NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction

Anonymous Author(s)
Affiliation
Address
email

Abstract

We present a novel neural surface reconstruction method, called *NeuS*, for reconstructing objects with high fidelity from 2D image inputs. Existing neural reconstruction approaches, such as DVR [21] and IDR [31], suffer from being easily trapped in a local minima, struggle with the objects with severe self-occulusions, abrupt depth changes and thin structures, and require foreground masks as supervision, due to the local optimization process in surface rendering used. On the other hand, volume rendering based methods have shown the robustness to these challenging cases; however, extracting high-quality surfaces from such a learned implicit representation is difficult because there are no sufficient constraints on the surface geometry to enforce a continuous zero level set. To address these issues, we represent 3D geometry as Signed Distance Function (SDF) and develop a new volume rendering method to train the neural SDF representation to ensure that the contribution to the ray color in color accumulation process is mostly from the first intersection of the camera ray with the surface. Experiments on the DTU dataset and the BlendedMVS dataset show that NeuS outperforms the state-of-the-arts in high-quality surface reconstruction, especially for the objects with complex structures and self-occlusion.

1 Introduction

Reconstructing surface geometry from multi-view images is a fundamental problem in computer vision and computer graphics. 3D reconstruction with neural implicit representations has recently become a highly promising alternative to classical reconstruction approaches [27, 6, 2] due to its high reconstruction quality and the potential to handle challenging cases that are difficult for classical approaches, such as non-lambertian surfaces and thin structures. Recent works model 3D geometry as SDF [31, 34, 14] or occupancy [21, 22]. To train their neural models, these methods use a differentiable surface rendering method to render a 3D object into images and use the render images to compare against input images for supervision. For example, IDR [31] produces impressive reconstruction results but it fails to reconstruct objects with complex structures that cause abrupt depth changes. The cause of this limitation is that the surface rendering method used in IDR only considers a single surface intersection point for each ray. Consequently, the gradient only exists at this single point, which is too local for effective back propagation and would get optimization stuck in local optima when there are abrupt changes of depth on images. As illustrated in Fig. 1 (a) top, with the radical depth change caused by the hole, the network would incorrectly predict the points near the front surface to be blue, failing to find the far-back blue surface. The actual test example in Fig. 1 (b) shows that IDR fails to correctly reconstruct the surfaces near the edges with abrupt depth changes.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Recently, NeRF [20] and its variants have explored to use a volume rendering method to learn a volume density field for novel view synthesis. This volume rendering approach samples multiple points along each ray and accumulates the alpha values and colors of the sampled points to produce the output pixel colors for training purposes. The advantage of the volume rendering approach is that it can handle abrupt depth changes, because it considers multiple points along the ray and so all the sample points, either on the near surface or on the far-surface, produce gradient signals for back propagation. For example, referring Fig. 1 (a) bottom, when the near surface (yellow) is found to have inconsistent colors with the input image, the volume rendering approach is capable to train the network to find the far-back surface to produce the correct scene representation. However, the volume rendering approach in NeRF is only designed for a volume density field (i.e. radiance field) which is not a proper surface representation. In fact, it is difficult to extract a high-quality surface from such the density field. For example, Fig. 1 (b) shows a surface extracted as a level-set surface of the density field computed by NeRF. Although the surface correctly accounts for abrupt depth changes, it contains conspicuous noise in the planar regions.

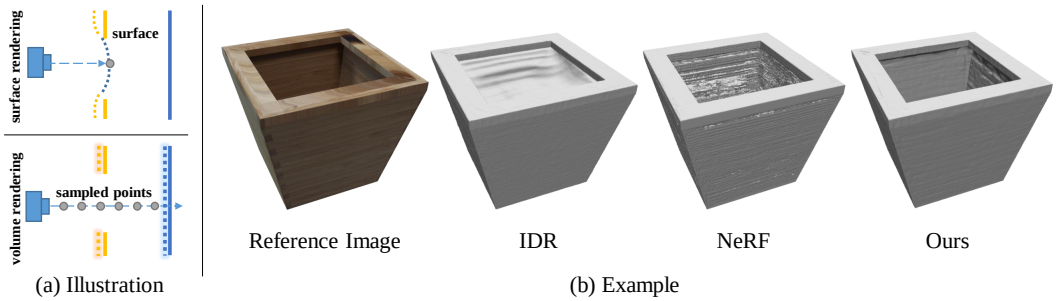


Figure 1: (a) Illustration of the surface rendering and volume rendering. (b) A toy example of bamboo planter, where there are occlusions on the top of the planter.

We present a new neural rendering scheme, called *NeuS*, for multi-view surface reconstruction that integrates the Signed Distance Function (SDF) for surface representation and the volume rendering approach to improve robustness of training a neural SDF representation for complex objects. By introducing a density distribution induced by SDF, our method enables volume rendering on an implicit SDF representation and thus has the best of both worlds, i.e. accurate surface representation by using an implicit SDF model and robust network training in the presence of abrupt depth changes as enabled by volume rendering. Note that simply applying the standard the volume rendering method to the density associated with SDF would lead to bias (i.e. systematic geometric errors) in the reconstructed surfaces, which is a subtle but important observation that we will elaborate later. Hence, we will present a novel volume rendering algorithm to ensure unbiased surface reconstruction. Experiments on both DTU dataset and BlendedMVS dataset demonstrated that NeuS is capable of reconstructing complex 3D objects with severe occlusions and delicate structures. It outperforms the state-of-the-art neural reconstruction methods, namely IDR [31] and UniSurf [22], in terms of reconstruction quality.

