

---

# Neural Signed Distance Function Inference through Splatting 3D Gaussians Pulled on Zero-Level Set

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

It is vital to infer a signed distance function (SDF) for multi-view based surface reconstruction. 3D Gaussian splatting (3DGS) provides a novel perspective for volume rendering, and shows advantages in rendering efficiency and quality. Although 3DGS provides a promising neural rendering option, it is still hard to infer SDFs for surface reconstruction with 3DGS due to the discreteness, the sparseness, and the off-surface drift of 3D Gaussians. To resolve these issues, we propose a method that seamlessly merge 3DGS with the learning of neural SDFs. Our key idea is to more effectively constrain the SDF inference with the multi-view consistency. To this end, we dynamically align 3D Gaussians on the zero-level set of the neural SDF, and then render the aligned 3D Gaussians through the differentiable rasterization. Meanwhile, we update the neural SDF by pulling neighboring space to the pulled 3D Gaussians, which progressively refine the signed distance field near the surface. With both differentiable pulling and splatting, we jointly optimize 3D Gaussians and the neural SDF with both RGB and geometry constraints, which recovers more accurate, smooth, and complete surfaces with more geometry details. Our numerical and visual comparisons show our superiority over the state-of-the-art results on the widely used benchmarks.

## 1 Introduction

3D scene representations are important to various computer vision applications, such as multi-view 3D reconstruction [66, 16, 73, 35, 38, 70], novel view synthesis [46, 3, 4, 47], and neural SLAM [24, 58, 32, 75] etc.. Mesh and point clouds are the most common 3D scene representations, and can be rendered by fast rasterization on GPUs. Instead, more recent neural radiance fields (NeRFs) [46] are continuous scene representations, but it is slow to render NeRFs due to the need of costly stochastic sampling along rays in volume rendering. More recently, 3D Gaussians with different attributes like color and opacity are used as a versatile differentiable volumetric representation [30] for neural rendering through splatting, dubbed 3D Gaussian Splatting (3DGS). It prompts the pros of both NeRFs and point based representations, which achieves both better quality and faster speed in rendering. Although 3D Gaussians can render plausible images, it is still a challenge to reconstruct surfaces based on the 3D Gaussians.

The key challenge comes from the gap between the discrete 3D Gaussians and the continuous geometry representations, such as implicit functions. Besides the discreteness, the sparseness caused by uneven distribution and the off-surface drift make 3D Gaussians even harder to use than scanned point clouds in surface inference. To overcome these obstacles, recent solutions usually add previous volume rendering based reconstruction methods [60, 47] to 3DGS as a complement branch [40, 69, 10], use monocular depth and normal images as priors to bypass the messy and unordered 3D Gaussians [14, 56], or use surface-aligned Gaussians [25, 71, 14] in rasterization to

37 approximate surfaces. However, how to learn continuous implicit representations to recover more  
38 accurate, smooth, and complete surfaces with sharp geometry details is still an open question.

39 To answer this question, we introduce a novel method to infer neural SDFs from multi-view RGB  
40 images through 3D Gaussian splatting. We progressively infer a signed distance field by training a  
41 neural network along with learning 3D Gaussians to minimize rendering errors through splatting. To  
42 more effectively constrain the surface inference with the multi-view consistency, we dynamically  
43 align 3D Gaussians with the zero-level set of the neural SDF, and render the aligned 3D Gaussians  
44 on the zero-level set by differentiable rasterization. Meanwhile, we update the neural SDF by  
45 pulling the neighboring space onto the disk determined by each 3D Gaussian on the zero-level set,  
46 which gradually refines the signed distance field near the surface. The capability of seamlessly  
47 merging neural SDFs with 3DGs not only get rid of the dependence of costly NeRFs like stochastic  
48 sampling on rays but also enables us to access the field attributes like signed distances and gradients  
49 during the splatting process, which provides a novel perspective and a versatile platform for surface  
50 reconstruction with 3DGs. The key to the 3D Gaussian alignment and neural SDF inference is a  
51 differentiable pulling operation which uses the predicted signed distances and gradients from the  
52 neural SDF. It provides a way of imposing geometry based constraints on 3D Gaussians besides  
53 the RGB based constraints through splatting. Our numerical and visual evaluations on widely used  
54 benchmarks show our superiority over the latest methods in terms of reconstruction accuracy and  
55 recovered geometry details. Our contributions are listed below,

- 56 • We propose to infer neural SDF through splatting 3D Gaussians pulled on the zero-level  
57 set, which can more effectively constrain surface inference with the multi-view consistency.  
58 This enables to recover more accurate, smooth, and complete surfaces with geometry details.
- 59 • We introduce to dynamically align 3D Gaussians to the zero-level set and update the  
60 neural SDF through a differentiable pulling operation. To this end, we propose novel loss  
61 terms and training strategies to work with the discrete and sparse 3D Gaussians in surface  
62 reconstruction.
- 63 • We achieve the state-of-the-art numerical and visual results in multi-view based surface  
64 reconstruction.

## 65 2 Related Work

66 Neural implicit representations have achieved remarkable progress in reconstructing 3D geometry with  
67 details [49, 44, 6, 13, 41, 8, 23]. Neural implicit functions can be learned by either 3D supervisions,  
68 such as signed distances [49, 13, 53, 37] and binary occupancy labels [44], or 2D supervisions, such  
69 as multi-view RGB images [60] and normal images [65]. In the following, we focus on reviewing  
70 methods of learning implicit representations from 3D supervisions and 2D supervisions separately.  
71 Then we provide a detailed discussion on the latest reconstruction methods based on 3D Gaussians.

### 72 2.1 Learning Implicit Representation from Multi-view Images

73 Neural Radiance Fields (NeRFs) [46] have become an essential technology for representing 3D scene  
74 through multi-view images. Many of its applications have been explored, resulting in significant  
75 advancements in areas such as acceleration [47, 11], dynamic scene [18, 7] and sparse rendering [57,  
76 26]. Besides these applications, extracting accurate surfaces from NeRFs remains a challenge.  
77 Mainstream approaches typically design various differentiable formulas to transform the density in  
78 neural radiance fields into implicit representations for volume rendering, such as signed distance  
79 function (SDF) [60, 35, 50], unsigned distance function (UDF) [38, 43, 15] and occupancy [48].  
80 With the learned implicit function fields, post-processing algorithms [39, 23, 44] are applied to  
81 extract the level set to obtain the reconstructed meshes. Following methods introduce different priors  
82 from SfM [20, 72] or large-scale datasets [70, 59, 36] to improve the reconstruction performance in  
83 large-scale scenes. Recent approaches focus on speeding up the neural rendering procedure, aiming  
84 to achieve high-quality meshes and rendering views within a short period of training time. They  
85 propose alternative data structures to replace the heavy MLP framework used in vanilla NeRF, such  
86 as sparse voxel grid [19], multi-resolution hash grid [47, 61] and radial basis function [11], or design  
87 subtle differentiable rasterization pipelines to achieve real-time rendering [12, 67, 52]. However,  
88 these methods still face the trade-off between rendering quality and training speed.

89 **2.2 Learning Implicit Representation from Point Clouds**

90 Since DeepSDF [49] and OccNet [44] were proposed, learning implicit representation from point  
 91 clouds has achieved remarkable results in geometry modeling. These methods use ground truth signed  
 92 distances and binary occupancy labels calculated from ground truth point clouds as supervisions to  
 93 learn the implicit representation of shapes. The supervisions can serve as different kinds of global  
 94 priors [6, 38, 51, 54, 45] and local priors [17, 55, 62, 28], which enables the neural implicit function to  
 95 better capture geometry details and generalize to unseen shapes during inference. Some other methods  
 96 infer SDFs without 3D supervisions. They train neural networks to overfit on single point clouds.  
 97 These methods introduce additional constraints [1, 2, 21], novel ways of using gradients [41, 74],  
 98 specially designed priors [42, 33, 9] and normals [5, 34, 63] to estimate signed or unsigned distances  
 99 and occupancy, which use point clouds as a reference.

100 **2.3 Surface Reconstruction with 3D Gaussians**

101 3D Gaussian Splatting [30] has become a new paradigm in multi-view neural rendering due to its fast  
 102 training speed, intuitive explicit representation and outstanding rendering performance. Recent studies  
 103 attempt to recover accurate geometry from 3D Gaussians. One kind of approaches involves combining  
 104 3D Gaussians with neural implicit surface functions [60, 47] to enhance the performance of both  
 105 branches, which employs mutual supervisions between the two components [69, 10, 40]. Another  
 106 kind of approaches encourages the reduction from 3D Gaussians to 2D Gaussians with a series of  
 107 regularization terms, which ensures the Gaussian primitives to align with the object surfaces [25, 22,  
 108 14]. Additionally, some methods introduce additional priors from large-scale datasets [56, 14] or  
 109 multi-view stereo [64], or use elaborately designed surface extraction algorithms [71, 68] to recover  
 110 3D geometry from 3D Gaussians. Although these efforts have achieved improved reconstructions,  
 111 they are still limited in capturing fine-grained geometry and lack precise perception of surfaces  
 112 represented by the neural implicit function. Different from all these mentioned methods, we propose  
 113 to seamlessly combine 3DGs with the learning of neural SDFs. Our method will provide novel ways  
 114 to jointly learning 3DGs and neural SDFs by more effectively using multi-view consistency and  
 115 imposing geometry constraints.

