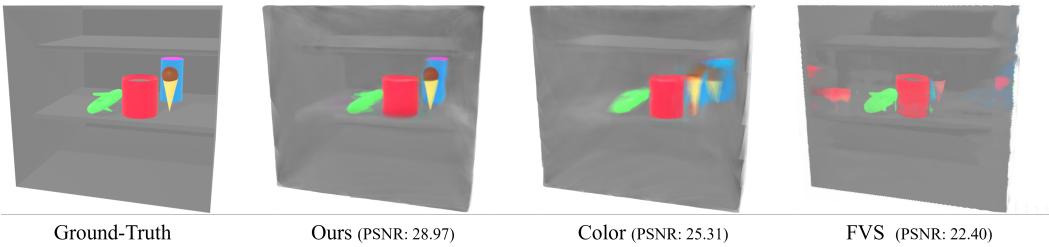


Figure 6: Comparison of NBV selection methods on the shelf scene with occluded objects. The views that can see 4 objects are limited.

based view selection applies a different aggregation strategy depending on the presence of the surface voxel, which enables the utilization of color uncertainty 3D-wise. We show that CSV-based view selection improves mesh reconstruction and image rendering qualities compared to other methods.

## References

- [1] Chen, S., Li, Y., and Kwok, N. M. (2011). Active vision in robotic systems: A survey of recent developments. *The International Journal of Robotics Research*, 30(11):1343–1377.
- [2] Cover, T. M. (1999). *Elements of information theory*. John Wiley & Sons.
- [3] Delmerico, J., Isler, S., Sabzevari, R., and Scaramuzza, D. (2018). A comparison of volumetric information gain metrics for active 3d object reconstruction. *Autonomous Robots*, 42(2):197–208.
- [4] Feng, Z., Zhan, H., Chen, Z., Yan, Q., Xu, X., Cai, C., Li, B., Zhu, Q., and Xu, Y. (2024). Naruto: Neural active reconstruction from uncertain target observations. *arXiv preprint arXiv:2402.18771*.
- [5] Gropp, A., Yariv, L., Haim, N., Atzmon, M., and Lipman, Y. (2020). Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*.
- [6] Isler, S., Sabzevari, R., Delmerico, J., and Scaramuzza, D. (2016). An information gain formulation for active volumetric 3d reconstruction. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3477–3484. IEEE.
- [7] Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., and Aanæs, H. (2014). Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 406–413.
- [8] Jin, L., Chen, X., Rückin, J., and Popović, M. (2023). Neu-nbv: Next best view planning using uncertainty estimation in image-based neural rendering. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11305–11312.
- [9] Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP ’06, page 61–70, Goslar, DEU. Eurographics Association.
- [10] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- [11] Lee, S., Chen, L., Wang, J., Liniger, A., Kumar, S., and Yu, F. (2022). Uncertainty guided policy for active robotic 3d reconstruction using neural radiance fields. *IEEE Robotics and Automation Letters*, 7(4):12070–12077.
- [12] Levoy, M. (1990). Efficient ray tracing of volume data. *ACM Transactions on Graphics (TOG)*, 9(3):245–261.
- [13] Li, R., Gao, H., Tancik, M., and Kanazawa, A. (2023a). Nerfacc: Efficient sampling accelerates nerfs. *arXiv preprint arXiv:2305.04966*.

- [14] Li, Z., Müller, T., Evans, A., Taylor, R. H., Unberath, M., Liu, M.-Y., and Lin, C.-H. (2023b). Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8465.
- [15] Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2021). Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106.
- [16] Müller, T., Evans, A., Schied, C., and Keller, A. (2022). Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15.
- [17] Pan, X., Lai, Z., Song, S., and Huang, G. (2022). Activenerf: Learning where to see with uncertainty estimation. In *European Conference on Computer Vision*, pages 230–246. Springer.
- [18] Ran, Y., Zeng, J., He, S., Chen, J., Li, L., Chen, Y., Lee, G., and Ye, Q. (2023). Neurar: Neural uncertainty for autonomous 3d reconstruction with implicit neural representations. *IEEE Robotics and Automation Letters*, 8(2):1125–1132.
- [19] Sun, C., Sun, M., and Chen, H.-T. (2022). Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469.
- [20] Thrun, S. (2002). Probabilistic robotics. *Communications of the ACM*, 45(3):52–57.
- [21] Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., and Wang, W. (2021). Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*.
- [22] Yan, D., Liu, J., Quan, F., Chen, H., and Fu, M. (2023). Active implicit object reconstruction using uncertainty-guided next-best-view optimziation. *arXiv preprint arXiv:2303.16739*.
- [23] Yang, J., Pavone, M., and Wang, Y. (2023). Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8254–8263.
- [24] Yu, A., Ye, V., Tancik, M., and Kanazawa, A. (2021). pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587.
- [25] Yu, Z., Chen, A., Antic, B., Peng, S., Bhattacharyya, A., Niemeyer, M., Tang, S., Sattler, T., and Geiger, A. (2022a). Sdfstudio: A unified framework for surface reconstruction.
- [26] Yu, Z., Peng, S., Niemeyer, M., Sattler, T., and Geiger, A. (2022b). Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in Neural Information Processing Systems (NeurIPS)*.
- [27] Zhan, H., Zheng, J., Xu, Y., Reid, I., and Rezatofighi, H. (2022). Activemap: Radiance field for active mapping and planning. *arXiv preprint arXiv:2211.12656*.
- [28] Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595.