2 Related Works

Classical Multi-view Surface and Volumetric Reconstruction. Traditional multi-view 3D reconstruction methods can be roughly classified into two categories: point- and surface-based reconstruction [2, 6, 8, 27] and volumetric reconstruction [5, 3, 28]. Point- and surface-based reconstruction methods estimate the depth map of each pixel by exploiting inter-image photometric consistency [7] and then fuse the depth maps into a global dense point cloud [17, 33]. The surface reconstruction is usually done as a post processing with methods like screened Poisson surface Reconstruction [13]. The reconstruction quality heavily relies on the quality of correspondence matching, and the difficulties in matching correspondence for objects without rich textures often lead to severe artifacts and missing parts in the reconstruction results. On the other hand, volumetric reconstruction methods side-step the problem of explicit correspondence matching by estimating occupancy and color in a voxel grid from multi-view images and evaluating the color consistency of each voxel. Due to limited achievable voxel resolution, these methods cannot achieve high accuracy.

Neural Implicit Representation. Neural implicit representation has recently become a popular alternative to classical scene representations, e.g. point cloud, voxel grids, meshes, due to its high achievable spatial resolution. This representation has been applied and made great progress in shape representation [18, 19, 23, 4, 1, 9, 32, 24], novel view synthesis [29, 16, 12, 20, 15, 25, 26] and multi-view 3D reconstruction [31, 21, 14, 11].

Our work mainly focus on learning implicit neural representation encoding both geometry and appearance from 2D images via classical rendering techniques. Limited in this scope, the related works can be roughly categorized based on the rendering techniques used, i.e. surface rendering based methods and volume rendering based methods. Surface rendering based methods [21, 14, 31] assume that the color of ray only relies on the color of an intersection of the ray with the scene geometry which makes the gradient only be backpropagated to a local region near the intersection. Therefore, such methods struggle with reconstructing complex objects with severe self-occlusions, sudden depth changes and thin parts. Furthermore, it requires object masks as supervision. On the contrary, our method performs well for such challenging cases without the need of masks.

Volume rendering based methods, such as NeRF[20], render an image by accumulating colors of the sampled points along each ray. Since during training, the gradient can be back-propagated to every sample points, it can handle sudden depth changes and synthesize high-quality images. However, extracting high-fidelity surface from the learned implicit field is difficult. This is because the density-based scene representation lacks sufficient constraints on level sets of the scene geometry and introduces ambiguities in density. In contrast, our method combines the advantages of surface rendering based and volume rendering based methods by constraining the scene space as a signed distance field and applying volume rendering to render the scene field. UNISURF [22], a concurrent unpublished work, also learns an implicit surface via volume rendering. It improves the reconstruction quality by shrinking the sample region of volume rendering during the optimization. Our method differs from UNISURF in that UNISURF represents the surface by occupancy values and gradually reduces the sample regions at some predefined steps to make the occupancy value converge to the surface while our method represents the scene by a signed distance field (SDF) and thus can naturally extracts the surface as the zero-set of the SDF, yielding better reconstruction accuracy than UNISURF as will be seen later in the Experiment section.

3 Method

Given a set of posed images $\{\mathcal{I}_k\}$ of a 3D object, our goal is to reconstruct the surface $\mathcal{S}$ of the object. The surface is represented by the zero-set of a Signed Distance Function (SDF), which is implicitly encoded by a neural network. In order to learn the parameters of this network, we developed a novel variant of the volume rendering method to render images from the implicit SDF and minimize the difference between the rendered images and the input images to train the network of SDF. This volume rendering approach ensures the robustness of optimization for network training and allows our NeuS method to reconstruct objects of complex structures.

3.1 Rendering Procedure

Scene representation. With NeuS, the 3D scene of an object O to be reconstructed is represented by two functions: $f : \mathbb{R}^3 \to \mathbb{R}$ that maps a point $\mathbf{x} \in \mathbb{R}^3$ to its signed distance to O, and $c : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}^3$ that encodes the color associated with a point $\mathbf{x} \in \mathbb{R}^3$ and a viewing direction $\mathbf{v} \in \mathbb{S}^2$. Both functions are implicitly encoded by neural networks of Multi-layer Perceptron (MLP). The surface of the object O is represented by the zero-set of its SDF function, that is,

$$\mathcal{S} = \left\{\mathbf{x} \in \mathbb{R}^3 | f(\mathbf{x}) = 0\right\}. \tag{1}$$

In order to apply a volume rendering method to training the SDF network, we introduce a density field $\bar{\phi}(\mathbf{x})$ induced by the signed distance function f:

$$\bar{\phi}(\mathbf{x}) = \phi_s(f(\mathbf{x})), \tag{2}$$

where $\phi_s(x)$ in principle can be any unimodal, i.e. bell-shaped, density distribution function centered at 0, and the standard deviation of $\phi_s(x)$ is $1/s$. In this paper, we choose to use $\phi(x) = se^{-sx}/(1 + e^{-sx})^2$, commonly known as the *logistic density distribution*, which is the derivative of the well-known Sigmoid function $\Phi_s(x) = (1 + e^{-sx})^{-1}$, i.e. $\phi(x) = \Phi_s'(x)$. Note that $\bar{\phi}(\mathbf{x})$ is a "virtual"

density field devised to facilitate the training of the SDF network via a volume rendering scheme, thus it is different from the radiance field used in the context of volume rendering for translucent objects or participating media.

