Generative Occupancy Fields for 3D Surface-Aware Image Synthesis

Anonymous Author(s)
Affiliation
Address
email

Abstract

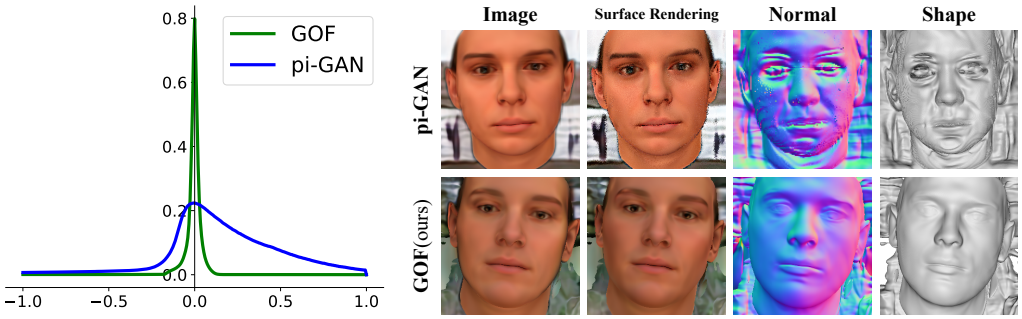
The advent of generative radiance fields has significantly promoted the development of 3D-aware image synthesis. The cumulative rendering process in radiance fields makes training these generative models much easier since gradients are distributed over the entire volume, but leads to diffused object surfaces. In the meantime, compared to radiance fields occupancy representations could inherently ensure deterministic surfaces. However, if we directly apply occupancy representations to generative models, during training they will only receive sparse gradients located on object surfaces and eventually suffer from the convergence problem. In this paper, we propose **Generative Occupancy Fields (GOF)**, a novel model based on generative radiance fields that can learn compact object surfaces without impeding its training convergence. The key insight of GOF is a dedicated transition from the cumulative rendering in radiance fields to rendering with only the surface points as the learned surface gets more and more accurate. In this way, GOF combines the merits of two representations in a unified framework. In practice, the training-time transition of *start from radiance fields and march to occupancy representations* is achieved in GOF by gradually shrinking the sampling region in its rendering process from the entire volume to a minimal neighboring region around the surface. Through comprehensive experiments on multiple datasets, we demonstrate that GON can synthesize high-quality images with 3D consistency and simultaneously learn compact and smooth object surfaces.

1 Introduction

Deep generative adversarial networks [1, 2, 3, 4] have demonstrated their superiority in synthesizing photorealistic and striking images. However, these models are often constrained in the 2D domain, struggling to generate 3D consistent images, let alone grasping the underlying 3D object shapes. 3D-aware image synthesis thus becomes an appealing and promising choice as it learns a 3D representation explicitly from a collection of unposed images. Consequently, it can not only synthesize 3D consistent images by manually controlling the rendering camera poses, but also pave the way for various downstream tasks such as shape editing and relighting.

Inspired by the success of neural radiance fields (NeRF) [5] in 3D scene modeling, recent 3D-aware generative models, referred to as generative radiance fields (GRAFs), have applied NeRF as the explicit 3D representation for image synthesis [6, 7]. With the help of NeRF, they are capable of hallucinating photorealistic images in a 3D consistent manner. Moreover, since NeRF holds the superior ability for rendering translucent objects by compositing colored densities along each ray in its volume rendering process, it also significantly facilitates the training of GRAFs as gradients are naturally distributed over the entire volume. However, they still incur an inevitable incapacity of capturing an accurate and compact object surface. As shown in Fig. 1(a), the state-of-the-art GRAF

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



(a) Cumulative Rendering Weights

(b) Performance Comparison

Figure 1: **(a)** The cumulative rendering weights (color weights) of our approach GOF more focus on the surface (y-axis) than previous methods like pi-GAN [6], which indicates our predicted volume densities more concentrate on the object surfaces. **(b)** Owing to the diffused volume densities, the preceding method pi-GAN captures messy surface normals and object shapes. Moreover, the image rendered only with the surface points is quite noisy. In contrast, more surface-centralized densities predicted by our method ensure compact and smooth object surfaces thus enable a high-quality surface rendering during inference. (**Zoom in for best view**)

model pi-GAN is prone to predict diffused object surfaces, as the volume densities are smoothly spread around the surfaces. Such diffused surfaces could significantly hamper the applications of GRAF models in downstream tasks such as shape recovery. Moreover, under different light conditions, the artifacts of surfaces could be amplified and inherited through the rendering process, resulting in synthesized images that are messy and faulty.

In this work, we propose **Generative Occupancy Fields (GOF)**, a novel GRAF-like image synthesis model that can learn compact object surfaces. GOF is inspired by the design of occupancy networks [8] that implicitly represents a 3D surface with the continuous decision boundary of a neural classifier. In this way, occupancy networks are capable of effectively locating surfaces via root-finding and encouraging the compactness of modeled surfaces inherently. However, GOF avoids directly applying such a design to 3D-aware image synthesis. While occupancy networks require precise object masks to train [9, 10], a more crucial factor is that they rely on the surface points for differentiable rendering [11, 12, 9, 13]. A generative model equipped with occupancy representations will thus meet severe convergence problems during training due to the sparsity of gradients. To unify the merits of both NeRF and occupancy networks for 3D-aware image synthesis, GOF adopts the design of GRAFs and at the same time leverages a nontrivial transition from the cumulative rendering to rendering with only the surface points, *i.e. start from radiance fields and march to occupancy representations*. Specifically, GOF will reinterpret the alpha values in the cumulative rendering process as occupancy values, so that it can locate the learned surface via root-finding. Subsequently, it can naturally encourage the compactness of learned surfaces by gradually shrinking the sampling region in the rendering process from the entire volume to a minimal neighboring region around the surface.