Table 3: Details in network architectures that encode surface and density representation and ActiveNeRF [17].

	<b>Networks</b>	Surface	Density	ActiveNeRF [17]
<b>SDF network</b>	MLP hidden layer	8	8	8
	MLP size	256	256	256
	Activation	Softplus	ReLU	ReLU
	positional encoding	6	10	10
<b>Color network</b>	skip connection layer	4	4	4
	MLP hidden layer	4	1	1
	MLP size	256	128	128
<b>Hyper-parameters</b>	direction encoding	4	4	4
	RGB loss	1	1	1
	eikonal loss	0.1	-	-
	uncertain loss (RGB)	0.001	0.001	1
	uncertain loss (beta)	0.01	0.01	0.5
	uncertain loss (sigma)	0.0	0.0	0.01
	beta min	0.001	0.001	0.01
Batch size		512	512	1024
Learning rate		$5 \times 10^{-4}$	$5 \times 10^{-4}$	$5 \times 10^{-4}$

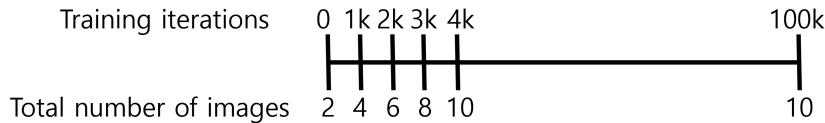


Figure 7: Illustration of the active learning scheme in the 2-image setting.

## A Experimental details and results

### A.1 Comparison of methods

First, we elaborate on the differences in network architecture (Tab. 3) among ActiveNeRF [17] and neural networks encoding surface representation and density representation. In Tab. 3, we used the NeuS [21] architecture for models with surface representation and the NeRF [15] architecture for networks with density representation. Except for ActiveNeRF [17], all networks are implemented based on SDFstudio [25]. We used the code in SDFstduo that uses the grid sampling, which refers to the accelerated grid sampling utilized by NerfAcc [13] and Instant NGP [16]. The grid sampling method selects 3D points by referencing an occupancy grid. During ray marching, the sample point within an occupancy grid cell with a lower occupancy probability than a given threshold is skipped. Therefore, with grid sampling, a variable number of samples are chosen in each ray. We employed grid sampling for efficient and rapid training.

Now, we elaborate on details in NBV selection methods: Color, Entropy, and FVS. In the Color method, we used IG formulation from ActiveNeRF [17]. So, we employed a different sampling method in IG calculation, which samples a constant number of points in a ray as in ActiveNeRF. In the Entropy method in surface representation, we substitute  $\alpha$  in Eq. (2) for  $p(x)$  in Eq. (4). As  $p(x)$  is the occupancy probability of a voxel, its value is in the range of 0 to 1. In the same way,  $\alpha$  is 0 when a voxel is not a surface voxel and 1 when a voxel belongs to a surface. Finally, we similarly implemented the FVS method with Algorithm 1 in Appendix A.5. Instead of sorting the candidate views  $C$  using IG values ( $G_s$ ), FVS sorts candidate views only with distances.

### A.2 Experiment settings

In this subsection, we provide detailed explanations about the active learning scheme (Fig. 7), including the number of total iterations, selection intervals, the number of steps for frequency

Table 4: Evaluation of image rendering (PSNR, SSIM, and LPIPS) on Blender. In the 1/2/4-image setting, 1/2/4 NBVs are selected at each view-selection iteration. 5, 10, and 20 images are selected using NBV selection methods in 1, 2, and 4-image settings.

Methods	1-image setting			2-image setting			4-image setting		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
<b>Density (except Ours)</b>									
Random	11.05	0.744	0.442	12.34	0.759	0.404	26.68	0.909	0.089
FVS	11.28	0.747	0.435	11.72	0.743	0.441	26.77	0.911	0.088
Entropy	11.45	0.752	0.423	12.70	0.779	0.383	25.71	0.899	0.100
Color	13.09	0.776	0.362	16.64	0.807	0.270	18.82	0.822	0.261
ActiveNeRF [17]	-	-	-	20.01	0.832	0.204	26.24	0.856	0.124
ActiveNeRF (repro.)	-	-	-	14.78	0.792	0.251	17.00	0.820	0.200
Ours	16.63	0.809	0.265	21.22	0.854	0.170	26.08	0.907	0.106
<b>Surface</b>									
Random	15.88	0.812	0.268	16.24	0.831	0.268	21.79	0.874	0.151
FVS	16.07	0.808	0.270	19.40	0.845	0.194	21.62	0.873	0.155
Entropy	13.78	0.801	0.307	15.41	0.818	0.291	21.28	0.869	0.159
Color	15.93	0.809	0.270	19.25	0.841	0.208	23.91	0.884	0.127
Ours	16.63	0.809	0.265	21.22	0.854	0.170	26.08	0.907	0.106

