Diffusion-SDF: Conditional Generative Modeling of Signed Distance Functions

Gene Chou   Yuval Bahat   Felix Heide

Princeton University

Abstract

Probabilistic diffusion models have achieved state-of-the-art results for image synthesis, inpainting, and text-to-image tasks. However, they are still in the early stages of generating complex 3D shapes. This work proposes Diffusion-SDF, a generative model for shape completion, single-view reconstruction, and reconstruction of real-scanned point clouds. We use neural signed distance functions (SDFs) as our 3D representation to parameterize the geometry of various signals (e.g., point clouds, 2D images) through neural networks. Neural SDFs are implicit functions and diffusing them amounts to learning the reversal of their neural network weights, which we solve using a custom modulation module. Extensive experiments show that our method is capable of both realistic unconditional generation and conditional generation from partial inputs. This work expands the domain of diffusion models from learning 2D, explicit representations, to 3D, implicit representations. Code is released at https://github.com/princeton-computational-imaging/Diffusion-SDF

1. Introduction

Diffusion probabilistic models [51, 18] have become a popular choice for generative tasks and can produce impressive results, such as the images generated by DALLE-2 [44] and Stable Diffusion [46] from text input. Diffusion models are a type of likelihood-based models whose training objective can be expressed as a variational lower bound [18, 53]. On a high level, they learn to gradually remove noise from a signal and repeat this process to generate samples from Gaussian noise. Recent advances [38, 10, 44, 46] show that diffusion models produce images with quality on par with state-of-the-art generative adversarial networks (GANs) [15] without the common drawbacks of mode collapse [38, 37] and unstable training [36, 2]. Diffusion has also applied to 3D tasks although these works are still in the early stages of producing complex shapes. In this work, we investigate the generation of 3D shapes of neural signed distance functions via diffusion.

3D modeling and generation are essential to vision and graphics tasks. 3D generation of high-quality assets and large volumes of realistic data is often essential where training data is expensive to collect [24, 27, 47, 34, 48, 3]. Additionally, generation can be applied to 3D reconstruction of imperfect visual observations as there exists a one-to-many mapping that requires a probabilistic approach to solve. This has applications in self-driving [54, 67, 39] and robotics grasping [4, 22, 65] where occlusion and camera measurement errors are common.

We propose Diffusion-SDF, a generative model for shape completion, single-view reconstruction, and reconstruction of real-scanned point clouds. We choose neural signed distance functions (SDFs) [40] as the 3D representation to parameterize the surfaces described by various input signals such as point clouds and 2D images. They implicitly encode an object surface by the signed distances between 3D coordinate queries to their closest surface point through a coordinate-based MLP [40, 8]. Compared to discrete 3D representations [25, 30, 43, 13, 16], SDFs have proven to be a versatile representation that supports arbitrary resolution during test-time [55], small memory footprints [9], and strong generalization [8].

We make the following two key insights. First, implicit functions can directly be used as data and diffusing them amounts to learning the reversal of the neural network weights. Furthermore, we introduce geometrical constraints to produce complex shapes and outputs consistent with the geometry of conditioned inputs. Very recently, Dupont *et al*. [11] similarly diffuse implicit functions but do not address SDFs nor geometric constraints. Second, by using SDFs as a unified 3D representation, we condition training to learn a mapping between various input types and their possible reconstructions. Then, we leverage a probabilistic diffusion model to generate diverse completions. Thus, our work can be applied to shape synthesis and multi-modal shape completion.