Intuitively, the main idea of NeuS is that, with the aid of the density field $\bar{\phi}(\mathbf{x})$, volume rendering can be used to train the SDF network with the supervision of the 2D input images. Upon successful completion of network optimization, $\bar{\phi}(\mathbf{x})$ should assume prominently high density values near the object's surface, and it is therefore expected that the zero-set of the network-encoded SDF yields an accurately reconstructed surface $\mathcal{S}$.

Rendering. To learn the parameters of MLPs of the SDF and the color field, we devise a volume rendering scheme to render images from the proposed SDF representation and compare the rendered images with the input images for network supervision. Given a pixel, we denote the ray emitted from this pixel as $\{\mathbf{p}(t) = \mathbf{o} + t\mathbf{v} | t \geq 0\}$ where $\mathbf{o}$ is the center of the camera and $\mathbf{v}$ is the unit direction vector of the ray. We accumulate the colors along the ray by

$$\hat{C} = \int_0^{+\infty} w(t) c(\mathbf{p}(t), \mathbf{v}) \mathrm{d}t, \tag{3}$$

where $\hat{C}$ is the output color for this pixel, $w(t)$ a weight for the point $\mathbf{p}(t)$ and $c(\mathbf{p}(t))$ is the color at the point $\mathbf{p}$ along the viewing direction $\mathbf{v}$. As a weight function, here $w(t)$ is required to satisfy that $w(t) \geq 0$ and $\int_0^{+\infty} w(t)\mathrm{d}t = 1$.

To evaluate this integral, we use a discrete approximation, which is similar to the composite trapezoid quadrature, by sampling n points $\{\mathbf{p}_i = \mathbf{o} + t_i\mathbf{v} | i = 1, ..., n, t_i < t_{i+1}\}$ along the ray and computing the approximate color

$$\hat{C} = \sum_{i=1}^{n} w_i c_i, \tag{4}$$

where $w_i = w(t_i)\delta_i$, $c_i = c(\mathbf{p}(t_i), \mathbf{v})$, and $\delta_i = t_{i+1} - t_i$.

Clearly, the key to computing an appropriate output color is to derive the weight function $w(t)$ on the ray based on the SDF function f of the scene. In the following, we first list the requirements of weight function w.

Requirements on weight function $w(t)$. A valid weight function needs to satisfy the following two properties.

1. **Unbiased**. $w(t)$ attains a locally maximal value at depth t only when $f(\mathbf{p}(t)) = 0$, that is, the point $\{\mathbf{p}(t)$ is on the surface defined by the zero-set of the SDF $(\mathbf{x})$.
2. **Occlusion-aware**. Given any two depth values t_0 and t_1 satisfying $f(t_0) = f(t_1)$ and $t_0 < t_1$, there is $w(t_0) > w(t_1)$. That is, when two points have the same SDF value (thus the same SDF-induced density value), the point near the view point should have a larger contribution to the final output color than does a point father away.

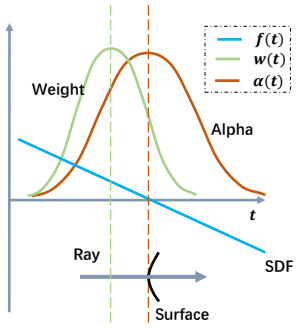


Figure 2: Weight bias of trivial solution II.

The unbiased property guarantees the reconstructed surface is exactly the zero-set of SDF because the points in the zero-set contributes most significantly to the output colors. Hence, a training procedure using volume rendering with a unbiased weight function $w(t)$ ensures that no bias is introduced in the reconstructed surface. The occlusion-aware property ensures that when a ray sequentially passes multiple surfaces, the rendering procedure will correctly use the color of the surface nearest to the camera to compute the output color.

Next, we will first analyze two naive ways of defining the weight function $w(t)$ before introducing our construction of $w(t)$. In fact, we will show that directly using the standard pipeline of volume rendering would produce undesirable bias in surface reconstruction (see Naive Solution 2 below).

Naive solution I. A straightforward way to construct a unbiased weight function is to directly use the density as weight

$$w_i = \frac{\bar{\phi}(\mathbf{p}(t_i))}{\sum_j \bar{\phi}(\mathbf{p}(t_j))}, \tag{5}$$

Obviously this weight function is unbiased but not occlusion-aware. If the ray passes two surfaces, the SDF function f will have two zero points on the ray, which results in two peaks on the weight function $w(t)$. In this case, the resulting weight functions will mix and average the color of two surfaces, producing inaccurate reconstruction of both surfaces.