116 **3 Method**

117 **Overview.** We aim to infer a neural SDF  $f$  from posed multi-view RGB images  $\{v_i\}_{i=1}^I$ , as shown  
 118 in Fig. 1. We learn 3D Gaussian functions  $\{g_j\}_{j=1}^J$  with their attributes like color, opacity, and shape  
 119 to represent the geometry and color in the 3D scene. Meanwhile, when learning the 3D Gaussians,  
 120 we introduce novel constraints to infer the continuous surfaces with the neural SDF. We rely on a  
 121 differentiable pulling operation and the differentiable rasterization to bridge the gap between the  
 122 discrete Gaussians and the continuous neural SDF, align 3D Gaussians on the zero-level set of the  
 123 neural SDF, and back propagate the supervision signals from both the rendering errors and other  
 124 geometry constraints to jointly optimize 3D Gaussians and the neural SDF.

125 **Neural Signed Distance Function.** We leverage an SDF  $f$  to represent the geometry of a scene.  
 126 An SDF  $f$  is an implicit function that can predict a signed distance  $s$  for an arbitrary location  $q$ ,  
 127 i.e.,  $s = f(q)$ . Recent methods usually train a neural network to approximate an SDF from signed  
 128 distance supervision or infer an SDF from 3D point clouds or multi-view images. A level set is an  
 129 iso-surface formed by the points with the same signed distance values. Zero-level set is a special  
 130 level set, which is formed by points with a signed distance of 0. We can use the marching cubes  
 131 algorithm [39] to extract the zero-level set a mesh surface. Another character of the zero-level set is  
 132 that the gradient of the SDF  $f$  at query  $q$  on the zero-level set, i.e.,  $\nabla f(q)$ , is the normal of  $q$ .

133 **3D Gaussian Splatting.** 3D Gaussians have become a vital differentiable volume representation for  
 134 scene modeling. We can learn a set of 3D Gaussians  $\{g_j\}_{j=1}^J$ , each of which has a set of learnable  
 135 attributes including mean, variances, rotation, opacity, and color. We can render the learnable  
 136 Gaussians  $\{g_j\}$  into RGB images through the volume rendering equation below,

$$C'(u, v) = \sum_{j=1}^J c_j * o_j * p_j(u, v) \prod_{k=1}^{j-1} (1 - o_k * p_k(u, v)), \quad (1)$$

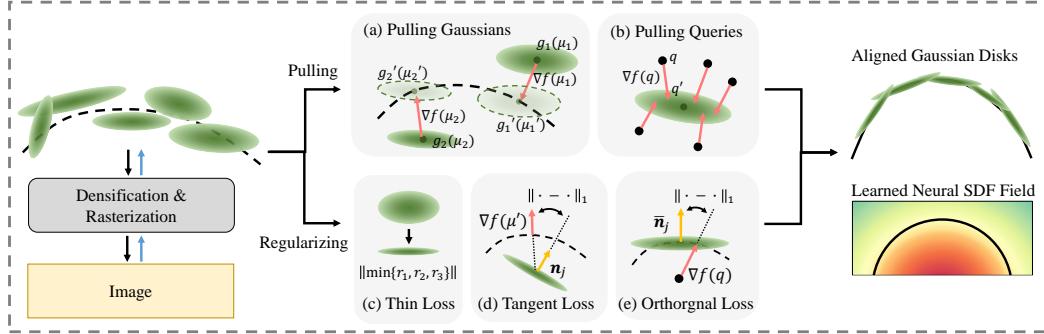


Figure 1: Overview of our method. We (a) pull 3D Gaussians onto the zero-level set for splatting, while (b) pulling the neighboring space onto the Gaussian disks for SDF inference. To better facilitate this procedure, we introduce three constraints: (c) push the Gaussians to become disks; (d) encourage the disk to be a tangent plane on the zero-level set; (e) constrain the query points to be pulled along the shortest path.

137 where  $\mathbf{C}'(u, v)$  is the rendered color at the pixel  $(u, v)$ ,  $c_i$ ,  $o_i$ , and  $p_i$  denote the color, the opacity,  
138 and the 2D projection of the  $j$ -th 3D Gaussian. At a query  $q = [x, y, z]$ , the probability from the  $j$ -th  
139 3D Gaussian is  $p_j(q) = \exp(-0.5 * (q - \mu_j)^T \Sigma^{-1} (q - \mu_j))$ , where  $\mu_j$  is the center of the  $j$ -th  
140 Gaussian, and  $\Sigma$  is the covariance matrix.

141 We can learn these 3D Gaussian functions through a differentiable rasterization. We render 3D  
142 Gaussians  $\{g_j\}$  into rendered RGB images  $v'_i$ . We can optimize the learnable attributes by minimizing  
143 the rendering errors to the ground truth observations  $v_i$ , where  $\mathbf{C}'(u, v)$  and  $\mathbf{C}(u, v)$  are the rendered  
144 and the GT color values at pixel  $(u, v)$ , i.e.,  $\min_{\{g_j\}} \|\mathbf{C}'(u, v) - \mathbf{C}(u, v)\|_2^2$ .

145 **Aligning 3D Gaussians with the Zero-level Set.** Since 3D Gaussians splatting is so flexible in  
146 volume rendering, it does not require 3D Gaussians to locate on the geometry surface for good  
147 rendering quality. While we expect 3D Gaussians to locate on geometry surface, so that we can more  
148 effectively leverage them and multi-view consistency as clues to infer more accurate neural SDFs  
149 for reconstruction. To this end, we introduce a differentiable pulling operation to pull 3D Gaussians  
150 on the zero-level set of the neural SDF  $f$ , and then, we render the pulled 3D Gaussians through the  
151 splatting.

152 Specifically, inspired by NeuralPull [41], we rely on the gradient field of the neural SDF  $f$  during the  
153 pulling operation. We move each one of the 3D Gaussians  $g_j$  using the predicted signed distance  
154  $s_j = f(\mu_j)$  and the gradient  $\nabla f(\mu_j)$ , where  $\mu_j$  is the mean value of the 3D Gaussian. As shown in  
155 Fig. 1 (a), this pulling operation will turn the 3D Gaussian  $g_j$  into a 3D Gaussian  $g'_j$  that get projected  
156 onto the zero-level set of SDF  $f$ , where  $g'_j$  shares the same attributes with  $g_j$  but has a different center  
157  $\mu'_j$ ,

$$\mu'_j = \mu_j - s_j * \nabla f(\mu_j) / \|\nabla f(\mu_j)\|. \quad (2)$$

158 **Signed Distance Inference with Pulled 3D Gaussians.** We infer signed distances in the field with  
159 pulled 3D Gaussians  $\{g'_j\}$ . Pulled 3D Gaussians provide a coarse estimation of the surface, which  
160 we can use as an reference. One challenge here is that the sparsity and non-uniformly distributed  
161 3D Gaussian do not show a clear geometry clue for surface inference. Although previous methods  
162 like NeuralTPS [9] and OnSurfPrior [42] manage to learn continuous implicit functions from sparse  
163 points, it is still difficult to recover surfaces from both sparse and non-uniformly distributed points.

164 To overcome this challenge, we introduce an approach to estimate neural SDFs from sparse 3D  
165 Gaussians. Like NeuralPull [41], we still use a differentiable pulling operation to pull neighboring  
166 space onto the surface but we regard the disk established by the shape of a 3D Gaussian as a pulling  
167 target, rather than a point, as shown in Fig. 2. To this end, we impose constraints not only on the  
168 shape of 3D Gaussians but also on the pulling operation. Specifically, we introduce three constraints.  
169 The first one constrains 3D Gaussians to be a thin disk. The second constraint encourages the thin  
170 disk to be a tangent plane on the zero-level set. The third constraint pushes queries to get pulled onto  
171 the thin disk along the normal.

172 The first constraint adds penalties if the smallest variance among the three variances of a 3D Gaussian  
 173  $g_j$  is too large, as shown in Fig. 1 (c). Thus, the loss for a thin disk Gaussian is listed below,

$$L_{Thin} = \|\min\{r_1, r_2, r_3\}\|_1, \quad (3)$$

174 where  $r_1, r_2$ , and  $r_3$  are variances along the three axes. Flattening a 3D Gaussian ellipsoid into a  
 175 disk has become a consensus in recent Gaussian reconstruction works [25, 10, 14]. The motivation is  
 176 that 2D planar disk primitives are more suitable for surface representation, making it easier to apply  
 177 alignment constraints. Additionally, we can naturally use the direction pointing along the axis with  
 178 the minimum variance  $\bar{r} = \min\{r_1, r_2, r_3\}$  to represent the normal  $n_j$  of the Gaussian  $g_j$ .