The proposed method consists of two steps, shown in Fig. 2. First, we create a compressed representation of SDFs using modulation [11, 5, 31]. We find that diffusing SDFs is impractical due to the large number of parameters

Figure 1. Our method generates clean meshes with diverse geometries. (**Top**) Unconditional generations from training on multiple classes. (**Bottom**) Conditional generation given various visual inputs, such as partial point clouds (same point cloud overlaid on sample), real-scanned point clouds, and 2D images. Our method captures details of conditioned geometry, such as the handle of the pitcher.

and the lack of a smoothed data distribution. Our modulation module consists of learning a generalizable encoder and a regularized latent space to create latent vectors that map to individual SDFs when combined with an SDF base network. Second, we train a diffusion model with the previously created latent vectors as data points. We follow the conventional approach of learning the reverse diffusion process [18, 44], but we combine it with our modulation scheme to introduce geometric information. We show that this geometric constraint is essential for the method to complete shapes consistent with guided inputs. Furthermore, we experiment with various input types for guiding generation. Shown in Fig. 1, we validate the method by shape generation and completion with conditioning of partial point clouds from Acronym [12], real-scanned point clouds from YCB [4], and 2D images from ShapeNet [6]. Our method generates diverse and realistic shapes for multiple tasks. We make the following contributions:

- We propose a probabilistic generative model that creates clean and diverse 3D meshes.

- We solve a learning problem of diffusing the weights of implicit neural functions while providing geometric guidance through our modulation module.

- Our method reconstructs plausible outputs from various imperfect observations such as sparse, partial point clouds, single images, and real-scanned point clouds.

- Extensive experiments show that our method achieves favorable performance in shape generation and completion compared to existing methods.

2. Related Work

Diffusion Probabilistic Models Diffusion probabilistic models [51, 18] generate samples from a distribution by learning to gradually remove noise from a datapoint. Recent advances [38, 10, 44, 46] show diffusion models produce high quality images without the drawbacks of mode collapse [38, 37] and unstable training [36, 2]. Diffusion has also been applied to 3D tasks although these works are still in the early stages of producing complex shapes.

One line of work [69, 28, 68] trained diffusion models to generate point clouds. Very recently, Dupont *et al.* [11] trained diffusion models on implicit neural representations but not SDFs. These existing methods perform unconditional generation and some only produce simple geometries. Our modulation scheme and conditioning mechanisms fix both of these issues. We also acknowledge concurrent works that combine diffusion and implicit functions; [42, 61] generate novel views from text and images with and without intermediate radiance fields, respectively.

Generative Modeling of 3D Shapes Many existing works [7, 45, 66, 60] that reconstruct partial scans and meshes are deterministic, but the relation between partial and completed shapes is a one-to-many mapping. To address this, [35, 63, 62, 69] proposed probabilistic models for generating multi-modal reconstructions that are consistent with partial inputs. Mittal *et al.* [35] train an autoregressive prior, Wu *et al.* [62] train a conditional GAN, Yan *et al.* [63] train a vector quantized deep implicit function (VQDIF), and Zhou *et al.* [69] train a diffusion model. We compare our method to these baselines.

More recently, some works [14, 20] have combined SDFs and generative modeling. Gao *et al.* [14] train on 2D image collections and combine differentiable rendering and 2D GANs. Hui *et al.* [20] is concurrent work that learns diffusion models. They convert SDFs into wavelet representations, then use them as input to diffusion models. Different from the proposed approach, this method cannot perform

conditional generation and learns to reverse wavelets as a surrogate to implicit functions.

Learning Implicit Signed Distance Functions   A rapidly growing body of work relies on implicit neural networks as an expressive scene representation that facilitates learning for 3D reconstruction and view synthesis tasks [40, 32, 33]. They use neural networks to map spatial coordinates to scene attributes, which offers a fully-differentiable and versatile way to represent 3D geometry. Park *et al.* [40] first proposed to map coordinates to signed distance values and reconstructed surfaces by interpolating grid points with signed distance values of zero.

Follow-up works [8, 41, 29] use point clouds as additional conditions to achieve greater detail, generalization, and unsupervised training. However, they condition their input on *full-view* point clouds with low levels of noise. Thus, they fail on real-world applications where one can only obtain partial and noisy point clouds. In this work, we close this gap and introduce a generative model that reconstructs plausible outputs for partial and noisy point clouds.