Thanks to the unified integration of radiance fields and occupancy representations, GOF can benefit from the representation effectiveness of radiance fields while ensuring the compactness of learned object surfaces through the shrinking process. As presented in Fig. 1(a), in GOF the distribution of cumulative rendering weights concentrates more closely around object surfaces compared to that of pi-GAN, eventually resulting in a compact and smooth surface. Moreover, GOF can thus alternatively render an image only with points on the learned surfaces like occupancy networks as illustrated in Fig. 1 (b). And during inference such a rendering scheme has the potential to alleviate the burden of sampling a large number of points along each ray for synthesizing a single image. Through exhaustive experiments on synthetic and real-world datasets, we demonstrate that GOF can achieve state-of-the-art performance on 3D-aware image synthesis. Meanwhile, it is capable of capturing compact and accurate 3D shapes that empower its applications in various downstream tasks such as 3D shape reconstruction. We validate this point by quantitative results of 3D shape reconstruction on the Synface dataset. Finally, we have also verified the ability of GOF in rendering high-quality images with only the surface points, which is hardly achievable in previous approaches.

2 Related Work

Neural implicit function for 3D representations. A plethora of works [14, 8, 15, 16, 17, 18, 19, 20] has exploited neural implicit functions for 3D geometry modeling. Among these works, neural radiance fields (NeRF) [5] has attracted growing attention due to its compelling results on novel view synthesis. It leverages an MLP network to approximate the radiance fields of static 3D scenes. And by learning to reconstruct existing views, it is capable of capturing 3D geometric details from only 2D supervision. A series of succeeding variants of NeRF have been proposed to improve it, including utilizing the spatial sparsity to reduce its computational complexity [21, 22], refining the rendering process to improve its efficiency [23, 24], as well as adopting reflectance decompositions to enhance its modeling capacity [25, 26]. There are also works that capitalize on the differentiable rendering of neural implicit functions for 3D reconstruction [11, 12, 9, 13, 27]. Specifically, SDFDiff [12] relies on the interpolation of eight neighboring SDF samples around the surface intersection to obtain the derivatives, while Atzmon *et al* [27] use a sample network to relate samples' positions to network parameters and thus achieve an improved generalization ability. More interestingly, by adopting occupancy representations, DVR [11] and IDR [9] show volumetric rendering is inherently differentiable so that network parameters can be optimized directly with derived analytic gradients. Different from methods aforementioned above, GOF is a generative model for 3D-aware image synthesis that can learn 3D representations from a set of 2D images with unknown camera poses.

Generative 3D-aware image synthesis. In order to synthesize 3D consistent images, researchers have explored a lot on how to incorporate 3D representations into the classical GAN model [1]. Some methods [28, 29, 30] resort to learning from 3D data directly, yet the requirement of 3D supervision limits their practical applicability. A more appealing alternative is thus learn from unposed 2D images in an unsupervised manner. Preceding works along this line of research adopt voxels as their intermediate 3D representations [31, 32, 33] and achieve explicit control over the pose of synthesized images. Inspired by the superior representation capacity of radiance fields over voxels, recent attempts [6, 7, 34] have replaced voxels with neural radiance fields [5] to improve the fidelity of synthesized 3D consistent images. Despite the striking performance, these models, referred to as generative radiance fields (GRAFs), tend to predict diffused object surfaces, which impedes its applicability in various downstream tasks. In this work, GOF aims at resolving this problem of GRAFs by combining them with the perspective of occupancy networks [8] and recent successes of recovering smooth and accurate shapes from natural images [35, 36, 37, 38]. A concurrent work Unisurf [10] shares similar spirits with our proposed GOF but is different in tasks and focuses. Specifically, Unisurf focuses on multi-view 3D reconstruction and attempts to alleviate the requirement of training-time precise masks through the integration of radiance fields and occupancy representations. Nevertheless, it still requires images with ground-truth poses for training. In contrast to Unisurf, GOF targets on the challenging task of 3D-aware image synthesis, where the synthesized images should be not only natural and vivid, but also consistent in the 3D space. By integrating radiance fields and occupancy representations, GOF is able to facilitate the convergence of GRAFs and ensure the compactness of learned object surfaces. Compared to existing GRAFs, the applicability of GOF is thus significantly broadened.

3 Methodology

We propose **generative occupancy fields (GOF)**, a novel synthesis model, belonging to generative radiance fields and aiming to learn from unposed images. Conditioned on a latent code $\mathbf{z} \sim p_{\mathbf{z}}$, our generator g_θ can generate a 3D radiance field $\mathbf{R}$, from which we can render a realistic image with a sampled camera pose $\xi \sim p_\xi$ and simultaneously recover smooth and compact object surfaces. In the following, we first present the background of neural radiance fields, and then introduce our proposed GOF model in detail.

3.1 Neural Radiance Fields

We adopt neural radiance fields (NeRF) as our explicit 3D representation for image synthesis, owing to its strong performance in novel view synthesis on complex scenes. NeRF represents a static scene as per-point volume densities and view-dependent RGB colors. Given a 3D point $\mathbf{x} \in \mathbb{R}^3$ in space and a view direction $\mathbf{d} \in \mathbb{R}^3$, NeRF capitalizes on a multi-layer perceptron (MLP) to predict the volume density $\sigma(\mathbf{x}) \in \mathbb{R}$ and the emitted color $\mathbf{c}(\mathbf{x}, \mathbf{d}) \in \mathbb{R}^3$. To render a novel view for the scene,