179 Based on the thin disk shape of Gaussians, the second constraint encourages the pulled Gaussians  
 180  $\{g'_j\}$  to be the tangent plane on the zero-level set, as shown in Fig. 1 (d). What we do is to align the  
 181 normal  $n_j$  of  $\{g_j\}$  with the true normal at the center  $\mu'_j$  of the pulled Gaussian on the zero-level set.  
 182 We use the gradient  $\nabla f(\mu'_j)$  of the neural SDF at  $\mu'_j$  as the expected normal here. Hence, we align  
 183 the normal  $n_j$  of a Gaussian with the normal  $\nabla f(\mu'_j)$  on the zero-level set,

$$L_{Tangent} = \|\nabla f(\mu'_j) - n_j\|_1 \quad (4)$$

184 With the disk-like Gaussians located on the  
 185 tangent plane, we introduce to sense the  
 186 signed distance field by pulling randomly  
 187 sampled queries on the Gaussian disks, as  
 188 shown in Fig. 1 (b). Turning the pulling  
 189 target from a point [41, 9] into a plane is  
 190 based on the observation that the 3D Gau-  
 191 ssian function with a boundary can cover the  
 192 surface although their centers  $\{\mu_1, \dots, \mu_j\}$   
 193 are sparse and non-uniformly distributed.  
 194 Thus, we expect the operation can pull a  
 195 query onto a Gaussian disk plane. Fig. 2  
 196 demonstrates the improvement of pulling  
 197 queries onto their nearest Gaussian disk  
 198 planes over the nearest Gaussian centers.  
 199 The comparisons show that pulling onto  
 200 the disk plane can improve the robustness  
 201 to the sparsity and non-uniformly Gaussian  
 202 distribution. With Gaussian centers, pulling  
 203 queries to centers can not recover the smooth and  
 204 continuous geometry in areas where almost no Gaussian centers appear. While pulling queries to the  
 205 Gaussian disk plane can recover more accurate and complete surfaces since the disk established by  
 the learned variance of Gaussian functions can mostly cover the gap.

206 Specifically, at a query  $q$ , we pull it onto the zero-level set using a similar way in Eq. 2, i.e.,  
 207  $q' = q - s * \nabla f(q) / |\nabla f(q)|$ . To encourage the query to get pulled onto the nearest pulled Gaussian  
 208 disk, we maximize its probability of its nearest pulled Gaussian  $\bar{g}$  which is determined in terms of the  
 209 distance between  $q$  and the Gaussian center  $\bar{\mu}$ ,

$$L_{Pull}(q'; \bar{\mu}) = -\log(e^{-1/2*(q' - \bar{\mu})^T \Sigma^{-1}(q' - \bar{\mu})}), \quad \bar{g} = \arg \min_{\{g'_j\}} \|\mu'_j - q\|_2^2. \quad (5)$$

210 Moreover, we expect the pulling can follow a direction orthogonal to the disk plane, which leads  
 211 to the minimum moving distance as defined by signed distances. To this end, we impose another  
 212 constraint on the gradient to ensure that the pulling can follow a path with the minimum distance to  
 213 the nearest pulled Gaussian disk, as shown in Fig. 1 (e),

$$L_{Orthogonal} = \|\nabla f(q) - \bar{n}_j\|_1, \quad (6)$$

214 where the constraint aligns the gradient at query  $q$  and the normal  $n_j$  of the pulled Gaussian disk  $\bar{\mu}$ .

215 **Rendering.** We also render the pulled Gaussians  $\{g'_j\}$  into images through splatting, and add penalties  
 216 on rendering errors,

$$L_{Splatting} = \|\mathbf{C}'(u, v) - \mathbf{C}(u, v)\|_2^2, \quad (7)$$

217 where  $\{g'_j\}$  are Gaussians pulled onto the zero-level set by the neural SDF  $f$  from the Gaussians  
 218  $\{g_j\}$  in Eq. 2. Each pair of  $g_j$  and  $g'_j$  share the same attributes expect the center location.

219 **Loss Function.** We optimize attributes of Gaussians  $\{g_j\}$  and the parameters of neural SDF  $f$  by the  
 220 following objective function, where  $\alpha, \beta, \gamma$ , and  $\delta$  are balance weights.

$$\min_{\{g_j\}, f} L_{Splatting} + \alpha L_{Thin} + \beta L_{Tangent} + \gamma L_{Pull} + \delta L_{Orthogonal}. \quad (8)$$

221 **Implementation Details.** Our code is build upon the source code released by 3DGS [30]. Similar  
 222 to [71], we make some changes to 3DGS’s densification strategy. The first one is to initialize the  
 223 newly cloned Gaussians around the original Gaussians rather than at the same positions. The second  
 224 one is to encourage 3DGS to split larger Gaussians into smaller ones more frequently. These strategies  
 225 aim to increase the number of primitives and to avoid underfitting in textureless areas. Regularization  
 226 parameters are set to  $\alpha=100, \beta=0.1, \gamma=1, \delta=0.1$ . We optimize our model for a total of 15k iterations.  
 227 We stop densification and incorporate the pulling and constraints at 7k iterations. The SDF network is  
 228 implemented as an MLP with 8 layers, 256 hidden units and ReLU activation function, and initialized  
 229 as a sphere, following [74]. The SDF network shares the same optimizer as that of 3D Gaussians,  
 230 while All experiments are conducted on a single NVIDIA 3090 GPU.

## 231 4 Experiments

### 232 4.1 Experiment Settings

233 **Evaluation Metrics and Datasets.** We evaluate the performance of our method on widely adopted  
 234 datasets including both object-level and large-scale ones, including DTU [27], Tanks and Temples  
 235 (TNT) [31] and Mip-NeRF 360 (M360) [4]. We use the raw image resolution of DTU and TNT, and  
 236 downscale the resolution of images in M360 by 4 for memory efficiency. To evaluate the accuracy of  
 237 the reconstructed meshes, we use Chamfer Distance (CD) for DTU and F-score for TNT, using the  
 238 official evaluation script. To evaluate the rendering quality in real-scene datasets, we report PSNR,  
 239 SSIM and LPIPS in evaluations on M360.

240 **Baselines.** We compare our geometry reconstruction accuracy with the state-of-the-art 3DGS  
 241 based reconstruction methods, including SuGaR [22], DN-Splatter [56], GaussianSurfels [14] and  
 242 2DGS [25]. For real-world scenes which do not have ground truth meshes for evaluations, we  
 243 compare the rendering quality with state-of-the-art neural rendering methods, including Instant-  
 244 NGP [47], Mip-NeRF 360 [4] and BakedSDF [67].

245 **Surface Extraction.** An advantage over the latest methods is the simplicity of extracting surfaces.  
 246 Different from methods like SuGaR [22] and GauS [68] which introduce specially designed algorithms  
 247 and take a long time for extracting surfaces, we adopt the marching cubes algorithm [39] to extract  
 248 mesh surfaces with the learned neural SDF  $f$ . For small scale scenes, we use a resolution of 800 to  
 249 extract surfaces, while we split large scale scenes into parts, each of which gets reconstructed with a  
 250 resolution of 800 to bypass the limitation of our computational resources.

### 251 4.2 Comparisons

252 **Comparisons on DTU.** We report accuracy of reconstructed meshes and training time against  
 253 baselines on DTU dataset in Tab. 1. Our method outperforms all Gaussian-based reconstruction  
 254 methods in terms of Chamfer Distance. Our method achieves comparable training time to the state-  
 255 of-the-art Gaussian-reconstruction method 2DGS [25] but gains better reconstruction accuracy than  
 256 2DGS. The visualization results in Fig. 3 highlight the advantages of our method. By employing  
 257 alignment constraints and pulling operations between the 3D Gaussians and the neural SDF field, we  
 258 can reconstruct significantly smoother and more complete surfaces than the baselines.

Table 1: Numerical comparisons in terms of CD on DTU dataset. Best results are highlighted as 1st, 2nd and 3rd.

Methods	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean	Time
NeuS [60]	1.00	1.37	0.93	0.43	1.10	0.65	0.57	1.48	1.09	0.83	0.52	1.20	0.35	0.49	0.54	0.84	~9h
3DGS [30]	2.14	1.53	2.08	1.68	3.49	2.21	1.43	2.07	2.22	1.75	1.79	2.55	1.53	1.52	1.50	1.96	15.1m
SuGaR [22]	1.47	1.33	1.13	0.61	2.25	1.71	1.15	1.63	1.62	1.07	0.79	2.45	0.98	0.88	0.79	1.33	1.6h
DN-Splatte [56]	1.60	2.03	1.42	1.44	2.37	2.11	1.62	1.95	1.88	1.48	1.63	1.82	1.20	1.50	1.40	1.70	31.2m
GSurfels [14]	0.66	0.93	0.54	0.41	1.06	1.14	0.85	1.29	1.53	0.79	0.82	1.58	0.45	0.66	0.53	0.88	10.9m
2DGS [25]	0.48	0.91	0.39	0.39	1.01	0.83	0.81	1.36	1.27	0.76	0.70	1.40	0.40	0.76	0.52	0.80	20.5m
Ours	0.51	0.56	0.46	0.39	0.82	0.67	0.85	1.37	1.25	0.73	0.54	1.39	0.35	0.88	0.42	0.74	21.8m

Table 2: Numerical comparisons on Tanks And Temples dataset. Best results are highlighted as 1st, 2nd and 3rd.