3. Diffusion Models and Neural SDFs

Background on Diffusion Probabilistic Models   While different variations of diffusion models exist, we describe a canonical one [51, 18]. From a data distribution $q(\mathbf{x})$, we denote a sampled datapoint as x_0, and iteratively add small Gaussian noise to obtain $x_1, x_2 ... x_T$, until x_T approximates an isotropic Gaussian. This forward step is a Markovian fixed process [18, 53] and can be defined as $q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$ and $q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$ where β_t is a variance schedule. In practice, we sample x_t using a closed form parameterization $\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$ where $\alpha_t = 1-\beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t}\alpha_i$, and $\epsilon \sim \mathcal{N}(0, I)$.

The goal of each training iteration is to train a model p_θ, often represented by a neural network, that inverts the forward diffusion (i.e., learns the *reverse* diffusion process): $p_\theta(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$ and $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$. The reverse process is also Markovian and we fix the variances Σ_θ. The reverse conditional probability is tractable when conditioned on x_0: $q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}(x_t, x_0), \tilde{\beta}_t I)$. We apply Bayes' rule to rearrange the terms and represent $\tilde{\mu}(x_t, x_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0$. The closed form parameterization of x_t yields $\tilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t)$.

Thus, we can train our model to predict $\tilde{\mu}_t$, or alternatively, ϵ_t by rearranging the terms. This work predicts $\tilde{\mu}_t$ for generating SDF samples.

For generation, we start with $x_T \sim \mathcal{N}(0, I)$ and iteratively denoise x_T with ancestral sampling [18]: $x_{t-1} = \mu_\theta(x_t, t) + \sigma_t \epsilon$ where σ_t is the fixed standard deviation at timestep t and $\epsilon \sim \mathcal{N}(0, I)$ is injected until the last step.

Background on Neural SDFs   While many works overfit SDFs to a single object [9, 55, 50], some have been able to learn SDFs conditioned on point cloud inputs that generate shapes from different categories. A successful approach is jointly training a PointNet encoder [43] and an SDF decoder [40, 8, 41], where shape features from the encoder are concatenated with 3D query points $x \in \mathbb{R}^3$ as input to the decoder. We denote the funcion $x, P \mapsto \Phi(x, P) = s$, where $P = \{p_i \in \mathbb{R}^3\}_{i=1}^{N}$ is a raw point cloud with N points, $\Phi : \mathbb{R}^3 \times \mathbb{R}^{3 \times N} \to \mathbb{R}$ is the SDF that predicts the signed distance value for a 3D coordinate, conditioned on a point cloud, and s denotes the predicted signed distance value between x and the shape described by P. The surface boundary of the shape is its zero-level set $S_0(\Phi(P))$, which can be formulated as $S_0(\Phi(P)) = \{z \in \mathbb{R}^3 \mid \Phi(z, P) = 0\}$.

4. Diffusing Neural Signed Distance Functions

Illustrated in Fig. 2, the proposed method is composed of three major components: a modulation scheme to represent SDFs as individual latent vectors, a diffusion model that takes the latent vectors as distribution samples for training, and a custom encoder and attention mechanism for conditional generation. In the following, we describe the components of our method in detail.

4.1. Modulating SDFs

We use modulation [11, 5, 31] to create an alternate representation of SDFs. Directly diffusing thousands of SDFs, where one SDF represents one object, is difficult because one must first train all SDFs separately (which would take thousands of GPU hours) and the distribution of thousands of SDFs is challenging to learn. We show in our supplement that existing diffusion models cannot directly learn from SDFs as training data. Thus, we map SDFs, represented by MLPs, to 1D latent vectors with two objectives: the diffusion model needs to learn and sample from the distribution of latent vectors effectively, and generated outputs of the diffusion model are mapped back into an SDF. This amounts to designing a latent space that needs to be continuous (interpolation between latent vectors corresponds to interpolation of geometry), complete (all points in the latent space are meaningful), and sufficiently diverse for holding information of hundreds of categories.