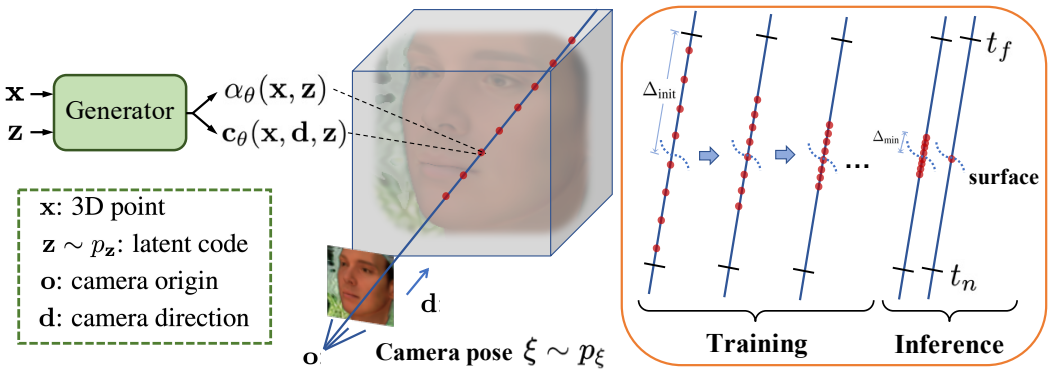


Figure 2: **Shrinking process**. During the training, the sampling interval Δ is initially a half of the distance between near t_n and far bounds t_f and shrinks gradually to a pre-defined value Δ_{min}. For inference, we can use cumulative rendering by sampling points in the minimal interval Δ_{min} and alternatively render only with the surface points.

NeRF leverages the classic volume rendering technique [39] to estimate the color of each pixel. It starts by accumulating the colored densities of N points $\{\mathbf{x}_i = \mathbf{o} + t_i\mathbf{d}\}$ sampled within near and far bounds $[t_n, t_f]$ along the camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, where $\mathbf{o}$ stands for the camera origin. The integrated color is then estimated via alpha composition as follows:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^{N} T_i\Big(1 - \exp\big(-\sigma(\mathbf{x}_i)\delta_i\big)\Big)\mathbf{c}(\mathbf{x}_i, \mathbf{d}), \text{ where } T_i = \exp(-\sum_{j=1}^{i-1}\sigma(\mathbf{x}_j)\delta_j), \tag{1}$$

where $\delta_i = |x_{i+1} - x_i|$ is the distance between adjacent points. Note that, equation (1) is naturally differentiable and NeRF can be directly optimized through the reconstruction error of existing views.

3.2 3D Surface-Aware Image Synthesis via Generative Occupancy Fields

To apply NeRF as the 3D representation, the proposed generative occupancy fields (GOF) incorporates an additional latent code $\mathbf{z} \sim p_\mathbf{z}$ into NeRF, such that synthesizing an image follows a reformulated cumulative rendering process:

$$\hat{\mathbf{C}}(\mathbf{r}, \mathbf{z}) = \sum_{i=1}^{N} T_i\Big(1 - \exp\big(-\sigma_\theta(\mathbf{x}_i, \mathbf{z})\delta_i\big)\Big)\mathbf{c}_\theta(\mathbf{x}_i, \mathbf{d}, \mathbf{z}), \text{ where } T_i = \exp(-\sum_{j=1}^{i-1}\sigma_\theta(\mathbf{x}_j)\delta_j). \tag{2}$$

However, directly training GOF according to Eq.(2) fails to maintain the surface compactness as reported in previous approaches [6, 7]. Actually, such a defect arises from an inevitable "shape-color ambiguity" of the cumulative rendering process, *i.e.*, small perturbations on surfaces still lead to realistic RGB images which are enough to fool the discriminator.

Owing to the constrained range of poses seen at training, the discriminator is less motivated to further concentrate the color weights $w_i = T_i\big(1 - \exp(-\sigma_\theta(\mathbf{x}_i, \mathbf{z})\delta_i)\big)$ aforementioned in Fig. 1 (a) on the exact object surface. On the other hand, we observe that although leading to diffused surfaces at the end, the variance of color weights w_i constantly decreases as the training proceeds. Inspired by this observation, in GOF we propose a training-time operation to facilitate the concentration of color weights w_i. The basic idea is gradually shrinking the sample region in the cumulative rendering process from the entire volume to a narrow interval around the surface, so that color weights are enforced to continuously move towards the exact surface.

To enable the proposed training-time shrinking process, GOF is required to locate the surface by thresholding the predicted densities $\sigma_\theta(\mathbf{x}, \mathbf{z})$, assuming points on the surface have the largest densities, However, values of the densities predicted in generative radiance fields could range from -10 to 50, making it hard to determine an effective threshold τ during the whole training period. On the other hand, in the cumulative rendering process shown in Eq.(2), we found that *the intermediate alpha values* used for numerical stability inherently fall in a fixed value range as:

$$\alpha_\theta(\mathbf{x}_i, \mathbf{z}) = 1 - \exp(-\sigma_\theta(\mathbf{x}_i, \mathbf{z})\delta_i) \in [0, 1]. \tag{3}$$

More importantly, these alpha values $\alpha_\theta(\mathbf{x}, \mathbf{z})$ come close to 1 for points in the occupied space while approaching 0 for points in the free space, making them resemble the occupancy values [8] in both quantity and semantics. Inspired by the similarity, we thus propose to reformulate generative radiance fields by predicting alpha values $\alpha_\theta(\mathbf{x}, \mathbf{z})$ directly instead of volume densities $\sigma_\theta(\mathbf{x}, \mathbf{z})$. In the mean time, we reinterpret the alpha values as occupancy values, and subsequently locate surfaces with root-finding, a more effective strategy originated in occupancy networks [11]. According to the above reformulation and reinterpretation, we thus dub our method as **Generative Occupancy Fields**.