Methods	Barn	Caterpillar	Courthouse	Ignatius	Meetingroom	Truck	Mean	Time
NeuS [60]	0.29	0.29	0.17	0.83	0.24	0.45	0.38	~12h
3DGS [30]	0.13	0.08	0.09	0.04	0.01	0.19	0.09	20.5m
SuGaR [22]	0.14	0.16	0.08	0.33	0.15	0.26	0.19	2.1h
DN-Splatte [56]	0.15	0.11	0.07	0.18	0.01	0.20	0.12	54.9m
GSurfels [14]	0.24	0.22	0.07	0.39	0.12	0.24	0.21	15.1m
2DGS [25]	0.36	0.23	0.13	0.44	0.16	0.26	0.26	39.4m
Ours	0.60	0.37	0.16	0.71	0.19	0.52	0.43	37.6m

259 **Comparisons on TNT.** We further evaluate our method using more challenging large-scale unbounded  
260 scenes on TNT dataset. Numerical comparisons in Tab. 2 show that we achieve higher F-score  
261 compared to baseline methods, even surpassing NeuS, which however takes about 12 hours to fit a  
262 scene. Notably, as the scene scale increases, the number of Gaussian primitives increases rapidly,  
263 causing the adjusted CUDA rasterization kernel of 2DGS to consume more time for rendering. In  
264 contrast, since our rasterization kernel is based on 3DGS, it is less sensitive to the number of Gaussians,  
265 which enables us to obtain a shorter training time than 2DGS. We provide visual comparisons in  
266 Fig. 4. Here we crop the reconstructed meshes to show the foreground objects that are of primary  
267 interest, as captured by the cameras. Please refer to the appendix for the reconstruction results of the  
268 background regions. The visual comparisons demonstrate that we can reconstruct more complete and  
269 smooth object surfaces, such as the ground, the truck’s hood and the statue’s left shoulder.

270 **Comparisons on MipNeRF 360.** We further evaluate our method in neural  
271 rendering for novel view  
272 synthesis on MipNeRF 360  
273 dataset. We report the  
274 numerical comparisons in  
275 Tab. 3. Our competitive  
276 results against the state-of-  
277 the-art novel view synthe-  
278 sis methods, indicate that  
279 our method is able to im-  
280 pose effective geometric  
281 constraints without compro-  
282 mising rendering quality. This provides a promising solution for learning continuous distance fields  
283 from discrete 3D Gaussians. Visual comparisons of mesh reconstructions are shown in Fig. 5, which  
284 demonstrate that our method is able to recover more smooth and complete surface by more effectively  
285 using the multi-view consistency.

Table 3: Quantitative evaluations of rendering quality on Mip-NeRF 360 [4] dataset. Best results are highlighted as 1st, 2nd and 3rd.

	Indoor Scene			Outdoor Scene		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRF[46]	26.84	0.790	0.370	21.46	0.458	0.515
Instant-NGP [47]	29.15	0.880	0.216	22.90	0.566	0.371
MipNeRF 360 [4]	31.72	0.917	0.180	24.47	0.691	0.283
BakedSDF [67]	27.06	0.839	0.258	22.47	0.585	0.349
3DGS [30]	30.99	0.926	0.199	24.24	0.705	0.283
SuGaR [22]	29.44	0.911	0.216	22.76	0.631	0.349
2DGS [25]	30.39	0.924	0.183	24.33	0.709	0.284
Ours	30.78	0.925	0.182	23.76	0.703	0.278

### 288 4.3 Ablation Studies

289 In this section, we conduct ablation studies on the key techniques of our method to demonstrate their  
290 effectiveness. The full quantitative results are reported in Tab. 4, which are conducted on all scenes  
291 of TNT dataset [31].

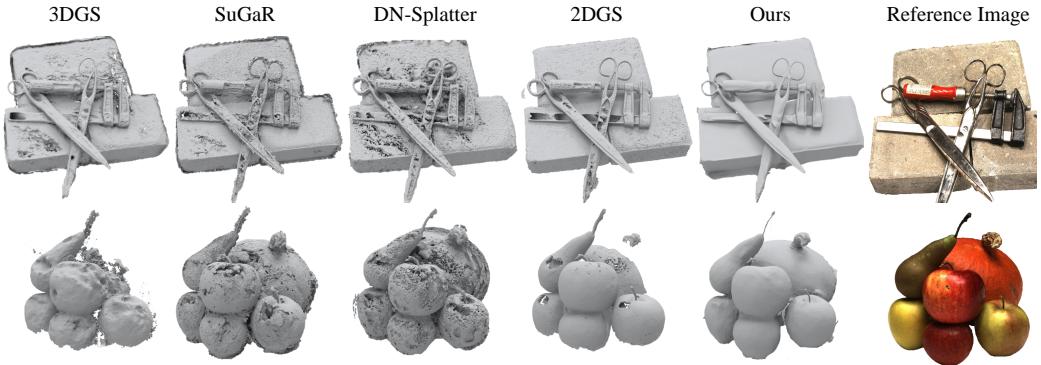


Figure 3: Visual comparisons on DTU dataset.

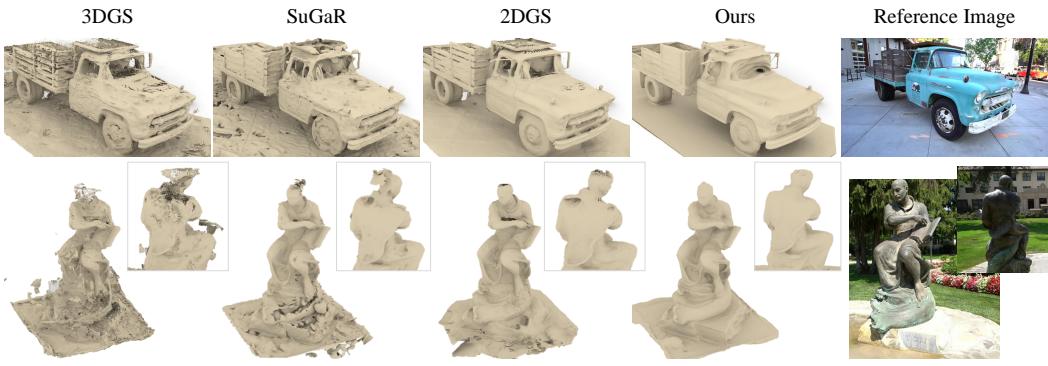


Figure 4: Visual comparisons on Tanks and Temples dataset.

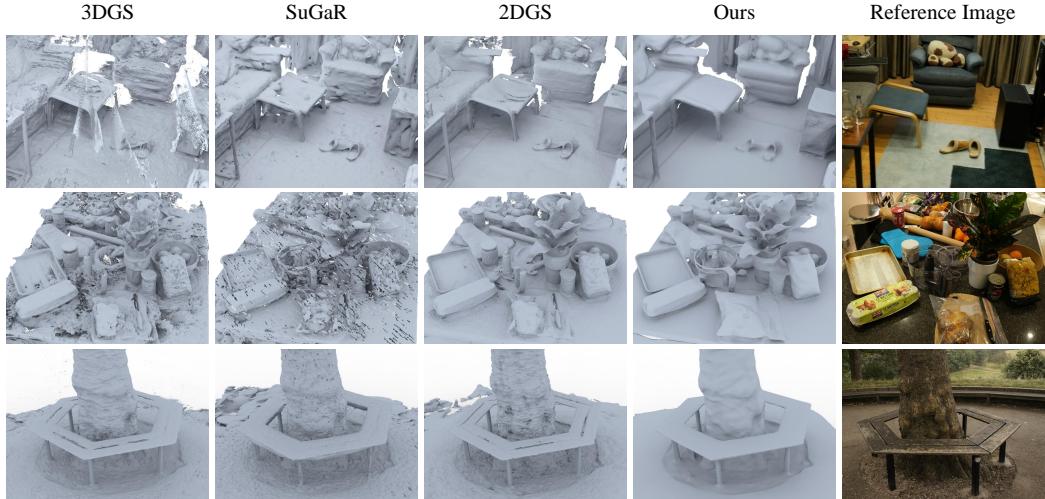


Figure 5: Visual comparisons on Mip-NeRF 360 dataset.

Table 4: Ablation studies on TNT dataset.

	Pulling		Constraint Terms			Mesh Extractions		
Methods	Pulled to centers	w/o Pull GS	w/o $L_{Thin}$	w/o $L_{Tan}$	w/o $L_{Oth}$	TSDF	Poisson	Full model
F-Score↑	0.41	0.33	0.39	0.37	0.43	0.29	0.42	0.46

292 **Pulling Operations.** We first examine  
 293 the effect of pulling Gaussians onto the  
 294 zero-level set, as reported in Tab. 4 ("w/o  
 295 Pull Gaussians" vs. "Ours"). The original  
 296 3DGS tends to produce floating ellip-  
 297 soids near the object surfaces to overfit the  
 298 training views. By pulling the Gaussians  
 299 to the zero-level set of the SDF field, the  
 300 Gaussians are consistently distributed on  
 301 the surface. As shown in Fig. 6, after get-  
 302 ting pulled onto the zero-level set, the Gau-  
 303 ssian centers are distributed on a thin layer  
 304 of the object surface, thus achieving an ac-  
 305 curate geometry estimation. Meanwhile,  
 306 we pull neighboring space onto Gaussian  
 307 disks to learn neural SDFs. Comparing to  
 308 NeuralPull [41] which pulls query points to  
 309 centers, we innovatively pull query points  
 310 to Gaussian disks, which bridge the gap  
 311 between continuous SDF field and sparse  
 312 Gaussian distributions, as highlighted in  
 313 Fig. 2 and Tab. 4 ("Pulled to centers" vs.  
 314 "Ours").