regularization, and the duration of the warm-up stage. The total training iterations in the Blender dataset’s 1, 2, and 4-image settings are  $50k$ ,  $100k$ , and  $200k$ , respectively. For the DTU dataset, we trained models for  $60k$  iterations in the 2-image setting and  $120k$  iterations in the 4-image setting. In terms of selection intervals, in the 2-image setting, the model selects the next-best views (NBV) for every  $1k$  iteration (*e.g.*,  $[1k, 2k, 3k, 4k]$ ), and in the 4-image setting, NBV selection occurs every  $2k$  iteration (*e.g.*,  $[2k, 4k, 6k, 8k]$ ). Similarly, the model selects the NBV for every  $0.5k$  iteration in the 1-image setting, (*e.g.*  $[0.5k, 1k, 1.5k, 2k]$ ). As the model selects NBVs four times, the total numbers of train images are 5, 10, and 20 in 1, 2, and 4-image settings, respectively.

We applied frequency regularization in the 1-and 2-image settings, as suggested in FreeNeRF [23]. Given the challenge of learning from a few images and inferring the next-best view, frequency regularization helped the model choose useful images by initially evaluating them with low frequency and gradually transitioning to high frequency. Following a setup similar to FreeNeRF [23], the frequency regularization ends at  $40k$  iterations (40% of total iterations) when the model is trained on 2-image setting Blender,  $30k$  iterations (60% of total iterations) on 1-image setting Blender, and at  $30k$  iterations (50% of total iterations) for models trained on 2-image setting DTU.

We used two warm-up stages, one from the NeuS [21] framework and the other from sampling strategies. According to NeuS [21], training a model to converge and learn a surface is challenging, so setting the learning rate of surface representation networks to linearly warm-up helps training. Therefore, we implemented the NeuS networks’ learning rate to increase from 0 to  $5 \times 10^{-4}$  in the first 500 iterations in the 1 and 2-image settings and 1000 iterations in the 4-image setting.

During the warm-up stage in sampling strategies, a constant number of points in a ray are sampled instead of grid sampling for sampling points from various locations in the scene. Additionally, all cells in the occupancy grid are updated. After the warm-up stage, 3D points are sampled with grid sampling, and the occupancy grid is partially updated by selecting  $n_o$  occupied cells and  $n_r$  randomly sampled cells, where  $n_o$  and  $n_r$  are parameters. It is important to note that the warm-up stage is lengthened if the model is trained on more images in different settings.

### A.3 Surfaceness implementation

The surfaceness estimation in the CSV grid is implemented with Nerfacc [13] grid update function. In the grid update step, the center point of a voxel with random noise is examined to output an SDF value of the point. Then, two adjacent points as described in Fig. 3b are sampled to examine the sign of the product of their SDF values. When the sign is negative and the outer point like (*a*) in Fig. 3b has a positive SDF value, the grid is updated to 1, and the other grids are updated to 0. The decay factor 0.95 and the warm-up steps mentioned above are applied during the update. For IG computation in Eq. (10), we defined the surface by filtering the surfaceness value from CSV using the threshold 0.8.