As GOF estimates alpha values instead of volume densities, the original volume rendering process in Eq.(2) conditioned on the latent code $\mathbf{z}$ is reformulated as

$$\hat{\mathbf{C}}(\mathbf{r}, \mathbf{z}) = \sum_{i=1}^{N} \alpha_\theta(\mathbf{x}_i, \mathbf{z}) \prod_{j<i} \big(1 - \alpha_\theta(\mathbf{x}_j, \mathbf{z})\big) \mathbf{c}_\theta(\mathbf{x}_i, \mathbf{d}, \mathbf{z}), \tag{4}$$

where the value range of $\alpha_\theta(\mathbf{x}_i, \mathbf{z})$ is guaranteed with a sigmoid function. And to locate the surface via root-finding, for a specific ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ we will evenly sample M points $\{\mathbf{x}_k = \mathbf{o} + t_k\mathbf{d}; k = 1, ..., M\}$ that partition the entire volume $[t_n, t_f]$ into M equally-spaced bins. After obtaining the corresponding alpha values $\{\alpha_\theta(\mathbf{x}_k, \mathbf{z}); k = 1, ..., M\}$ by querying the generator g_θ, the surface $\mathcal{S}$ is located in the $k^{\mathcal{S}}$-th bin where α_θ changes *for the first time* from free space ($\alpha_\theta < \tau$) to occupied space ($\alpha_\theta < \tau$):

$$k^{\mathcal{S}} = \operatorname*{argmin}_k \big(\alpha_\theta(\mathbf{x}_k, \mathbf{z}) < \tau \leq \alpha_\theta(\mathbf{x}_{k+1}, \mathbf{z})\big), \tag{5}$$

where τ is a pre-defined threshold. In practice, we empirically set τ as 0.5. In order to find the surface point $\mathbf{x}_s = \mathbf{o} + t_s\mathbf{d}$ more precisely, we further apply the above secant method iteratively for m_s times, resulting in a fine-grained bin $[\mathbf{x}_{k^{\mathcal{S}}}, \mathbf{x}_{k^{\mathcal{S}}+1}]$. It's worth noting that the M sampled points are only used for root-finding. Thus they do not require the computation of gradients in the implementation.

Based on the located surface $\mathcal{S}$, we can thus successfully conduct the proposed shrinking process, which is schematically elaborated in Fig. 2. Specifically, when sampling N points for Eq.(4) at each training step, we will only sample within a region neighboring the surface $[t_s - \Delta, t_s + \Delta]$:

$$t_i \sim \mathcal{U}\left[t_s - \Delta + \frac{2i}{N}\Delta, t_s - \Delta + \frac{2i+2}{N}\Delta\right], \text{ where } i = 0, 1, ..., N-1. \tag{6}$$

Δ is the sampling interval, which is set to $\Delta_{\text{init}} = (t_f - t_n)/2$ at the beginning, a half of the distance between near t_n and far bounds t_f. And it will decrease monotonically with an exponential decay rate γ until it drops to a pre-defined minimal value Δ_{min}. Formally, $\Delta_n = \max(\Delta_{\text{init}} \exp(-\gamma n), \Delta_{\text{min}})$ for n-th decaying step. Additionally, during training when the estimated t_s is too close to the near or the far bound so that the sampling region $[t_s - \Delta, t_s + \Delta]$ exceeds the original range $[t_n, t_f]$, we will shift the region $[t_s - \Delta, t_s + \Delta]$ back to within $[t_n, t_f]$. As shown in Fig. 2, at the beginning of training, points sampled for Eq.(4) will cover the entire volume, leading to dispersed gradients which facilitate the convergence of GOF. And as the training goes, the predicted surface will become more and more accurate, which is the outcome of gradually refining the sampling region, and in turn also makes the above shrinking operation valid.

Thanks to the dedicated shrinking process, the color weights w_i can successfully concentrate on the object surface as illustrated in Fig. 1(a). As a result, GOF is capable of synthesizing high-fidelity images in a 3D-consistent manner and simultaneously capturing compact object surfaces. During inference, to synthesize an image under a random camera pose $\xi \sim p_\xi$, the generator g_θ will fetch a truncated latent code $\hat{\mathbf{z}}$ and sample N points $\{\mathbf{x}_i\}$ on each ray within the minimal region $[t_s - \Delta_{\text{min}}, t_s + \Delta_{\text{min}}]$ for the rendering as in Eq.(4). An important benefit of learning a compact object surface is that we can effectively reduce the number of sampled points for rendering, even using only one point on each ray, *i.e.* the surface point. As shown in Fig. 1 (b), the image rendered with only the surface point is almost indistinguishable from that with multiple points. Such equivalence can be guaranteed theoretically when $\Delta_{\text{min}} \to 0$, and we include the proof in the supplementary material.

3.3 Loss Functions

Instead of training on posed 2D images, the proposed GOF leverages a corpus of unposed images for 3D-aware image synthesis, where multiple loss functions are adopted.

GAN Loss. Following pi-GAN [6], a GAN loss is used where GOF synthesizes fake images by randomly sampling camera poses ξ from a dataset-related distribution p_ξ and rendering according to Eq.(4). Denote I as a real image from the data distribution $p_\mathcal{D}$, the non-saturating GAN loss can be described as follows:

$$\begin{aligned}\mathcal{L}_{\text{origin}}(\theta_D, \theta_G) = \mathbf{E}_{\mathbf{z}\sim p_\mathbf{z}, \xi\sim p_\xi}\left[f\Big(D_{\theta_D}(G_{\theta_G}(\mathbf{z}, \xi))\Big)\right] \\ + \mathbf{E}_{I\sim p_\mathcal{D}}\left[f(-D_{\theta_D}(I)) + \lambda|\nabla D_{\theta_D}(I)|^2\right], \\ \text{where } f(u) = -\log(1+\exp(-u)).\end{aligned} \tag{7}$$

However, $\mathcal{L}_{\text{origin}}$ alone is not sufficient to guide the training, which may lead to messy images with smoke-like artifacts. Therefore, two more regularizations are incorporated to reduce artifacts and further smooth the learned surfaces.