Naive solution II. To make the weight function occlusion-aware, we may follow the standard volume rendering that sets the alpha (i.e opacity) to be equal to the density value $\bar{\phi}(f(\mathbf{p}(t_i))$ and compute the weight values by

$$\begin{cases} \alpha_i = \bar{\phi}(\mathbf{p}(t_i))\delta_i \\ w_i = \prod_k^{i-1}(1-\alpha_k)\alpha_i \end{cases} \tag{6}$$

where α is similar to the opacity concept in volume rendering, which is probability of surface existence on the depth range (t_i, t_{i+1}). $\prod_k^{i-1}(1-\alpha_k)$ denotes the probability that ray is not occluded by any depth before t_i. Obviously, the construction is occlusion-aware since Eq. 6 explicitly considers the occlusion relationship. However, such a construction is not unbiased, as will be proved in the supplementary materials. The intuition of this phenomenon is that α_i is obviously unbiased because $\alpha(t) = \bar{\phi}(f(\mathbf{p}(t)))$ but the production in Eq. 6 applied on the alpha values will introduce an offset to the final weights, as illustrated in Fig. 2.

Our solution. To construct an occlusion-aware and unbiased weight function, we first consider the case that there is only one surface. In this case, Eq. 5 is correct so we let the weight values satisfy Eq. 5 and then apply Eq. 6 to find the corresponding alpha value for this weight function. Then, we directly generalize these alpha values to compute weight values in all cases, including the case where there are multiple surfaces on the ray. With the aid of Eq. 6, the resulting weight values are actually able to handle the occlusion meanwhile they are also unbiased since they satisfy Eq. 5 around every surface. Specifically, our weight values are defined by

$$\begin{cases} \alpha(t_i) = \max\left(\frac{\Phi_s(-f(\mathbf{p}(t_{i+1})))-\Phi_s(-f(\mathbf{p}(t_i)))}{1-\Phi_s(-f(\mathbf{p}(t_i)))}, 0\right) \\ w_i = \prod_k^{i-1}(1-\alpha_k)\alpha_i \end{cases} \tag{7}$$

where $\Phi_s(x) = \int \phi(x)$ is the the Cumulative Density Function (CDF) of the logistic density distribution $\phi(x)$ defined in Eq. 2. Therefore, $\Phi_s(x) = (1+e^{-x})^{-1}$, the Sigmoid function.

Due to the utilization of alpha values, the proposed weight function is occlusion-aware. Meanwhile, we have the following theorem to ensure the unbiased property, which is also illustrated in Fig. 3.

Theorem 1 *If there exists a visible surface between two depth t_0 and t_1 of a ray, which means the ray is going from the outside of the surface into the inside of the surface with $f(\mathbf{p}(t_0)) > 0$, $f(\mathbf{p}(t_1)) < 0$ and $f(\mathbf{p}(t))$ is monotonically decreasing in the range (t_0, t_1), then the weight $w(t)$ computed by Eq. 7 will have a local maxima in this range at the depth t where $f(\mathbf{p}(t)) = 0$.*

The proof is left in the supplementary materials. We discuss the intuition behind the design by the example drawn in Fig. 3.

Intuition. Let us consider the case that there is a visible surface between the depth t_0 and the depth t_1. If we explicitly compute the w_i in Eq. 7, we can find that

$$w(t_i) = \frac{\Phi_s(-f(\mathbf{p}(t_{i+1}))) - \Phi_s(-f(\mathbf{p}(t_i))}{1-\Phi_s(-f(\mathbf{p}(t_0)))}, \tag{8}$$

where $\Phi_s(-f(\mathbf{p}(t_0)))$ can be regarded as a constant. In this case, if we regard the density function $1-\Phi_s$ as the proportion that the ray can go through the sample point, i.e., ***transmittance***, $\Phi_s(-f(\mathbf{p}(t_{i+1}))) - \Phi_s(-f(\mathbf{p}(t_i))$ is actually the probability of a surface that exists in the sampled section (t_i, t_{i+1}), i.e., ***attenuation***. So in our solution, we would like to set $\Phi_s(-f(\mathbf{p}(t_{i+1}))) - \Phi_s(-f(\mathbf{p}(t_i))$ as the weight value instead of the alpha value in this sampled section, which ensures that the weight value is unbiased.

3.2 Training

We optimize our neural networks and inverse standard deviation s by randomly sample a batch of pixels and their corresponding rays in world space $P = \{C_k, M_k, \mathbf{o}_k, \mathbf{v}_k\}$, where C_k is its color and M_k is its optional mask, from an image in every iteration. We assume the batch size is m.

Losses. To optimize the parameters, we define our objective function as:

$$\mathcal{L} = \mathcal{L}_{color} + \lambda \mathcal{L}_{reg} + \beta \mathcal{L}_{mask}. \tag{9}$$

The color loss $\mathcal{L}_{color}$ is defined as

$$\mathcal{L}_{color} = \frac{1}{m} \sum_{k} \mathcal{R}(\hat{C}_k, C_k). \tag{10}$$

Same as [31], we empirically choose $\mathcal{R}$ as L1 loss, which in our observation is robust to outliers and stable to train.