315 **Constraint Terms.** We further explore the  
 316 effect of our constraint terms, as reported  
 317 in Tab. 4 ("Constraint Terms"). Our full  
 318 model provides the best performance when  
 319 applying all constraint terms. The orthog-  
 320 onal loss helps to learn a more regularized  
 321 SDF field, while the thin loss and tangent  
 322 loss provide constraints to align the orienta-  
 323 tion of Gaussian disks with the gradient of  
 324 neural SDF on the zero-level sets, resulting  
 325 in a good normal field and a reconstructed  
 326 mesh, as shown in Fig. 8.

327 **Mesh Extraction.** We also report the re-  
 328 construction accuracy using TSDF fusion and screened Poisson recon-  
 329 struction [29], as shown in Fig. 9 and Tab. 4 ("Mesh Extractions"). For TSDF fusion,  
 330 we render depth maps and fuse them  
 331 using a voxel size of 0.004 and truncation threshold as 0.02, the same as 2DGS [25]. For screened  
 332 Poisson, we use the Gaussian centers and normals as input. Unlike 2DGS [25] and GSurfels [14]  
 333 which incorporate rendered depth into the differentiable rasterization pipeline, we do not directly  
 334 optimize depths, resulting in noisy depth maps and unsatisfactory reconstruction results. However,  
 335 since the positions and normals of the Gaussians are well optimized through our approach, screened  
 336 Poisson reconstruction can achieve relatively good results.

## 336 5 Conclusion

337 We propose a method to learn neural SDFs for multi-view surface reconstruction with 3D Gaussian  
 338 splatting. Our results show that we can more effectively leverage multi-view consistency to recover  
 339 more accurate, smooth, and complete surfaces with geometry details by rendering 3D Gaussians  
 340 pulled on the zero-level set. To this end, we dynamically align 3D Gaussians to the zero-level set and  
 341 update neural SDFs through both differentiable pulling and splatting for both RGB and geometry  
 342 constraints. Our methods successfully refine the signed distance field near the surface in a progressive  
 343 manner, leading to plausible surface reconstruction. Our ablation studies justify the effectiveness of  
 344 our novel modules, loss terms, and training strategies. Our evaluations show our superiority over the  
 345 latest methods in terms of accuracy, completeness, and smoothness.

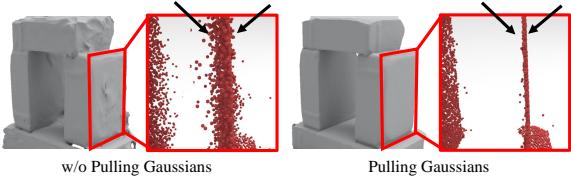


Figure 6: Visualization of Gaussian centers with or without pulled onto zero-level set. We are able to obtain consistent and smooth Gaussian distributions by pulling operation.

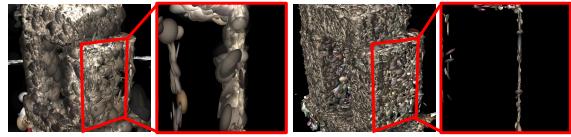


Figure 7: Comparisons between Gaussian ellipsoids learned by original 3DGS and Gaussian disks learned by our method.

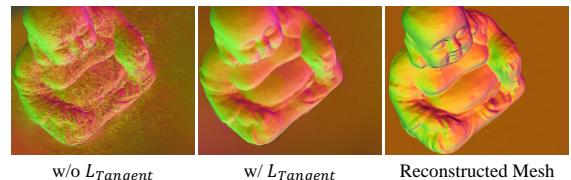


Figure 8: Qualitative ablation studies for Tangent loss.

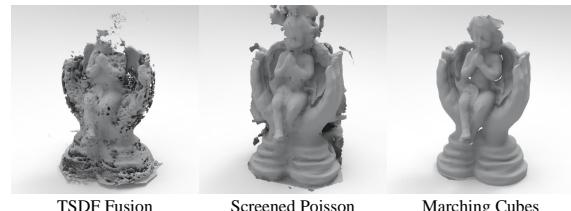


Figure 9: Comparisons of different mesh extraction methods.

346 **References**

- 347 [1] Matan Atzmon and Yaron Lipman. SAL: Sign agnostic learning of shapes from raw data. In  
348 *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.
- 349 [2] Matan Atzmon and yaron Lipman. SALD: sign agnostic learning with derivatives. In *International  
350 Conference on Learning Representations*, 2021.
- 351 [3] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla,  
352 and Pratul P. Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural  
353 radiance fields. *Proceedings of International Conference on Computer Vision (ICCV)*, 2021.
- 354 [4] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-  
355 NeRF 360: Unbounded anti-aliased neural radiance fields. *Proceedings of the IEEE Conference  
356 on Computer Vision and Pattern Recognition*, 2022.
- 357 [5] Alexandre Boulch, Pierre-Alain Langlois, Gilles Puy, and Renaud Marlet. Needrop: Self-  
358 supervised shape representation from sparse point clouds using needle dropping. In *International  
359 Conference on 3D Vision*, 2021.
- 360 [6] Alexandre Boulch and Renaud Marlet. Poco: Point convolution for surface reconstruction. In  
361 *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.
- 362 [7] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceed-  
363 ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141,  
364 2023.
- 365 [8] Chao Chen, Yu-Shen Liu, and Zhizhong Han. Gridpull: Towards scalability in learning implicit  
366 representations from 3d point clouds. In *Proceedings of the ieee/cvpr international conference  
367 on computer vision*, pages 18322–18334, 2023.
- 368 [9] Chao Chen, Yu-Shen Liu, and Zhizhong Han. Unsupervised inference of signed distance  
369 functions from single sparse point clouds without learning priors. In *Proceedings of the  
370 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17712–17723,  
371 2023.
- 372 [10] Hanlin Chen, Chen Li, and Gim Hee Lee. Neusg: Neural implicit surface reconstruction with  
373 3d gaussian splatting guidance. *arXiv preprint arXiv:2312.00846*, 2023.
- 374 [11] Zhang Chen, Zhong Li, Liangchen Song, Lele Chen, Jingyi Yu, Junsong Yuan, and Yi Xu.  
375 Neurbf: A neural fields representation with adaptive radial basis functions. In *Proceedings of  
376 the IEEE/CVF International Conference on Computer Vision*, pages 4182–4194, 2023.
- 377 [12] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. Mobilenerf:  
378 Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile  
379 architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern  
380 Recognition*, pages 16569–16578, 2023.
- 381 [13] Gene Chou, Ilya Chugunov, and Felix Heide. GenSDF: Two-stage learning of generalizable  
382 signed distance functions. In *Proc. of Neural Information Processing Systems (NeurIPS)*, 2022.
- 383 [14] Pinxuan Dai, Jiamin Xu, Wenxiang Xie, Xinguo Liu, Huamin Wang, and Weiwei Xu. High-  
384 quality surface reconstruction using gaussian surfels. In *SIGGRAPH 2024 Conference Papers*.  
385 Association for Computing Machinery, 2024.
- 386 [15] Junkai Deng, Fei Hou, Xuhui Chen, Wencheng Wang, and Ying He. 2s-udf: A novel two-stage  
387 udf learning method for robust non-watertight model reconstruction from multi-view images.  
388 2024.
- 389 [16] Yikang Ding, Wentao Yuan, Qingtian Zhu, Haotian Zhang, Xiangyue Liu, Yuanjiang Wang, and  
390 Xiao Liu. TransMVSNet: Global context-aware multi-view stereo network with transformers.  
391 In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*,  
392 pages 8585–8594, 2022.