Normal Regularization. The first regularization is a prior on the surface normal smoothness, which is specially useful for learning from 2D real-world images [11]. In GOF, this normal prior is only employed for the surface points $\mathbf{x}_s \in \mathcal{S}$ to encourage a natural and smooth surface:

$$\mathcal{L}_{\text{normal}} = \sum_{\mathbf{x}_s\in\mathcal{S}} ||\mathbf{n}_\theta(\mathbf{x}_s, \mathbf{z}) - \mathbf{n}_\theta(\mathbf{x}_s + \boldsymbol{\epsilon}, \mathbf{z})||_2, \tag{8}$$

where $\boldsymbol{\epsilon}$ is a small random 3D perturbation and $\mathbf{n}_\theta$ denotes the normal vector, which can be computed by $\mathbf{n}_\theta(\mathbf{x}, \mathbf{z}) = \nabla_\mathbf{x}\alpha_\theta(\mathbf{x}, \mathbf{z})/||\nabla_\mathbf{x}\alpha_\theta(\mathbf{x}, \mathbf{z})||_2$.

Opacity Regularization. Since alpha values predicted in GOF can be regarded as occupancy values, ideally the entropy of them should be 0 so that $\alpha_\theta(\mathbf{x}, \mathbf{z})$ values will equal 1 for points in the occupied space and 0 for points in the free space. We thus apply the second opacity regularization, aiming to reduce the entropy of predicted alpha values:

$$\mathcal{L}_{\text{opacity}} = \frac{1}{N}\sum_{i=1}^{N} \log(\alpha_\theta(\mathbf{x}_i, \mathbf{z})) + \log(1 - \alpha_\theta(\mathbf{x}_i, \mathbf{z})). \tag{9}$$

In summary, the final loss function for training GOF can be written as:

$$\mathcal{L}(\theta, \phi) = \mathcal{L}_{\text{origin}}(\theta, \phi) + \lambda_{\text{normal}}\mathcal{L}_{\text{normal}} + \lambda_{\text{opacity}}\mathcal{L}_{\text{opacity}}, \tag{10}$$

where λ_{normal} and λ_{opacity} are both balancing coefficients.

4 Experiments

Implementation Details. Unless stated otherwise, in all experiments we set N, the number of points sampled for rendering, to 12, and set M, the number of bins used in root-finding, to 12. As for the near and far bounds, we set t_n and t_f respectively to 0.88 and 1.12. As discussed in Sec.3.2, we apply an iterative process in root-finding. In practice, the number of iterations is set to $m_s =$ times. During the inference, GOF requires $M + m_s + N$ queries to obtain the color of a pixel, while existing methods require $2N$ queries due to the use of a hierarchical sampling strategy. Recall it is sufficient for GOF to sample only the surface point to render an image, GOF is thus capable of using just $M + m_s + 1$ queries, potentially speeding up the rendering process in off-line applications. In Table 2 we also include the values of important dataset-dependent hyperparameters of GOF. More training and implementation details can be found in the supplemental material.

Datasets. To assess our method comprehensively, We conduct experiments on three datasets, namely CelebA [40], BFM [41], and Cats [42]. Specifically, CelebA is a high-resolution face dataset containing $200,000$ diverse face images. Following pi-GAN [6], we crop all images in CelebA from the top of the hair to the bottom of the chin as a pre-processing step. As for the Cats dataset, it contains $6,444$ cat faces of size 128×128. Finally, BFM is a synthetic face dataset rendered with Basel Face Model, where each face is paired with a ground-truth depth map, making it a good benchmark for quantitatively evaluating the quality of learned surfaces.

Comparison with baselines. To validate the effectiveness of GOF, we compare it with two representative GRAF methods, namely GRAF [7] and pi-GAN. Firstly, Fig. 3 demonstrates the qualitative comparison between these three methods, where we include the synthesized images, the learned

Table 1: **Quantitative results** on BFM, CelebA and Cats datasets, on three metrics Frechet Inception Distance (FID), Inception Score (IS) and the variance of color weights $\Sigma_{w_i}(\times 10^{-3})$.

	BFM			CelebA			Cats		
	FID↓	IS↑	Σ_{w_i} ↓	FID↓	IS↑	Σ_{w_i} ↓	FID↓	IS↑	Σ_{w_i} ↓
GRAF [7]	50.2	1.27	4.114	41.4	1.86	6.512	28.6	1.65	1.053
pi-GAN [6]	17.4	2.35	3.832	15.8	2.55	6.095	17.6	1.99	0.863
Ours (w/o priors)	16.4	2.66	1.443	17.2	2.44	4.782	19.3	1.86	0.591
Ours	**15.3**	**2.89**	**1.426**	**15.0**	**2.62**	**4.115**	**17.1**	**2.09**	**0.533**