We add an Eikonal term [9] on the sampled points to regularize the network and enforce the outputs f_θ follow a signed distance field by

$$\mathcal{L}_{reg} = \frac{1}{nm} \sum_{k,i} (\nabla_f(\hat{\mathbf{p}}_{k,i}) - 1)^2. \tag{11}$$

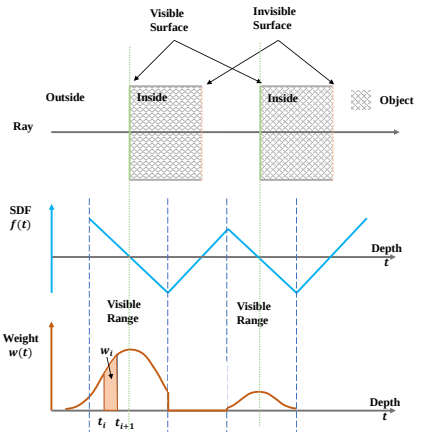


Figure 3: An example for the illustration of our weight function design.

The mask loss $\mathcal{L}_{mask}$ is an optional term, which is defined as

$$\mathcal{L}_{mask} = \text{BCE}(M_k, \hat{O}_k), \tag{12}$$

where $\hat{O}_k = \sum_{i=1}^{n} T_{k,i}\alpha_{k,i}$ is the sum of weights along the ray $\mathbf{o}_k + t\mathbf{v}_k$, and BCE is the binary cross entropy loss.

Hierarchical sampling. Like all other volume rendering techniques, the strategy of sampling will significantly influence the final results. In this work, we follow a similar hierarchical sampling strategy as NeRF [20]. We first uniformly sample the points on the ray and then conduct importance sampling on top of the coarse probability estimation. The difference is that, unlike NeRF which simultaneously optimizes a coarse network and a fine network, we only maintain one network, where probability in coarse sampling is computed by querying the network with a large fixed standard deviation $1/s$ while the probability of fine sampling is computed with the network with trainable s.

4 Experiments

4.1 Experimental settings.

Datasets. To evaluate our approach and baseline methods, we use 15 scenes from the DTU dataset [10], same as those used in IDR [31], with a wide variety of materials, appearance and geometry including challenging cases for reconstruction algorithms, such as non-Lambertian surfaces and thin structures. Each scene contains 49 or 64 images with the image resolution of 1600×1200. Each scene was tested with and without foreground masks provided by IDR [31]. We further tested on 7 challenging scenes from the low-res set of the BlendedMVS dataset [30](CC-4 License). Each scene has $31 - 143$ images at 768×576 pixels and masks are provided by the BlendedMVS dataset. We further captured two thin objects with 32 input images to test our approach on thin structure reconstruction.

Baselines. (1) The state-of-the-art surface rendering approach – IDR [31]: IDR can reconstruct surface with high quality but requires foreground masks as supervision for training; Since IDR has demonstrated superior quality compared to another surface rendering based method – DVR [21], we did not conduct a comparison with DVR. (2) The state-of-the-art volume rendering approach – NeRF [20]: NeRF achieves impressive results in novel view synthesis, however, extracting high-quality surface is not trivial. We use a density threshold of 25 to extract mesh from the learned implicit field. We validate this choice in the supplemental materials. (3) A widely-used classical MVS method – COLMAP [27]: We reconstruct a mesh from the output point cloud of COLMAP with screened Poisson Surface Reconstruction [13]. (4) The concurrent work which unifies surface rendering and volume rendering with an occupancy field as scene representation – UNISURF [22]. More details of the baseline methods are included in the supplemental material.