- 393 [17] Philipp Erler, Paul Guerrero, Stefan Ohrhallinger, Niloy J. Mitra, and Michael Wimmer.  
 394 Points2Surf: Learning implicit surfaces from point clouds. In *European Conference on Com-*  
 395 *puter Vision*, 2020.
- 396 [18] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo  
 397 Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of*  
 398 *the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12479–12488,  
 399 2023.
- 400 [19] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo  
 401 Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF*  
 402 *Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022.
- 403 [20] Qiancheng Fu, Qingshan Xu, Yew-Soon Ong, and Wenbing Tao. Geo-Neus: Geometry-  
 404 consistent neural implicit surfaces learning for multi-view reconstruction. In *Advances in*  
 405 *Neural Information Processing Systems*, 2022.
- 406 [21] Amos Groppe, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric  
 407 regularization for learning shapes. *arXiv*, 2002.10099, 2020.
- 408 [22] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d  
 409 mesh reconstruction and high-quality mesh rendering. *Proceedings of the IEEE/CVF Conference*  
 410 *on Computer Vision and Pattern Recognition*, 2024.
- 411 [23] Benoit Guillard, Federico Stella, and Pascal Fua. Meshudf: Fast and differentiable meshing of  
 412 unsigned distance field networks. In *European Conference on Computer Vision*, pages 576–592.  
 413 Springer, 2022.
- 414 [24] Yasaman Haghighi, Suryansh Kumar, Jean-Philippe Thiran, and Luc Van Gool. Neural implicit  
 415 dense semantic slam. *arXiv preprint arXiv:2304.14560*, 2023.
- 416 [25] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian  
 417 splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*.  
 418 Association for Computing Machinery, 2024.
- 419 [26] Han Huang, Yulun Wu, Junsheng Zhou, Ge Gao, Ming Gu, and Yu-Shen Liu. Neusurf: On-  
 420 surface priors for neural surface reconstruction from sparse input views. In *Proceedings of the*  
 421 *AAAI Conference on Artificial Intelligence*, volume 38, pages 2312–2320, 2024.
- 422 [27] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale  
 423 multi-view stereopsis evaluation. In *IEEE Conference on Computer Vision and Pattern Recog-*  
 424 *nition*, pages 406–413, 2014.
- 425 [28] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas  
 426 Funkhouser, et al. Local implicit grid representations for 3D scenes. In *Proceedings of the*  
 427 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020.
- 428 [29] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transac-*  
 429 *tions on Graphics (ToG)*, 32(3):1–13, 2013.
- 430 [30] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian  
 431 splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14,  
 432 2023.
- 433 [31] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Bench-  
 434 marking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.
- 435 [32] Xin Kong, Shikun Liu, Marwan Taher, and Andrew J Davison. vmap: Vectorised object mapping  
 436 for neural field slam. *arXiv preprint arXiv:2302.01838*, 2023.
- 437 [33] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala,  
 438 and Timo Aila. Noise2noise: Learning image restoration without clean data. In *International*  
 439 *Conference on Machine Learning*, volume 80, pages 2971–2980, 2018.

- 440 [34] Shujuan Li, Junsheng Zhou, Baorui Ma, Yu-Shen Liu, and Zhizhong Han. Neaf: Learning  
 441 neural angle fields for point normal estimation. In *Proceedings of the AAAI Conference on*  
 442 *Artificial Intelligence*, volume 37, pages 1396–1404, 2023.
- 443 [35] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu,  
 444 and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *IEEE*  
 445 *Conference on Computer Vision and Pattern Recognition*, 2023.
- 446 [36] Zhihao Liang, Zhangjin Huang, Changxing Ding, and Kui Jia. Helixsurf: A robust and efficient  
 447 neural implicit surface learning of indoor scenes with iterative intertwined regularization. In  
 448 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages  
 449 13165–13174, 2023.
- 450 [37] David B Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. Bacon: Band-limited  
 451 coordinate networks for multiscale scene representation. In *Proceedings of the IEEE/CVF*  
 452 *conference on computer vision and pattern recognition*, pages 16252–16262, 2022.
- 453 [38] Xiaoxiao Long, Cheng Lin, Lingjie Liu, Yuan Liu, Peng Wang, Christian Theobalt, Taku  
 454 Komura, and Wenping Wang. Neuraludf: Learning unsigned distance fields for multi-view  
 455 reconstruction of surfaces with arbitrary topologies. In *Proceedings of the IEEE/CVF Conference*  
 456 *on Computer Vision and Pattern Recognition*, pages 20834–20843, 2023.
- 457 [39] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface  
 458 construction algorithm. *Computer Graphics*, 21(4):163–169, 1987.
- 459 [40] Xiaoyang Lyu, Yang-Tian Sun, Yi-Hua Huang, Xiuzhe Wu, Ziyi Yang, Yilun Chen, Jiangmiao  
 460 Pang, and Xiaojuan Qi. 3dgsr: Implicit surface reconstruction with 3d gaussian splatting. *arXiv*  
 461 *preprint arXiv:2404.00409*, 2024.
- 462 [41] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Neural-pull: Learning signed  
 463 distance function from point clouds by learning to pull space onto surface. In *International*  
 464 *Conference on Machine Learning*, pages 7246–7257. PMLR, 2021.
- 465 [42] Baorui Ma, Yu-Shen Liu, Matthias Zwicker, and Zhizhong Han. Reconstructing surfaces for  
 466 sparse point clouds with on-surface priors. In *IEEE Conference on Computer Vision and Pattern*  
 467 *Recognition*, 2022.
- 468 [43] Xiaoxu Meng, Weikai Chen, and Bo Yang. Neat: Learning neural implicit surfaces with  
 469 arbitrary topologies from multi-view images. In *Proceedings of the IEEE/CVF Conference on*  
 470 *Computer Vision and Pattern Recognition*, pages 248–258, 2023.
- 471 [44] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger.  
 472 Occupancy networks: Learning 3D reconstruction in function space. In *Proceedings of the*  
 473 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.
- 474 [45] Zhenxing Mi, Yiming Luo, and Wenbing Tao. SSRNet: Scalable 3D surface reconstruction  
 475 network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.
- 476 [46] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi,  
 477 and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In  
 478 *European Conference on Computer Vision*, 2020.
- 479 [47] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics  
 480 primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–  
 481 15, 2022.
- 482 [48] Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: Unifying neural implicit  
 483 surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF*  
 484 *International Conference on Computer Vision*, pages 5589–5599, 2021.
- 485 [49] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove.  
 486 DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceed-*  
 487 *ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174,  
 488 2019.

- 489 [50] Minyoung Park, Mirae Do, Yeon Jae Shin, Jaeseok Yoo, Jongkwang Hong, Joongrock Kim, and  
 490 Chul Lee. H2o-sdf: Two-phase learning for 3d indoor reconstruction using object surface fields.  
 491 In *The Twelfth International Conference on Learning Representations*, 2023.
- 492 [51] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger.  
 493 Convolutional occupancy networks. In *European Conference on Computer Vision*, pages  
 494 523–540. Springer, 2020.
- 495 [52] Christian Reiser, Rick Szeliski, Dor Verbin, Pratul Srinivasan, Ben Mildenhall, Andreas Geiger,  
 496 Jon Barron, and Peter Hedman. Merf: Memory-efficient radiance fields for real-time view  
 497 synthesis in unbounded scenes. *ACM Transactions on Graphics (TOG)*, 42(4):1–12, 2023.
- 498 [53] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek  
 499 Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of  
 500 detail: Real-time rendering with implicit 3D shapes. In *IEEE Conference on Computer Vision  
 501 and Pattern Recognition*, 2021.
- 502 [54] Jiapeng Tang, Jiabao Lei, Dan Xu, Feiying Ma, Kui Jia, and Lei Zhang. SA-ConvONet: Sign-  
 503 agnostic optimization of convolutional occupancy networks. In *Proceedings of the IEEE/CVF  
 504 International Conference on Computer Vision*, 2021.
- 505 [55] Edgar Treitschke, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Carsten Stoll, and Chris-  
 506 tian Theobalt. PatchNets: Patch-Based Generalizable Deep Implicit 3D Shape Representations.  
 507 In *European Conference on Computer Vision*, 2020.
- 508 [56] Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho  
 509 Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing. *arXiv  
 510 preprint arXiv:2403.17822*, 2024.
- 511 [57] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling  
 512 depth ranking for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International  
 513 Conference on Computer Vision*, pages 9065–9076, 2023.
- 514 [58] Hengyi Wang, Jingwen Wang, and Lourdes Agapito. Co-slam: Joint coordinate and sparse  
 515 parametric encodings for neural real-time slam, 2023.
- 516 [59] Jiepeng Wang, Peng Wang, Xiaoxiao Long, Christian Theobalt, Taku Komura, Lingjie Liu,  
 517 and Wenping Wang. NeuRIS: Neural reconstruction of indoor scenes using normal priors. In  
 518 *European Conference on Computer Vision*, 2022.
- 519 [60] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang.  
 520 NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction.  
 521 In *Advances in Neural Information Processing Systems*, 34, 2021.
- 522 [61] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie  
 523 Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *Pro-  
 524 ceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3295–3306,  
 525 2023.
- 526 [62] Francis Williams, Teseo Schneider, Claudio Silva, Denis Zorin, Joan Bruna, and Daniele  
 527 Panozzo. Deep geometric prior for surface reconstruction. In *IEEE Conference on Computer  
 528 Vision and Pattern Recognition*, 2019.
- 529 [63] Francis Williams, Matthew Trager, Joan Bruna, and Denis Zorin. Neural splines: Fitting 3D  
 530 surfaces with infinitely-wide neural networks. In *IEEE Conference on Computer Vision and  
 531 Pattern Recognition*, pages 9949–9958, 2021.
- 532 [64] Yaniv Wolf, Amit Bracha, and Ron Kimmel. Surface reconstruction from gaussian splatting via  
 533 novel stereo views. *arXiv preprint arXiv:2404.01810*, 2024.
- 534 [65] Cao Xu and Taketomi Takafumi. Supernormal: Neural surface reconstruction via multi-view  
 535 normal integration. In *CVPR*, 2024.