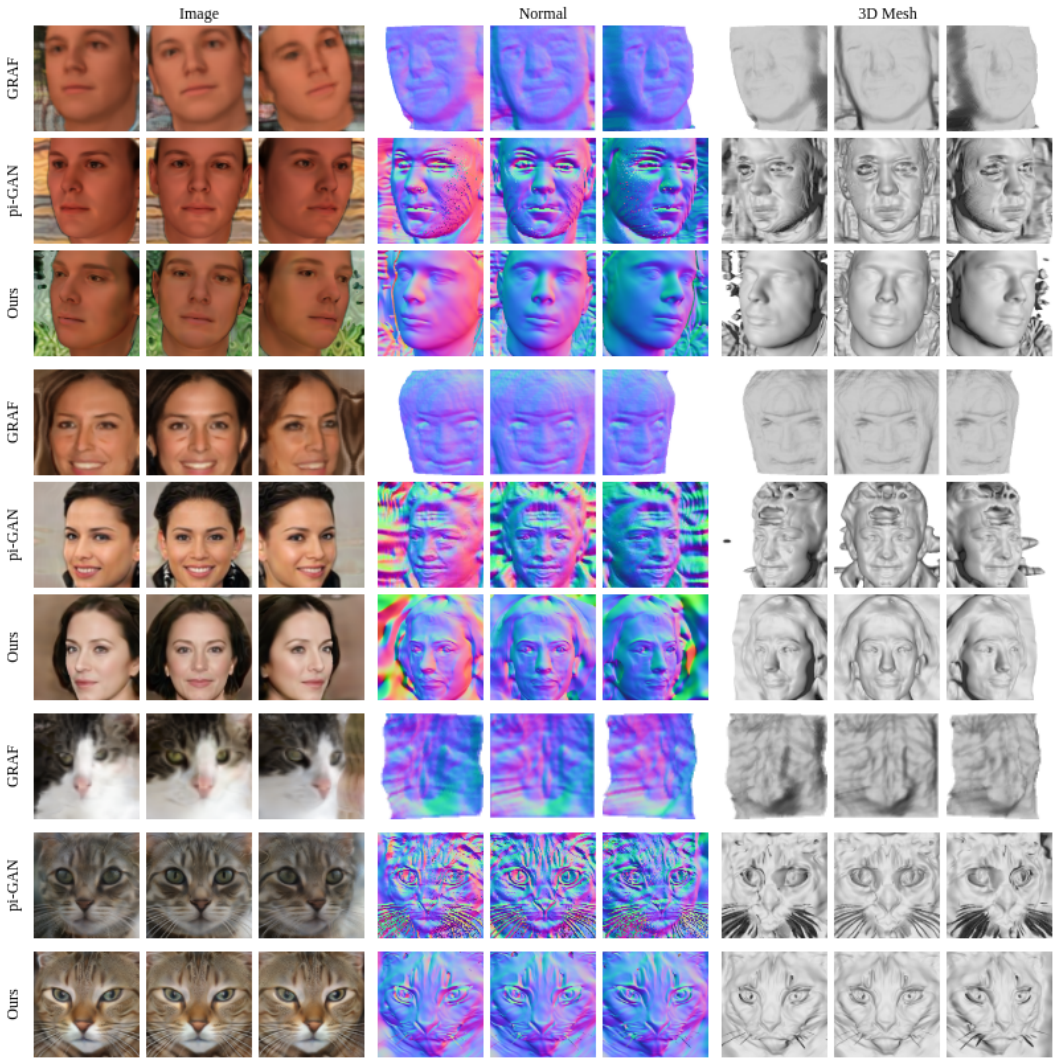


Figure 3: **Qualitative comparison** on BFM (top), CelebA (middle), and Cats (bottom) datasets. Our method synthesizes realistic images while ensuring compact object surfaces.

surfaces in the form of 3D meshes, as well as the corresponding normal maps. As can be observed, GRAF struggles to render good images, let alone estimating compact and reasonable underlying surfaces. Compared to GRAF, pi-GAN can synthesize images and estimate corresponding surfaces with improved quality. However, messy parts can be clearly recognized on its learned surfaces and normal maps, indicating it is incapable of capturing the compact 3D geometric details. In contrast to both pi-GAN and GRAF, the proposed GOF is shown to hallucinate realistic images with 3D consistency and simultaneously learn smooth surface normals as well as compact object surfaces, which verifies the benefit of adopting the transition from radiance fields to occupancy fields. More qualitative results of synthesized images and corresponding surfaces are included in Fig. 4.

Table 2: The setting of several important dataset-dependent hyperparameters.

dataset	BFM	CelebA	Cats
$\gamma(\times 10^{-5})$	4.0	1.0	2.0
Δ_{min}	0.01	0.02	0.04
λ_{opacity}	0.2	0.01	0.02
λ_{normal}	0.002	0.05	0.05
iterations	80K	200K	80K

Table 3: Comparisons on the compactness and accuracy of learned surfaces.

Method	SIDE ↓	MAD ↓
Supervised	0.412	10.84
Unsup3d	0.795	16.51
GRAF	1.866	26.69
pi-GAN	0.757	20.46
GAN2Shape	0.779	14.94
Ours	0.807*	**14.11**

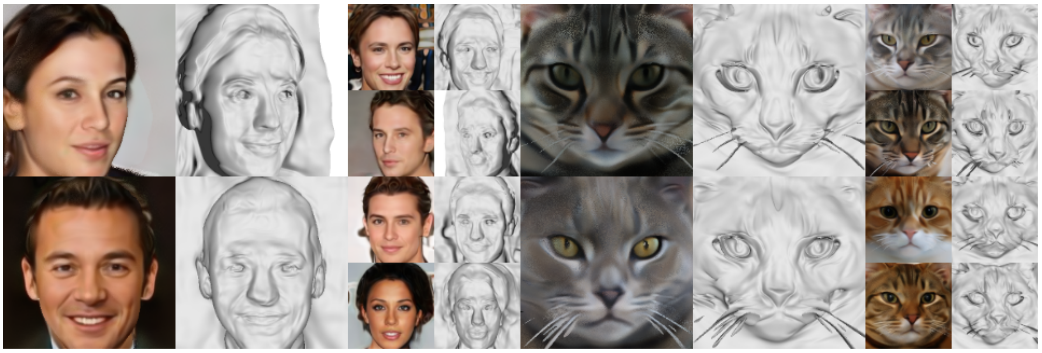
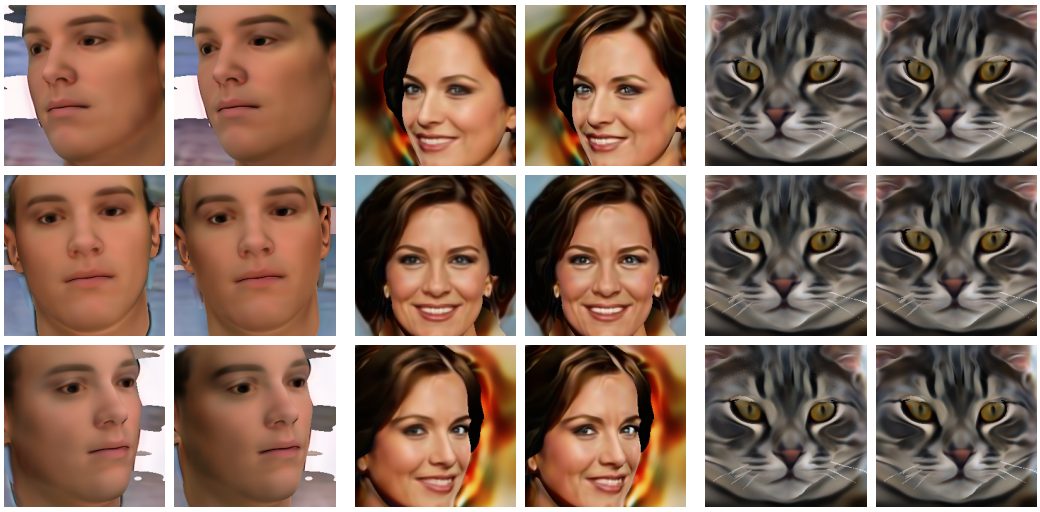