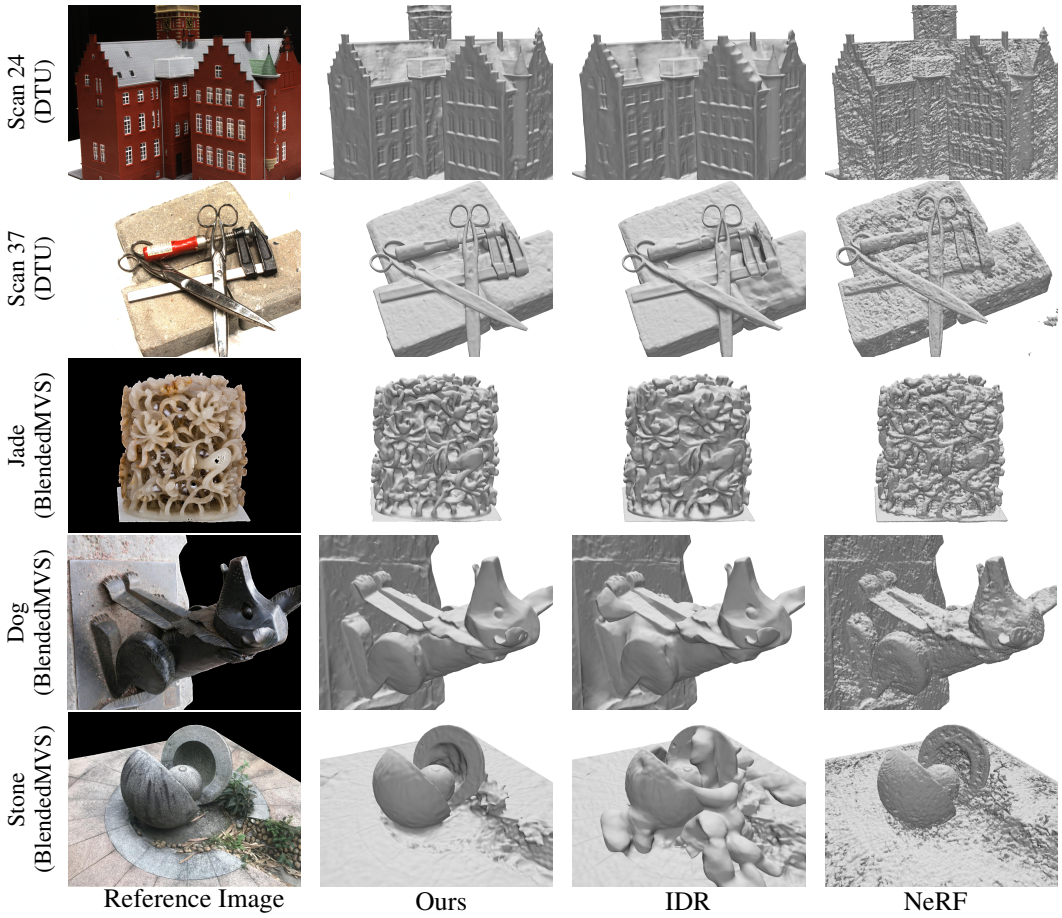


Figure 4: Comparions on surface reconstruction with mask supervision.

Implementation details. Similar to the network architecture of IDR [31], the signed distance function f is modeled by a MLP that consists of 8 hidden layers with hidden size of 256. The function c for color prediction is modeled by a MLP with 4 hidden layers with size of 256, which is conditioned on the spatial location x, normal n, and the feature vector from f. Positional encoding [20] is applied to spatial location x with 6 frequencies and to view direction v with 4 frequencies. We assume the region of interest is inside a unit sphere. The number of coarse and fine sampling is 64 and 64 respectively. For the 'w/o mask' setting, we sample additional 32 points outside the sphere, the outside scene is presented using NeRF++ [35]. Geometric initialization is used to produce an approximate SDF as proposed in [1]. We sample 512 rays per batch and train our model for 300k iterations for 14 hours (for the 'w/ mask' setting) and 16 hours (for the 'w/o mask' setting) on a single NVIDIA RTX2080Ti GPU.

	w/ mask			**w/o** mask			
ScanID	IDR	NeRF	Ours	COLMAP	NeRF	UNISURF	Ours
scan24	1.63	1.83	**1.15**	**0.81**	1.90	1.32	1.37
scan37	1.87	2.39	**0.95**	2.05	1.60	1.36	**1.21**
scan40	**0.63**	1.79	0.8	**0.73**	1.85	1.72	**0.73**
scan55	0.48	0.66	**0.39**	1.22	0.58	0.44	**0.4**
scan63	**1.04**	1.79	1.26	1.79	2.28	1.35	**1.2**
scan65	0.79	1.44	**0.72**	1.58	1.27	0.79	**0.7**
scan69	0.77	1.5	**0.69**	1.02	1.47	0.80	**0.72**
scan83	1.33	1.2	**0.94**	3.05	1.67	1.49	**1.01**
scan97	1.16	1.96	**1.14**	1.40	2.05	1.37	**1.16**
scan105	**0.76**	1.27	0.77	2.05	1.07	0.89	**0.82**
scan106	0.67	1.44	**0.66**	1.00	0.88	**0.59**	0.66
scan110	**0.9**	2.61	1.35	**1.32**	2.53	1.47	1.69
scan114	0.42	1.04	**0.39**	0.49	1.06	0.46	**0.39**
scan118	**0.51**	1.13	**0.51**	0.78	1.15	0.59	**0.49**
scan122	0.53	0.99	**0.52**	1.17	0.96	0.62	**0.51**
mean	0.9	1.54	**0.82**	1.36	1.49	1.02	**0.87**

Table 1: Quantitative evaluation on DTU dataset. COLMAP results are achieved by trim=0.

4.2 Comparisons

We conducted the comparisons in two settings, with mask supervision (w/ mask) and without mask supervision (w/o mask). We measure the reconstruction quality with the Chamfer distances in the

same way as UNISURF [22] and IDR [31] and report the scores in Table 1. The results show that our approach outperforms the baseline methods on the DTU dataset in both settings – w/ and w/o mask in terms of the Chamfer distance. Note that the reported scores of IDR in the setting of with mask and NeRF and UNISURF in the w/o mask setting are from IDR [31] and UniSURF [22].