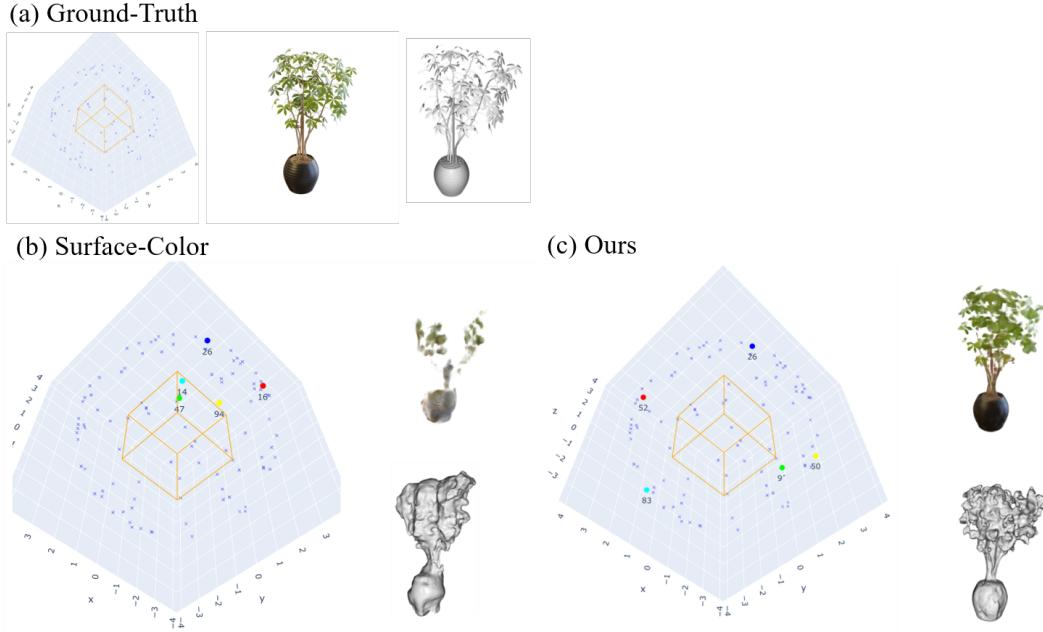


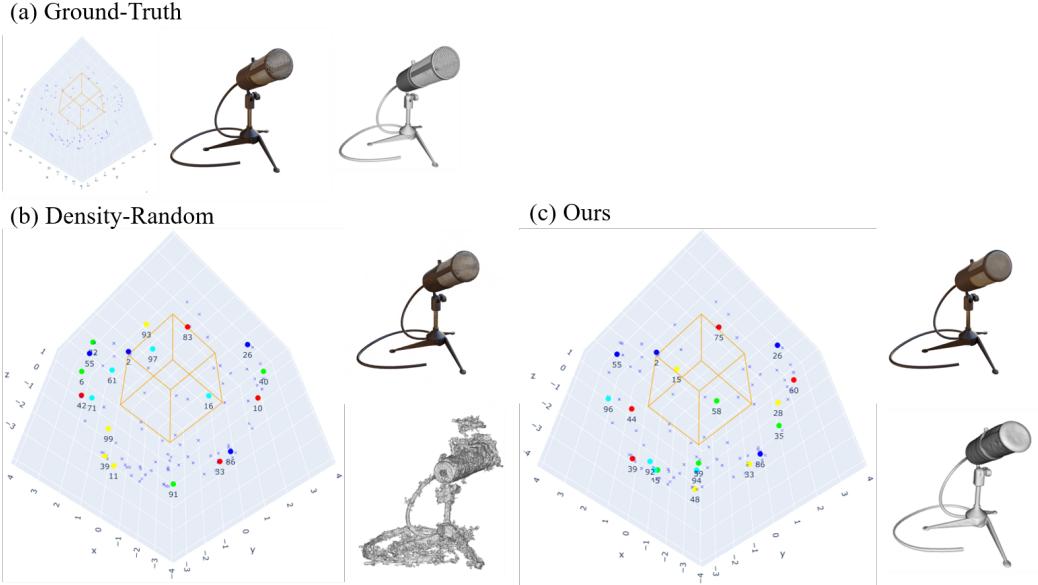
Figure 8: View selections, rendered images, and reconstructed meshes of the two best-performing algorithms in 1-image settings are compared. In the 1-image setting, 1 NBV is selected at each view-selection iteration. The results of the view selection are shown in the order of blue-cyan-green-yellow-red. The initial 1 view (shown in blue) is fixed in all NBV selection methods.

#### A.4 More results

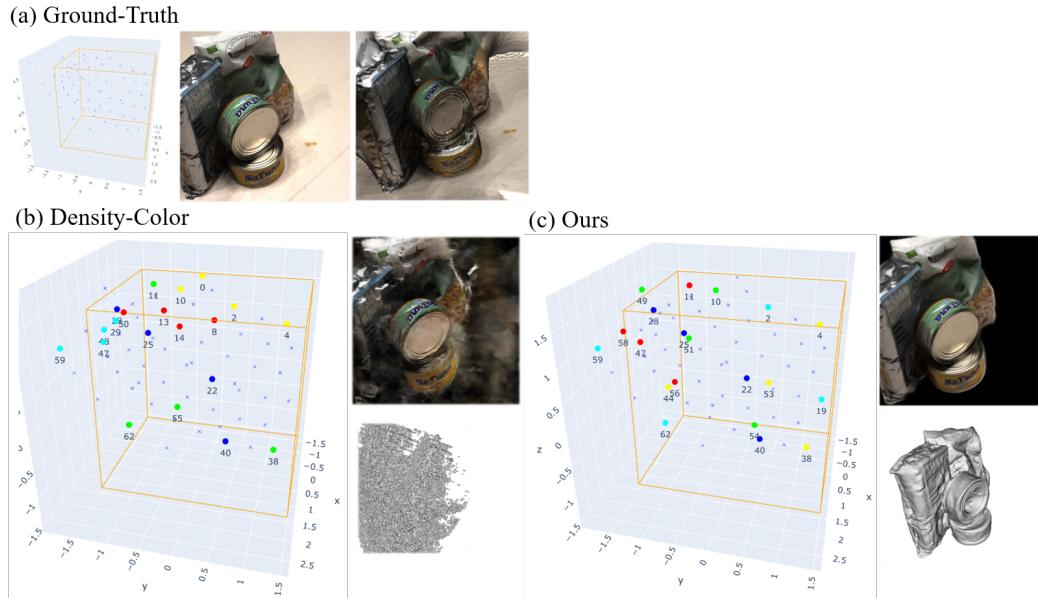
We report quantitative results on Blender in Tab. 4. Ours outperformed other NBV selection methods in 1 and 2-image settings. Shown in Appendix A.2, the Surface-Color method selects views from a certain region, resulting in incomplete image rendering and reconstruction in other views. In the 4-image setting, the Random and FVS methods in density representation outperformed ours in image rendering criteria. However, in Fig. 9a, the mesh reconstruction of the Density-Random is poor as the density representation does not infer an explicit surface. Also, the total number of train images is large enough for the random selection to cover the hemisphere, *the benefit of effective image selection is reduced*.