Figure 4: Generated images and their 3D meshes on CelebA and Cats datasets.



BFM CelebA Cats

Figure 5: **Rendering with only surface points.** Images (right) rendered only with surface points are indistinguishable from those (left) obtained with cumulative rendering.

To quantitatively evaluate the quality of generated images, we report the Frechet Inception Distance (FID) scores and Inception Score (IS) scores in Table 1. Specifically, GOF demonstrates substantial improvements over baseline methods across both of these two metrics. To further measure the compactness of learned surfaces, the concentration of color weights w_i as shown in Sec. 3.2 is computed as the evaluation metric. Specifically, we calculate the color weights w_i of N uniformly sampled points in the entire volume as well as 1 surface point $\mathbf{x}_s$ for each ray, and the variance of these estimated w_i is regarded as the concentration of color weights of a single image. Intuitively, a smaller variance implies the learned surface is more compact. While in GOF the surface point is located via root-finding, in pi-GAN and GRAF, the surface point is located by the weighted depth.

(a) BFM (b) CelebA

Figure 6: **Qualitative ablation on proposed priors.** Please refer to the main text for details.

Finally, for each method, the overall concentration of color weights is averaged over 1000 randomly synthesized images. The results in terms of this new metric are also included in Table 1.

We further evaluate the compactness and accuracy of learned surfaces on the BFM dataset, since it contains ground-truth depth maps. Specifically, $50K$ images are generated by each method, together with their corresponding depth maps[1]. For each method, we train a separate CNN on these generated images and depth maps to predict depths from images. Subsequently, we can measure the accuracy of learned surfaces by running the CNN on the test split of BFM and comparing its outputs to the ground-truth depth maps using the scale-invariant depth error (SIDE) and the mean angle deviation (MAD). While MAD focuses more on the compactness of surfaces, SIDE emphasizes more on the accuracy of surfaces. As shown in Table 3, GOF significantly outperforms baseline methods on the MAD metric, and is comparable to strong baselines on the SIDE metric.

Rendering only with surface points. As mentioned in Sec. 3.2, GOF is able to render an image using only the surface points. To verify this, we showcase in Fig. 5 images rendered by GOF using multiple points and only the surface point. As can be observed, images synthesized with these two strategies are nearly indistinguishable from each other. Thus, GOF possesses the potential to significantly reduce the number of generator queries when synthesizing an image. Owing to the reduction on the burden of queries, GOF enables a light rendering scheme that is promising for applications on mobile devices.

Ablation studies. We here analyze the effects of the proposed priors λ_{normal} and λ_{opacity}. Table 1 include the results of GOF without both these priors, where a moderate performance degradation is observed. We also include qualitative samples in Fig. 6, which contains images synthesized by GOF with and without these priors. As shown by the smoke-like artifacts around the cheek part and the hollows on the face surfaces, the proposed priors are beneficial for synthesizing realistic images and learning compact surfaces. It is worth noting that although the performance of GOF is deteriorated due to the absence of these priors, images and surfaces produced by GOF are still of reasonable quality when compared to that from previous approaches, indicating the transition from radiance fields to occupancy fields is the main cause that leads to the success of GOF.

5 Conclusion

In this work, we propose generative occupancy fields (GOF), a novel generative radiance fields for 3D-aware image synthesis. The crux of GOF is a dedicated transition from the cumulative rendering in radiance fields to rendering with only the surface points. Such a transition is inspired by the resemblance between the alpha values in radiance fields and the occupancy values in occupancy networks, so that we can reinterpret one as the other. In practice, such a transition is achieved during training by gradually shrinking the sampling region in the rendering process of GOF from the entire volume to a minimal neighboring region around the surface, where the surface is located via root-finding on predicted alpha values. Thanks to the transition, surfaces learned by GOF continuously converge during the training, ensuring their compactness at the end. On three diverse datasets, GOF is shown to demonstrate great superiority in synthesizing 3D consistent images and in the meantime capturing compact surfaces, significantly broadening the application of generative radiance fields in downstream tasks.

[1]The details of obtaining depth maps are included in the supplemental material.

References

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014.

[2] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," in *International Conference on Learning Representations*, 2018.

[3] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019.

[4] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8110–8119, 2020.

[5] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, pp. 405–421, Springer, 2020.

[6] E. R. Chan, M. Monteiro, P. Kellnhofer, J. Wu, and G. Wetzstein, "pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis," in *CVPR*, 2021.

[7] K. Schwarz, Y. Liao, M. Niemeyer, and A. Geiger, "Graf: Generative radiance fields for 3d-aware image synthesis," in *NeurIPS*, vol. 33, 2020.

[8] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4460–4470, 2019.

[9] L. Yariv, Y. Kasten, D. Moran, M. Galun, M. Atzmon, B. Ronen, and Y. Lipman, "Multiview neural surface reconstruction by disentangling geometry and appearance," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[10] M. Oechsle, S. Peng, and A. Geiger, "Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction," *arXiv preprint arXiv:2104.10078*, 2021.