We conduct the qualitative comparisons on the DTU dataset and the BlendedMVS dataset in both settings, w/ mask and w/o mask, in Figure 4 and Figure 6, respectively. As shown in Figure 4 for the setting of w/ mask, IDR shows limited performance for reconstructing thin metals parts in Scan 37 (DTU) and Jade (BlendedMVS), and fails to handle sudden depth changes in Stone (BlendedMVS) due to the local optimization process in surface rendering. The extracted meshes of NeRF's results are noisy since the volume density field has not sufficient constraint on level sets of 3D geometry. Regarding the w/o mask setting, we visually compare our method with NeRF and COLMAP in the setting of w/o mask in Figure 6, which show our reconstructed surfaces are with more fidelity than baselines. We further show a comparison with UNISURF [22] on two examples in the w/o mask setting. The UNISURF's results are provided by the authors of UNISURF. Our method works better for the objects with abrupt depth changes. More qualitative images and videos are included in the supplementary materials.

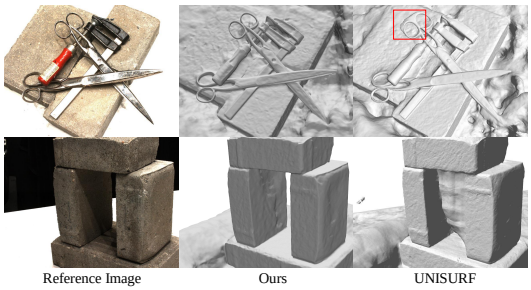


Figure 5: Visual comparisons with UNISURF.

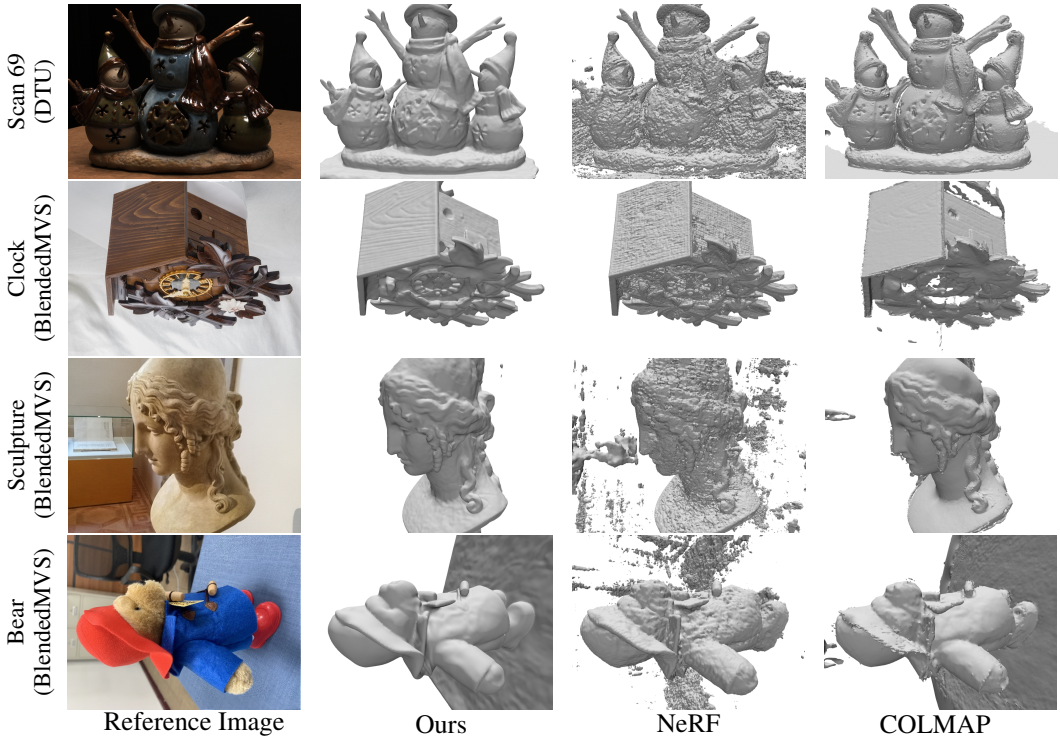


Figure 6: Comparions on surface reconstruction without mask supervision.

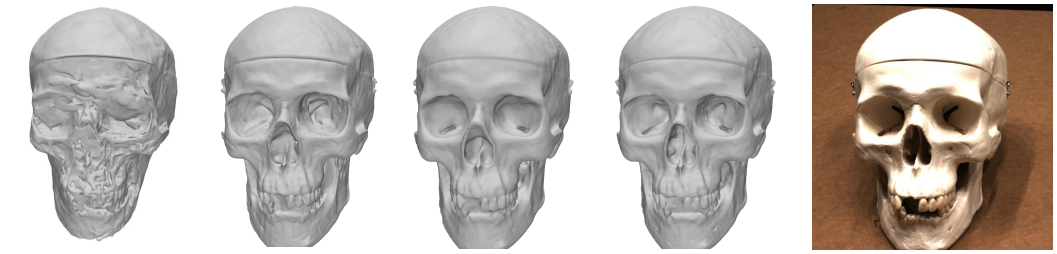
4.3 Analysis

Ablation study. To evaluate the effect of the weight calculation, We test three different kinds of weight constructions described in Sec. 3.1: (a) Naive solution I. (b) Naive solution II. (d) Full Model. As shown in Figure 7, if naive solution I is used, there are severe artifacts; Although reconstruction

geometry of naive sollution II looks plausible, the quantitative result is worse than our weight choice (d) in terms of the Chamfer distance. This is because it introduces a bias to the surface reconstruction.