In the DTU dataset, the Density-Color method performed similarly to ours in image rendering criteria. However, in Fig. 9b, Density-Color showed poor mesh reconstruction results. Also, Density-Color selected views with high IG values without considering distances between views, which can disrupt the diversity in view selection. During the experiment with the DTU dataset, we excluded 10 test images from the training set from the start, but they are still shown on the camera poses visualization of Ground Truth in DTU-related figures. The mesh reconstruction of the DTU dataset, composed of point clouds, was performed using Kadzhdan *et al.* [9]. We use the same Chamfer distances evaluation used in MonoSDF [26].

We investigated the effect of considering the distance between selected  $k$  NBVs in Tab. 5. The NBV selection that considers the distance worked better in every criterion than the selection that does not consider the distance. The detailed strategy of considering the distance is explained in Appendix A.5. We also report qualitative results in ablation studies (Fig. 10, Fig. 11). In the DTU dataset (Fig. 10), we demonstrate that selecting camera views by referencing the surface information leads to diverse camera selection across scenes. As DTU has a smaller range of camera views than Blender,  $N\text{-}S$  have similar views across scenes, while  $Y\text{-}S$  exhibit diverse selections. In the Blender dataset (Fig. 11), the chair exhibits bulky noise between the armrests in  $Y\text{-}S + Top\text{-}K$ . This type of error frequently occurs when the camera views are inadequately chosen. Also, spreading out the camera views is important, especially when reconstructing objects with light reflections, as shown in Blender materials.



(a) results in a 4-image setting on Blender.



(b) results in a 4-image setting on DTU.

Figure 9: View selections, rendered images, and reconstructed meshes of the two best-performing algorithms in 1/4-image settings are compared. In the 4-image setting, 4 NBVs are selected at each view-selection iteration. The results of the view selection are shown in the order of blue-cyan-green-yellow-red. The initial 4 views (shown in blue) are fixed in all NBV selection methods.

### A.5 Multiple next-best view selection

Selecting multiple NBVs becomes essential when acquiring diverse perspectives within limited computational resources and time constraints. Without additional criteria, simply choosing the top  $k$  candidates based on the IG value can lead to the selection of similar views from specific regions of the camera sphere. This redundancy may limit the diversity of information. To address this issue, we introduce a selection strategy to ensure that the selected  $k$  images are sufficiently spaced apart. Algorithm 1 outlines the procedure for selecting these  $k$  candidates.

---

**Algorithm 1** Multiple next-best view selection

---

- 1: **Input:** candidate camera index array  $C$ , train camera index array  $T$ , camera pose array  $P$ , information gain  $G_s$ , distance threshold  $\tau$ , the number to select  $K$
- 2: **Output:** next-best view index array  $A$
- 3: select  $idx$  from  $C$  that has the maximum  $G_s$  value
- 4: remove  $idx$  from  $C$  and add  $idx$  in  $A$
- 5: **for**  $i \leftarrow 1$  to  $K - 1$  **do**
- 6:     add  $idx$  in  $T$  and descending sort  $C$  by the  $G_s$  value
- 7:     compute the pairwise distance matrix  $D$  between  $P[T]$  and  $P[C]$
- 8:     get a boolean matrix  $B_C = (D \geq \tau)$  and do "AND" operation along  $T$
- 9:     **while**  $B_C$  is all False along  $C$  **do**
- 10:          $\tau \leftarrow \tau \times 0.95$  ▷ Lower the distance threshold
- 11:          $B_C = (D \geq \tau)$  and do "AND" operation along  $T$  ▷ Filter again with the relaxed criteria
- 12:          $idx = C[B_C][0]$  ▷ Select the first element in sorted  $C$  filtered with the distance
- 13:         remove  $idx$  from  $C$  and add  $idx$  in  $A$

---

Table 5: Ablation studies on CSV-based view selection in a 2-image setting. The column  $K$  denotes how  $k$  NBVs are selected. After sorting train candidate views by IG value, *Top* selects the  $k$  views with the highest IG values, and *Dist.* selects the  $k$  views that are more than a certain distance from existing train views. The column  $S$  denotes how the surface status of CSV is utilized in IG computation (Eq. (10)). *Y* means the NBV selection methods use the same IG formulation as  $G_s(v)$  and *N* means the methods change the integration target to  $\forall x \in \mathcal{X}_r$  in  $G_s(v)$ , which the surface status is not considered.

Blender			DTU								
$S$	$K$		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	Acc.↓	Comp.↓	Chamfer↓
Y	Dist.		<b>21.22</b>	<b>0.854</b>	<b>0.170</b>	<b>28.19</b>	<b>0.867</b>	<b>0.168</b>	<b>1.829</b>	<b>2.176</b>	<b>2.002</b>
	Top		18.48	0.834	0.212	27.56	0.858	0.174	2.041	2.508	2.274
N	Dist.		20.13	0.844	0.181	27.18	0.855	0.180	2.060	2.441	2.251
	Top		19.94	0.842	0.187	25.64	0.836	0.193	2.127	2.613	2.370

Algorithm 1 outlines selecting multiple next-best views that cover a wider range of XYZ coordinates. The main idea is to select views with high IG value while considering the distance from the already incorporated train views. A view is selected if it has a longer distance than the distance threshold  $\tau$  from all train views and has the highest IG value among the satisfied candidates. We empirically set  $\tau$  to 1.732, and when no candidate satisfies the distance threshold, the threshold decreases by multiplying it with the decay factor 0.95.