[11] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3504–3515, 2020.

[12] Y. Jiang, D. Ji, Z. Han, and M. 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.

[13] S. Liu, Y. Zhang, S. Peng, B. Shi, M. Pollefeys, and Z. Cui, "Dist: Rendering deep implicit signed distance function with differentiable sphere tracing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2019–2028, 2020.

[14] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 165–174, 2019.

[15] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5939–5948, 2019.

[16] K. Genova, F. Cole, D. Vlasic, A. Sarna, W. T. Freeman, and T. Funkhouser, "Learning shape templates with structured implicit functions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7154–7164, 2019.

[17] M. Michalkiewicz, J. K. Pontes, D. Jack, M. Baktashmotlagh, and A. Eriksson, "Implicit surface representations as layers in neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4743–4752, 2019.

[18] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li, "Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2304–2314, 2019.

[19] S. Saito, T. Simon, J. Saragih, and H. 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*, pp. 84–93, 2020.

[20] V. Sitzmann, E. R. Chan, R. Tucker, N. Snavely, and G. Wetzstein, "Metasdf: Meta-learning signed distance functions," *arXiv preprint arXiv:2006.09662*, 2020.

[21] L. Liu, J. Gu, K. Z. Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," *arXiv preprint arXiv:2007.11571*, 2020.

[22] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, "Plenoctrees for real-time rendering of neural radiance fields," *arXiv preprint arXiv:2103.14024*, 2021.

[23] T. Neff, P. Stadlbauer, M. Parger, A. Kurz, J. H. Mueller, C. R. A. Chaitanya, A. Kaplanyan, and M. Steinberger, "Donerf: Towards real-time rendering of compact neural radiance fields using depth oracle networks," *arXiv preprint arXiv:2103.03231*, 2021.

[24] P. Hedman, P. P. Srinivasan, B. Mildenhall, J. T. Barron, and P. Debevec, "Baking neural radiance fields for real-time view synthesis," *arXiv preprint arXiv:2103.14645*, 2021.

[25] M. Boss, R. Braun, V. Jampani, J. T. Barron, C. Liu, and H. Lensch, "Nerd: Neural reflectance decomposition from image collections," *arXiv preprint arXiv:2012.03918*, 2020.

[26] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. T. Barron, "Nerv: Neural reflectance and visibility fields for relighting and view synthesis," *arXiv preprint arXiv:2012.03927*, 2020.

[27] M. Atzmon, N. Haim, L. Yariv, O. Israelov, H. Maron, and Y. Lipman, "Controlling neural level sets," *arXiv preprint arXiv:1905.11911*, 2019.

[28] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling," in *NIPS*, 2016.

[29] X. Chen, D. Cohen-Or, B. Chen, and N. J. Mitra, "Towards a neural graphics pipeline for controllable image generation," *Computer Graphics Forum*, vol. 40, no. 2, 2021.

[30] J.-Y. Zhu, Z. Zhang, C. Zhang, J. Wu, A. Torralba, J. Tenenbaum, and B. Freeman, "Visual object networks: Image generation with disentangled 3d representations," in *NeurIPS*, 2018.

[31] Y. Liao, K. Schwarz, L. Mescheder, and A. Geiger, "Towards unsupervised learning of generative models for 3d controllable image synthesis," in *CVPR*, pp. 5871–5880, 2020.

[32] T. Nguyen-Phuoc, C. Richardt, L. Mai, Y.-L. Yang, and N. Mitra, "Blockgan: Learning 3d object-aware scene representations from unlabelled images," in *NeurIPS*, Nov 2020.

[33] T. Nguyen-Phuoc, C. Li, L. Theis, C. Richardt, and Y.-L. Yang, "Hologan: Unsupervised learning of 3d representations from natural images," in *ICCV*, pp. 7588–7597, 2019.

[34] M. Niemeyer and A. Geiger, "Giraffe: Representing scenes as compositional generative neural feature fields," in *CVPR*, 2021.

[35] S. Wu, C. Rupprecht, and A. Vedaldi, "Unsupervised learning of probably symmetric deformable 3d objects from images in the wild," in *CVPR*, 2020.

[36] X. Pan, B. Dai, Z. Liu, C. C. Loy, and P. Luo, "Do 2d gans know 3d shape? unsupervised 3d shape reconstruction from 2d image gans," in *ICLR*, 2021.

[37] M. Sahasrabudhe, Z. Shu, E. Bartrum, R. Alp Guler, D. Samaras, and I. Kokkinos, "Lifting autoencoders: Unsupervised learning of a fully-disentangled 3d morphable model using deep non-rigid structure from motion," in *ICCV Workshops*, pp. 0–0, 2019.

[38] X. Li, S. Liu, K. Kim, S. De Mello, V. Jampani, M.-H. Yang, and J. Kautz, "Self-supervised single-view 3d reconstruction via semantic consistency," *ECCV*, 2020.

[39] J. T. Kajiya and B. P. Von Herzen, "Ray tracing volume densities," *ACM SIGGRAPH computer graphics*, vol. 18, no. 3, pp. 165–174, 1984.

[40] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *ICCV*, pp. 3730–3738, 2015.

[41] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3d face model for pose and illumination invariant face recognition," in *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 296–301, Ieee, 2009.

[42] W. Zhang, J. Sun, and X. Tang, "Cat head detection-how to effectively exploit shape and texture features," in *ECCV*, pp. 802–816, Springer, 2008.

Checklist

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] Please refer to the supplementary material.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See the supplementary material.
   (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? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
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] We provide implementation details in the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See the supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] This is a common practice [6, 7] as training generative implicit model is time-consuming.
   (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 the supplementary material.
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]
   (b) Did you mention the license of the assets? [Yes] See the supplementary material.
   (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] See the supplementary material.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See the supplementary material.
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]