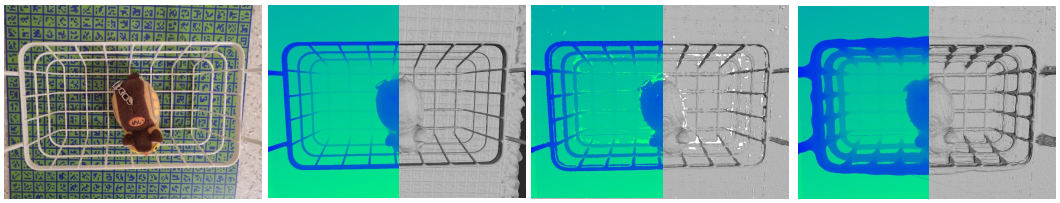
We also studied the effect of geometric initialization [1]. When the random initialization is used, artifacts appear at nose and eyes of the skull. More analysis can be found in supplementary materials.

Thin structures. We additionally show results on two challenging thin objects with 32 input images. As shown in Fig. 8, our method is able to accurately reconstruct these thin structures, especially on the edges with abrupt depth changes. Note that the plane with rich texture under the object is used for camera calibration.



(b) Naive Solution I	(b) Naive Solution II	(c) **w/o** Geo-Init.	(d) Full Model	Reference Image
3.22	1.68	0.87	0.72	Chamfer Distance

Figure 7: Ablation study.



Reference Image (a) Ours (b) COLMAP$_{trim\,=\,10}$ (c) COLMAP$_{trim\,=\,7}$

Figure 8: Comparison on scenes with thin structure objects. Left half is the depth map while right half is the reconstructed surface.

5 Conclusion

We have proposed *NeuS*, a new approach to multiview surface reconstruction that represents 3D surfaces as neural SDF and developed a new volume rendering method for training the implicit SDF representation. NeuS produces high-quality reconstruction and successfully reconstructs objects with severe occlusion and complex structures. It outperforms the state-of-the-arts both qualitatively and quantitatively. One limitation of our method is that although our method does not heavily rely on correspondence matching of texture features, the performance would still degrade for textureless objects (we show the failure cases in the supplemental materials). Moreover, NeuS has only a single standard deviation parameter s that is used to model the probability for all the spatial location. Hence, an interesting future research topic is to model the probability with different variances for different spatial locations together with the optimization of scene representation, depending on different local geometric characteristics. Negative societal impact: like many other learning-based works, our method requires a large amount of computational resources for network training, which can be a concern for global climate change.

References

[1] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2565–2574, 2020.

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009.

[3] Adrian Broadhurst, Tom W Drummond, and Roberto Cipolla. A probabilistic framework for space carving. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pages 388–393. IEEE, 2001.

[4] Z. Chen and H. Zhang. Learning implicit fields for generative shape modeling. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5932–5941, 2019.

[5] Jeremy S De Bonet and Paul Viola. Poxels: Probabilistic voxelized volume reconstruction. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 418–425, 1999.

[6] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009.

[7] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.

[8] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Gipuma: Massively parallel multi-view stereo reconstruction. *Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e. V*, 25(361-369):2, 2016.

[9] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020.

[10] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413, 2014.

[11] Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[12] Srinivas Kaza et al. *Differentiable volume rendering using signed distance functions*. PhD thesis, Massachusetts Institute of Technology, 2019.

[13] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3), July 2013.

[14] Petr Kellnhofer, Lars Jebe, Andrew Jones, Ryan Spicer, Kari Pulli, and Gordon Wetzstein. Neural lumigraph rendering. *arXiv preprint arXiv:2103.11571*, 2021.

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

[16] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Transactions on Graphics (TOG)*, 38(4):65, 2019.

[17] Paul Merrell, Amir Akbarzadeh, Liang Wang, Philippos Mordohai, Jan-Michael Frahm, Ruigang Yang, David Nistér, and Marc Pollefeys. Real-time visibility-based fusion of depth maps. pages 1–8, 01 2007.

[18] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.

[19] Mateusz Michalkiewicz, Jhony K. Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[20] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421. Springer, 2020.

[21] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020.

[22] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. *arXiv preprint arXiv:2104.10078*, 2021.

[23] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[24] Songyou Peng, Michael Niemeyer, Lars M. Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. *ArXiv*, abs/2003.04618, 2020.

[25] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. *ICCV*, 2019.

[26] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 84–93, 2020.

[27] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pages 501–518. Springer, 2016.

[28] Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173, 1999.

[29] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, pages 1119–1130, 2019.

[30] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1790–1799, 2020.

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

[32] Wang Yifan, Shihao Wu, Cengiz Oztireli, and Olga Sorkine-Hornung. Iso-points: Optimizing neural implicit surfaces with hybrid representations. *arXiv preprint arXiv:2012.06434*, 2020.

[33] Christopher Zach, Thomas Pock, and Horst Bischof. A globally optimal algorithm for robust tv-l1 range image integration. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.

[34] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. *arXiv preprint arXiv:2104.00674*, 2021.

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

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 5.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See supplementary material.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The code and the data are proprietary.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.1.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4.1.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cited the creators.
   (b) Did you mention the license of the assets? [Yes] See Section 4.1
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]