- 536 [66] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for  
537 unstructured multi-view stereo. *European Conference on Computer Vision*, 2018.
- 538 [67] Lior Yariv, Peter Hedman, Christian Reiser, Dor Verbin, Pratul P Srinivasan, Richard Szeliski,  
539 Jonathan T Barron, and Ben Mildenhall. Bakedsdf: Meshing neural sdf for real-time view  
540 synthesis. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–9, 2023.
- 541 [68] Chongjie Ye, Yinyu Nie, Jiahao Chang, Yuantao Chen, Yihao Zhi, and Xiaoguang Han.  
542 Gaustudio: A modular framework for 3d gaussian splatting and beyond. *arXiv preprint arXiv:2403.19632*, 2024.
- 544 [69] Mulin Yu, Tao Lu, Lining Xu, Lihan Jiang, Yuanbo Xiangli, and Bo Dai. Gsdf: 3dgs meets sdf  
545 for improved rendering and reconstruction. *arXiv preprint arXiv:2403.16964*, 2024.
- 546 [70] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. MonoSDF:  
547 Exploring monocular geometric cues for neural implicit surface reconstruction. *ArXiv*,  
548 abs/2202.00665, 2022.
- 549 [71] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient and compact  
550 surface reconstruction in unbounded scenes. *arXiv preprint arXiv:2404.10772*, 2024.
- 551 [72] Jingyang Zhang, Yao Yao, Shiwei Li, Tian Fang, David McKinnon, Yanghai Tsin, and Long  
552 Quan. Critical regularizations for neural surface reconstruction in the wild. In *Proceedings  
553 of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6270–6279,  
554 2022.
- 555 [73] Dongxu Zhao, Daniel Lichy, Pierre-Nicolas Perrin, Jan-Michael Frahm, and Soumyadip Sen-  
556 gupta. Mvpsnet: Fast generalizable multi-view photometric stereo. In *Proceedings of the  
557 IEEE/CVF International Conference on Computer Vision*, pages 12525–12536, 2023.
- 558 [74] Junsheng Zhou, Baorui Ma, Yu-Shen Liu, Yi Fang, and Zhizhong Han. Learning consistency-  
559 aware unsigned distance functions progressively from raw point clouds. In *Advances in Neural  
560 Information Processing Systems (NeurIPS)*, 2022.
- 561 [75] Zihan Zhu, Songyou Peng, Viktor Larsson, Zhaopeng Cui, Martin R Oswald, Andreas Geiger,  
562 and Marc Pollefeys. Nicer-slam: Neural implicit scene encoding for rgb slam. *arXiv preprint  
563 arXiv:2302.03594*, 2023.

564 **A Appendix**

565 **A.1 Background Reconstruction**



Figure 10: Visualization of reconstructed backgrounds.

566 Since our SDF field learns by fitting Gaussian ellipsoids, it can infer implicit surfaces at any location  
567 where Gaussians are distributed. Therefore, our method has the same capability to reconstruct  
568 backgrounds as methods like TSDF fusion, as shown in Fig. 10. Current works generally utilize  
569 screened Poisson or TSDF fusion to reconstruct meshes [25, 22] and tend to reconstruct large sky  
570 spheres in the background. Our method learns neural SDFs and utilize marching cubes to reconstruct  
571 mesh, which avoid such bad cases.

572 **A.2 Theoretical Analysis**

573 We provide a theoretical analysis here to demonstrate  
 574 the advantage of pulling queries onto disks compared to  
 575 pulling queries onto centers. We provide a visual com-  
 576 parison of the two strategies in Fig. 11, showcasing the  
 577 changes of the loss function and the loss gradients as the  
 578 query point approaches the Gaussian center. As the query  
 579 point gets closer to the Gaussian center, the loss function  
 580 of “pulled to centers” decays at a constant rate, and the  
 581 gradient of the loss remains constant. In contrast, for “pulled  
 582 to disks”, the loss function decreases quadratically, and  
 583 the gradient of the loss gradually diminishes. This means  
 584 that under the influence of the disk loss, as the query point  
 585 approximates the center, the received gradient becomes  
 586 smaller, reducing the driving force that pushes the query  
 587 point towards the center. In other words, the disk loss has  
 588 a higher “tolerance” for the query point not being pulled  
 589 to the center. This explains why we can learn a continu-  
 590 ous field from a sparse and non-uniformly distribution of  
 591 Gaussian ellipsoids using the disk loss, whereas the center loss would lead to the SDF field overfitting  
 592 to every Gaussian center.

### 593 A.3 Limitations & Future Works

594 While our method successfully recovers accu-  
 595 rate appearance and geometry reconstruction for  
 596 a wide range of objects and scenes, it also has  
 597 several limitations. Firstly, the neural SDF is  
 598 seamlessly integrated with Gaussian ellipsoids,  
 599 making it difficult to avoid the inherent draw-  
 600 backs of original 3D Gaussians, such as the lack  
 601 of transparent objects and areas with strong spec-  
 602 ular reflections. Secondly, although we address  
 603 the issue of learning a continuous SDF field  
 604 from sparse and non-uniformly distributed Gaus-  
 605 sian ellipsoids by pulling query points to disks,  
 606 our method shows limited performance in ex-  
 607 tremely sparse areas. In very distant regions of unbounded scenes or areas with colors similar to the  
 608 background color, where 3DGS reconstructs no ellipsoids or only a few ellipsoids, our method tends  
 609 to produce holes. Thirdly, due to the continuous and smooth characteristics of MLPs, our SDF tends  
 610 to capture the low-frequency features of objects, making it difficult to reconstruct high-frequency  
 611 details. A failure case is shown in Figure. 12, where although we can reconstruct the very smooth  
 612 tablecloth, the details of the lego are missing. There are two potential solutions for this issue in the  
 613 future: one is to enhance the representation capability of the SDF by integrated with latest implicit  
 614 representation learning methods, such as BACON [37] and GridPull [8]; the other one is to dig into  
 615 the capabilities of TSDF fusion and screened Poisson in reconstructing our SDF field, which have the  
 616 ability to reconstruct arbitrary resolution details.

### 617 A.4 More Results

618 We provide additional reconstruction results in Fig. 13, 14, 15, which further justifies the superiority  
 619 of our method.

### 620 A.5 Source Codes

621 We provide our demonstration code as our supplementary materials. We will release the source code,  
 622 data and instructions upon acceptance.

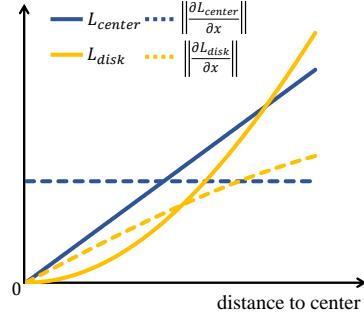


Figure 11: Visualization of loss and gradient between *Pulled to centers* and *Pulled to disks* with the distance of a query point to the Gaussian center.

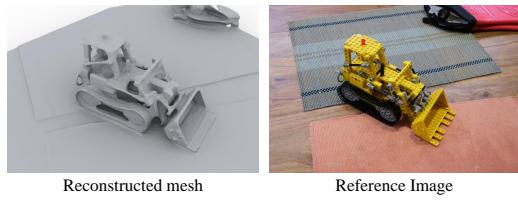


Figure 12: Failure case. This is because the SDF network cannot accurately capture high-frequency details due to the smooth characteristic of MLPs.

15



Figure 13: More visualization results on DTU dataset.



Figure 14: More visualization results on TNT dataset.



Figure 15: More visualization results on MipNeRF 360 dataset.

623 **NeurIPS Paper Checklist**

624 **1. Claims**

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

627 Answer: [Yes]

628 Justification: Our main claims made in the abstract and introduction accurately reflect the  
629 paper's contributions and scope.

630 Guidelines:

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

640 **2. Limitations**

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

642 Answer: [Yes]

643 Justification: We discuss the limitations and failure cases in the Appendix A.3.

644 Guidelines:

- 645 • The answer NA means that the paper has no limitation while the answer No means that  
646 the paper has limitations, but those are not discussed in the paper.
- 647 • The authors are encouraged to create a separate "Limitations" section in their paper.
- 648 • The paper should point out any strong assumptions and how robust the results are to  
649 violations of these assumptions (e.g., independence assumptions, noiseless settings,  
650 model well-specification, asymptotic approximations only holding locally). The authors  
651 should reflect on how these assumptions might be violated in practice and what the  
652 implications would be.
- 653 • The authors should reflect on the scope of the claims made, e.g., if the approach was  
654 only tested on a few datasets or with a few runs. In general, empirical results often  
655 depend on implicit assumptions, which should be articulated.
- 656 • The authors should reflect on the factors that influence the performance of the approach.  
657 For example, a facial recognition algorithm may perform poorly when image resolution  
658 is low or images are taken in low lighting. Or a speech-to-text system might not be  
659 used reliably to provide closed captions for online lectures because it fails to handle  
660 technical jargon.
- 661 • The authors should discuss the computational efficiency of the proposed algorithms  
662 and how they scale with dataset size.
- 663 • If applicable, the authors should discuss possible limitations of their approach to  
664 address problems of privacy and fairness.
- 665 • While the authors might fear that complete honesty about limitations might be used by  
666 reviewers as grounds for rejection, a worse outcome might be that reviewers discover  
667 limitations that aren't acknowledged in the paper. The authors should use their best  
668 judgment and recognize that individual actions in favor of transparency play an impor-  
669 tant role in developing norms that preserve the integrity of the community. Reviewers  
670 will be specifically instructed to not penalize honesty concerning limitations.