#### A.6 New dataset with imbalanced viewpoints

We collected a new dataset that observes complex and occluded objects with imbalanced viewpoints. In Fig. 12, we explain how imbalanced the dataset is. We collected a total of 80 train images and 10 test images. The train images consist of 60 common views, 10 high-angle views, and 10 low-angle views. The test images consist of 4 high-angle views, 2 low-angle views, and 4 common views. Since the data collection in the real world tends to have an imbalance of viewpoints in data, we collected a new dataset reflecting the situation.

In Fig. 13a, the FVS method selects views in both ends as the method prefers the furthest view from existing ones. Since views on both ends cannot observe 4 occluded objects simultaneously, the FVS method fails to reconstruct 4 objects. On the other hand, the Color method selects relatively diverse views, but the selected views are not spaced apart enough to infer depth and make targets converge. Ours successfully reconstructed 4 objects with accurate colors.

In Fig. 13b, the Random method selected most views from common viewpoints, so it could not reconstruct 2 holes in each outlet. Also, the Entropy method could not reconstruct the yellow part in the outlet. It may not converge well with the imprecise initialization, reconstructed with a selected input sequence. Conversely, Ours reconstructs the yellow part and two holes well with the appropriate view selection.

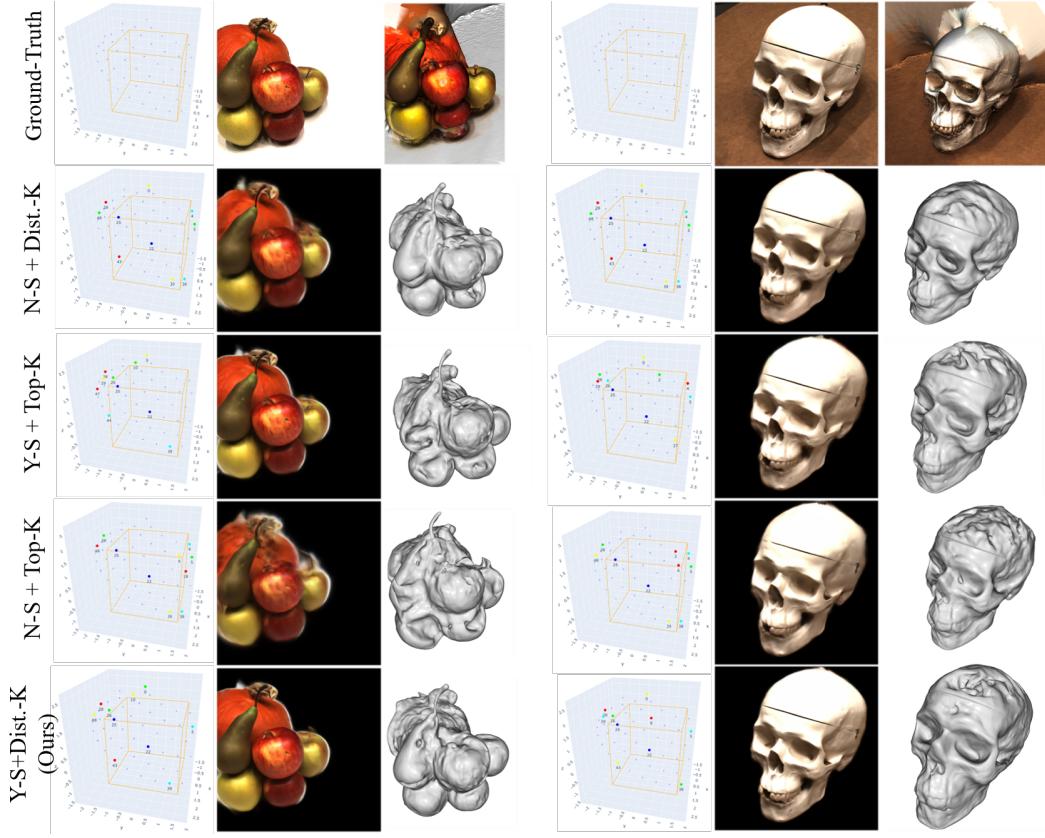


Figure 10: Qualitative results in ablation studies on DTU. The selected views, rendered RGB images, and reconstructed meshes are shown from left to right. The ablation studies are conducted in a 2-image setting, so 2 NBVs are selected at each view-selection iteration. The results of the view selection are shown in the order of blue-cyan-green-yellow-red. The initial 2 views (shown in blue) are fixed in all NBV selection methods.

## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [Yes]

Justification: We clarified the contribution at the end of the Introduction.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

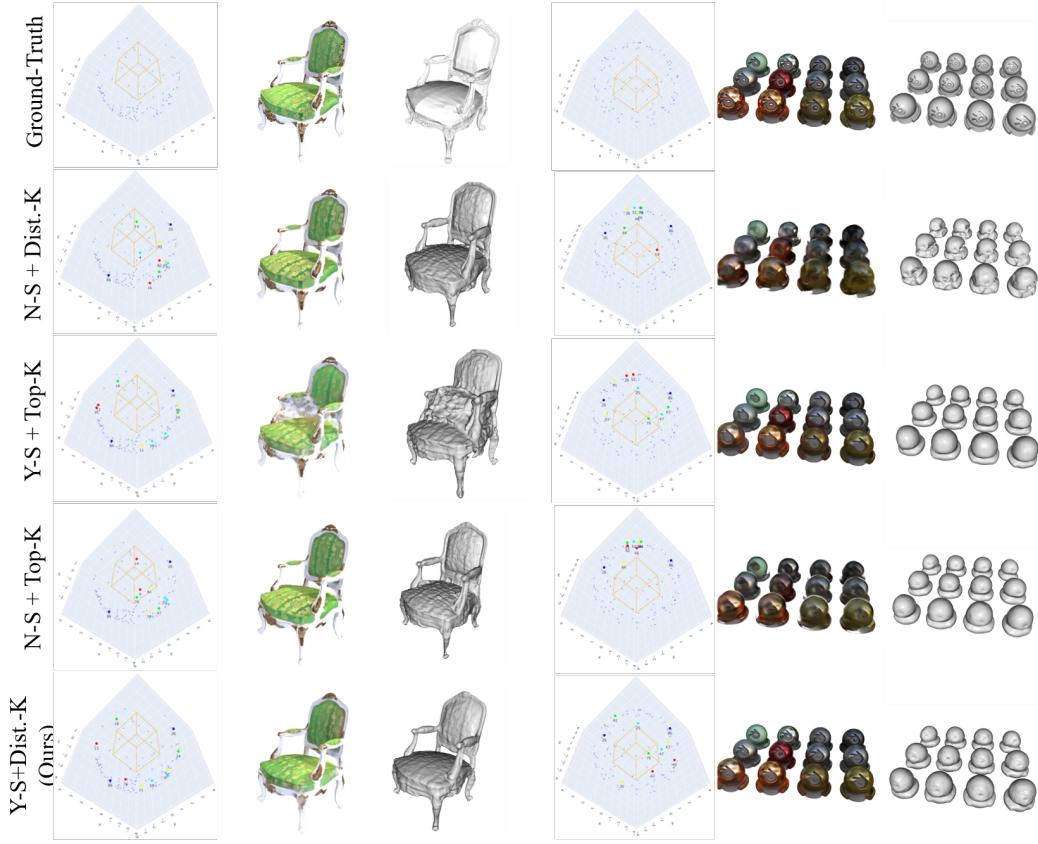


Figure 11: Qualitative results in ablation studies on Blender. The view selections, rendered RGB images, and reconstructed meshes are shown from left to right. The ablation studies are conducted in a 2-image setting, so 2 NBVs are selected at each view-selection iteration. The results of the view selection are shown in the order of blue-cyan-green-yellow-red. The initial 2 views (shown in blue) are fixed in all NBV selection methods. Our selection of camera views covers wider views, including top and 360-degree views, resulting in more accurate rendering and meshes.

Justification: We provide the limitation on Line 254.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.

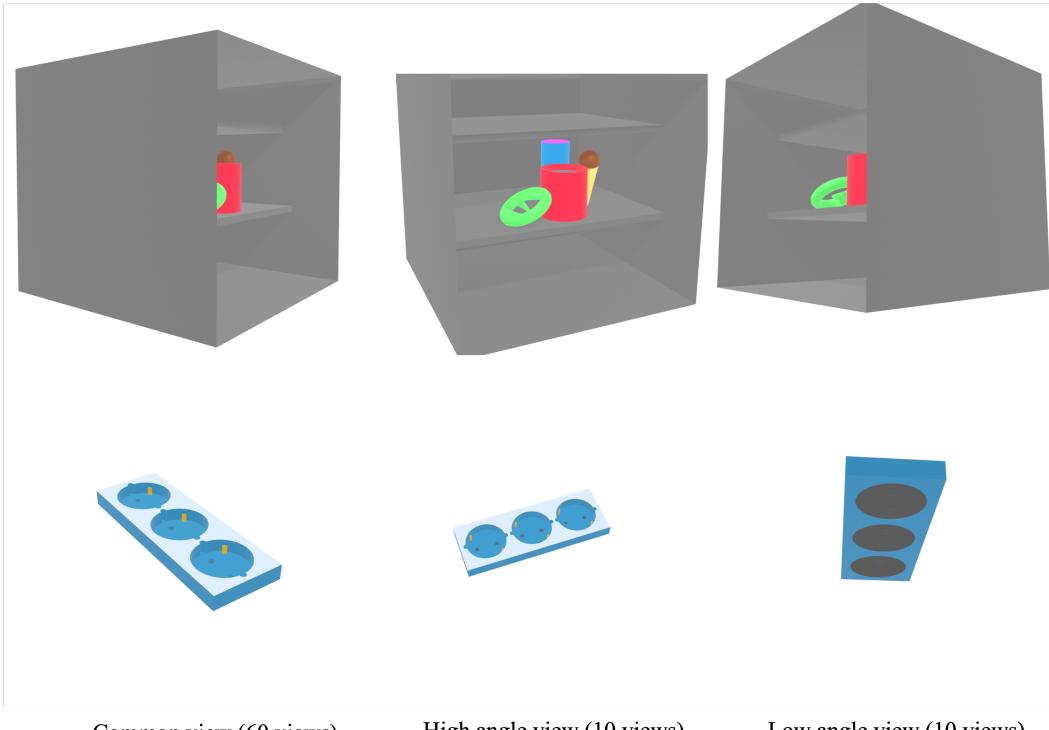


Figure 12: The imbalanced viewpoints of our new dataset.

- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

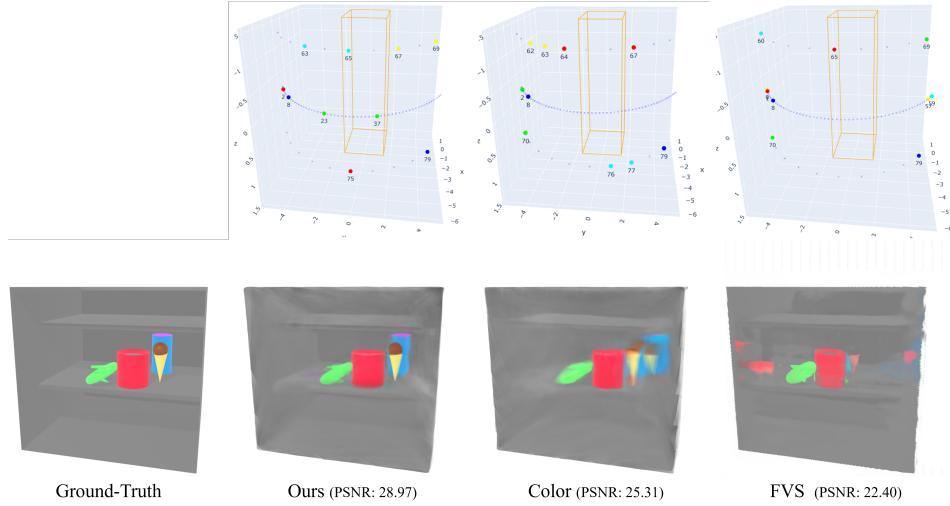
Answer: [NA]

Justification: We do not have theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

### 4. Experimental Result Reproducibility



(a) View selections and rendered images of *shelf* in a 2-image setting.

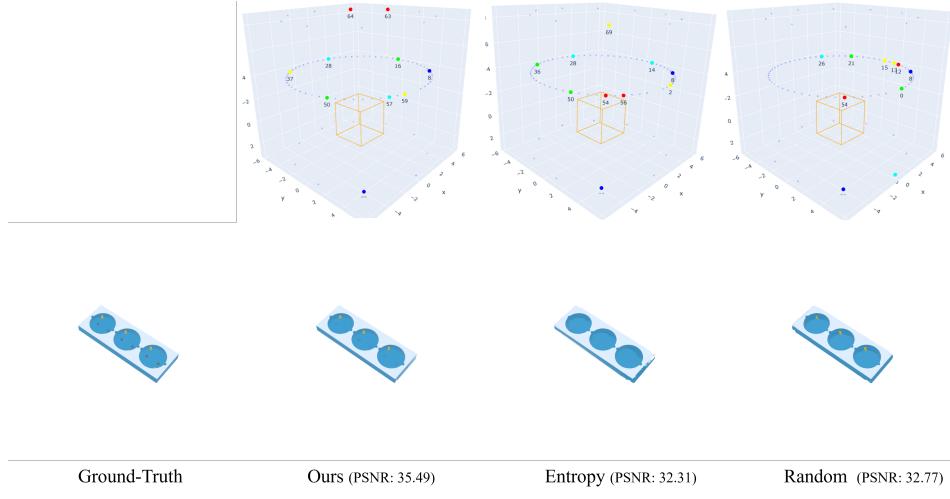


Figure 13: View selections, rendered images, and PSNR metrics of the three best-performing NBV selections in 2-image settings are compared. In the 2-image setting, two NBVs are selected at each view-selection iteration. The results of the view selection are shown in the order of blue-cyan-green-yellow-red. The initial 2 views (shown in blue) are fixed in all NBV selection methods.

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide the explanation of the training scheme and hyperparameters in Sec. 6.1 and Appendix A.2.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.

- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

## 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We do not provide open access yet at submission time, but we have a plan to provide open access when we can reveal our identity. We submit our code and data as supplementary materials.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: In Sec. 6.1, we provide details about data splits, hyperparameters, and the type of optimizer.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We provide the standard deviation of averaged data in Fig. 5.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

## 8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide the type of compute workers and the time of execution in Sec. 6.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

## 9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We use popular datasets that are actively used.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

## 10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: This work deals with the view selection methods in active 3D reconstruction. There is no way that humans intercept and modify the training process, yet.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

## 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We do not have data or models that have a high risk for misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

## 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: In Sec. 6.1, we referenced the original paper of DTU and Blender dataset, and we mentioned the format of the datasets is SDFStudio [25] version. The URL is <https://github.com/autonomousvision/sdfstudio>.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](http://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

## 13. New Assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We introduce the new dataset in Sec. 6 and explain the details of the dataset in Appendix A.6. We will provide the documentation alongside the dataset when we make it public.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

## 14. Crowdsourcing and Research with Human Subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not include experiments with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

**15. Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper does not include experiments with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.