671 **3. Theory Assumptions and Proofs**

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

674 Answer: [NA]

675 Justification: We discussed why pulling queries onto disks is better than pulling queries onto  
676 centers in the appendix. There is no theoretical theorems or lemmas in it, but We tried our  
677 best to give a theoretical analysis for this discussion.

678 Guidelines:

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

#### 689 4. Experimental Result Reproducibility

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

693 Answer: [Yes]

694 Justification: We provide the detailed information in reproducing our methods in Section 3  
695 of the main paper.

696 Guidelines:

- 697 • The answer NA means that the paper does not include experiments.
- 698 • If the paper includes experiments, a No answer to this question will not be perceived  
699 well by the reviewers: Making the paper reproducible is important, regardless of  
700 whether the code and data are provided or not.
- 701 • If the contribution is a dataset and/or model, the authors should describe the steps taken  
702 to make their results reproducible or verifiable.
- 703 • Depending on the contribution, reproducibility can be accomplished in various ways.  
704 For example, if the contribution is a novel architecture, describing the architecture fully  
705 might suffice, or if the contribution is a specific model and empirical evaluation, it may  
706 be necessary to either make it possible for others to replicate the model with the same  
707 dataset, or provide access to the model. In general, releasing code and data is often  
708 one good way to accomplish this, but reproducibility can also be provided via detailed  
709 instructions for how to replicate the results, access to a hosted model (e.g., in the case  
710 of a large language model), releasing of a model checkpoint, or other means that are  
711 appropriate to the research performed.
- 712 • While NeurIPS does not require releasing code, the conference does require all submis-  
713 sions to provide some reasonable avenue for reproducibility, which may depend on the  
714 nature of the contribution. For example
  - 715 (a) If the contribution is primarily a new algorithm, the paper should make it clear how  
716 to reproduce that algorithm.
  - 717 (b) If the contribution is primarily a new model architecture, the paper should describe  
718 the architecture clearly and fully.
  - 719 (c) If the contribution is a new model (e.g., a large language model), then there should  
720 either be a way to access this model for reproducing the results or a way to reproduce  
721 the model (e.g., with an open-source dataset or instructions for how to construct  
722 the dataset).
  - 723 (d) We recognize that reproducibility may be tricky in some cases, in which case  
724 authors are welcome to describe the particular way they provide for reproducibility.  
725 In the case of closed-source models, it may be that access to the model is limited in  
726 some way (e.g., to registered users), but it should be possible for other researchers  
727 to have some path to reproducing or verifying the results.

#### 728 5. Open access to data and code

729 Question: Does the paper provide open access to the data and code, with sufficient instruc-  
730 tions to faithfully reproduce the main experimental results, as described in supplemental  
731 material?

732 Answer: [Yes]

733 Justification: We provide our demonstration code as a part of our supplementary materials.  
734 We will release the source code, data and instructions upon acceptance.

735 Guidelines:

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

## 755 6. Experimental Setting/Details

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

759 Answer: [Yes]

760 Justification: We provide the training and testing details in the experiment section (Section  
761 4).

762 Guidelines:

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

## 768 7. Experiment Statistical Significance

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

771 Answer: [No]

772 Justification: We report the average performance as the experimental results.

773 Guidelines:

- 774 • The answer NA means that the paper does not include experiments.
- 775 • The authors should answer “Yes” if the results are accompanied by error bars, confi-  
776 dence intervals, or statistical significance tests, at least for the experiments that support  
777 the main claims of the paper.
- 778 • The factors of variability that the error bars are capturing should be clearly stated (for  
779 example, train/test split, initialization, random drawing of some parameter, or overall  
780 run with given experimental conditions).

- 781           • The method for calculating the error bars should be explained (closed form formula,  
 782           call to a library function, bootstrap, etc.)  
 783           • The assumptions made should be given (e.g., Normally distributed errors).  
 784           • It should be clear whether the error bar is the standard deviation or the standard error  
 785           of the mean.  
 786           • It is OK to report 1-sigma error bars, but one should state it. The authors should  
 787           preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis  
 788           of Normality of errors is not verified.  
 789           • For asymmetric distributions, the authors should be careful not to show in tables or  
 790           figures symmetric error bars that would yield results that are out of range (e.g. negative  
 791           error rates).  
 792           • If error bars are reported in tables or plots, The authors should explain in the text how  
 793           they were calculated and reference the corresponding figures or tables in the text.

## 794           8. Experiments Compute Resources

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

798           Answer: [Yes]

799           Justification: We report our training time with baseline methods in the experiment part.

800           Guidelines:

- 801           • The answer NA means that the paper does not include experiments.
- 802           • The paper should indicate the type of compute workers CPU or GPU, internal cluster,  
 803           or cloud provider, including relevant memory and storage.
- 804           • The paper should provide the amount of compute required for each of the individual  
 805           experimental runs as well as estimate the total compute.
- 806           • The paper should disclose whether the full research project required more compute  
 807           than the experiments reported in the paper (e.g., preliminary or failed experiments that  
 808           didn't make it into the paper).

## 809           9. Code Of Ethics

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

812           Answer: [Yes]

813           Justification: The research conducted in the paper conform, in every respect, with the  
 814           NeurIPS Code of Ethics.

815           Guidelines:

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

## 821           10. Broader Impacts

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

824           Answer: [Yes]

825           Justification: We discuss the applications and potential impacts of our method in the  
 826           introduction.

827           Guidelines:

- 828           • The answer NA means that there is no societal impact of the work performed.
- 829           • If the authors answer NA or No, they should explain why their work has no societal  
 830           impact or why the paper does not address societal impact.

- 831           • Examples of negative societal impacts include potential malicious or unintended uses  
832           (e.g., disinformation, generating fake profiles, surveillance), fairness considerations  
833           (e.g., deployment of technologies that could make decisions that unfairly impact specific  
834           groups), privacy considerations, and security considerations.  
835           • The conference expects that many papers will be foundational research and not tied  
836           to particular applications, let alone deployments. However, if there is a direct path to  
837           any negative applications, the authors should point it out. For example, it is legitimate  
838           to point out that an improvement in the quality of generative models could be used to  
839           generate deepfakes for disinformation. On the other hand, it is not needed to point out  
840           that a generic algorithm for optimizing neural networks could enable people to train  
841           models that generate Deepfakes faster.  
842           • The authors should consider possible harms that could arise when the technology is  
843           being used as intended and functioning correctly, harms that could arise when the  
844           technology is being used as intended but gives incorrect results, and harms following  
845           from (intentional or unintentional) misuse of the technology.  
846           • If there are negative societal impacts, the authors could also discuss possible mitigation  
847           strategies (e.g., gated release of models, providing defenses in addition to attacks,  
848           mechanisms for monitoring misuse, mechanisms to monitor how a system learns from  
849           feedback over time, improving the efficiency and accessibility of ML).

850           **11. Safeguards**

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

854           Answer: [NA]

855           Justification: The paper poses no such risks.

856           Guidelines:

- 857           • The answer NA means that the paper poses no such risks.  
858           • Released models that have a high risk for misuse or dual-use should be released with  
859           necessary safeguards to allow for controlled use of the model, for example by requiring  
860           that users adhere to usage guidelines or restrictions to access the model or implementing  
861           safety filters.  
862           • Datasets that have been scraped from the Internet could pose safety risks. The authors  
863           should describe how they avoided releasing unsafe images.  
864           • We recognize that providing effective safeguards is challenging, and many papers do  
865           not require this, but we encourage authors to take this into account and make a best  
866           faith effort.

867           **12. Licenses for existing assets**

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

871           Answer: [Yes]

872           Justification: We use the open-sourced datasets and codes under their licenses.

873           Guidelines:

- 874           • The answer NA means that the paper does not use existing assets.  
875           • The authors should cite the original paper that produced the code package or dataset.  
876           • The authors should state which version of the asset is used and, if possible, include a  
877           URL.  
878           • The name of the license (e.g., CC-BY 4.0) should be included for each asset.  
879           • For scraped data from a particular source (e.g., website), the copyright and terms of  
880           service of that source should be provided.  
881           • If assets are released, the license, copyright information, and terms of use in the  
882           package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets)  
883           has curated licenses for some datasets. Their licensing guide can help determine the  
884           license of a dataset.

- 885           • For existing datasets that are re-packaged, both the original license and the license of  
886           the derived asset (if it has changed) should be provided.  
887           • If this information is not available online, the authors are encouraged to reach out to  
888           the asset's creators.

889           **13. New Assets**

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

892           Answer: [NA]

893           Justification: The paper does not release new assets.

894           Guidelines:

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

903           **14. Crowdsourcing and Research with Human Subjects**

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

907           Answer: [NA]

908           Justification: The paper does not involve crowdsourcing nor research with human subjects.

909           Guidelines:

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

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

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

924           Answer: [NA]

925           Justification: The paper does not involve crowdsourcing nor research with human subjects.

926           Guidelines:

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