To this end, we jointly train a conditional SDF representation and a VAE [21]. We opt for the architecture of GenSDF [8], which is capable of learning hundreds of diverse categories using a unified model, so we train one model instead of thousands of SDFs. Specifically, our modulation module (see Fig. 2), consisting of a PointNet encoder Ψ and a VAE Θ, takes in a raw point cloud $P =$
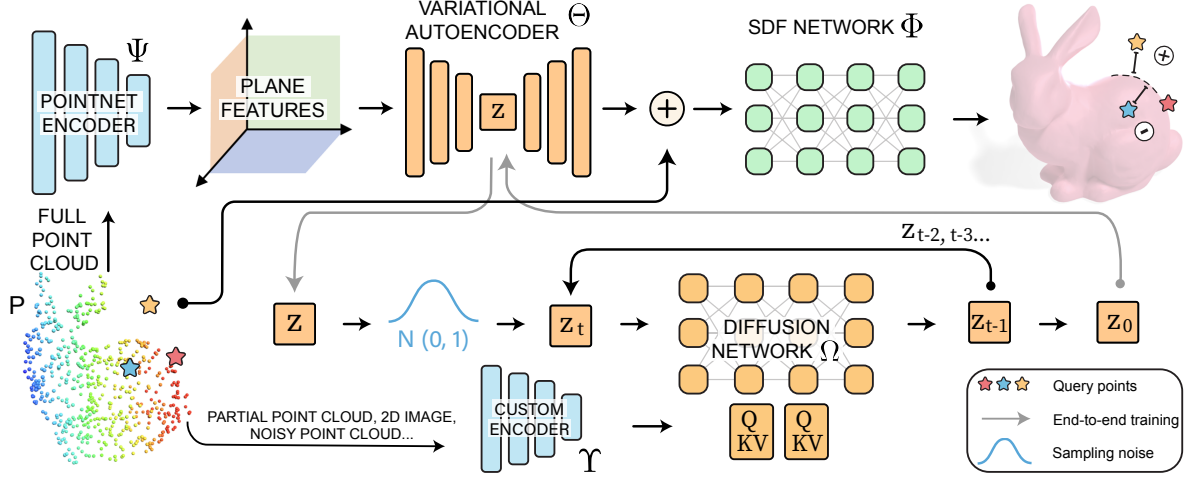
Figure 2. Our two-stage training pipeline. The first (**top**) trains SDFs jointly with a VAE [21] to produce latent vectors **z** each representing an SDF embedding. The second stage (**bottom**) uses the latent vectors as input to our diffusion model and can be guided by various inputs. We connect the two models (**gray arrow**) for end-to-end training. During test time, the diffusion model takes input z sampled from a Gaussian distribution and we combine its output with the SDF network to form a complete SDF representation.

$\{p_i \in \mathbb{R}^3\}_{i=1}^N$ with N points, and outputs plane features π and π' and a latent vector z as follows

$$\pi = \Psi(P), \ z = \Theta_{enc}(\pi), \ \pi' = \Theta_{dec}(z), \qquad (1)$$

where $\Theta_{enc}, \Theta_{dec}$ are the encoder and decoder of the VAE, respectively. Equivalently, $\pi' = (\Theta \circ \Psi)(P)$. Other than compression, the VAE regularizes the latent space. Next, we pass the concatenation of query points $x \in \mathbb{R}^3$ and π' into the SDF network Φ. We denote the predicted signed distance value as $s = \Phi(x|z)$. This formulation allows us to swap out different latent vectors z for producing different shape representations, including generated latents from the diffusion model which we show later.

The training objective of this latent parameterization is to learn accurate predictions of signed distance values and regularize the latent space of the VAE as

$$\mathcal{L}_{\text{mod}} = \|\Phi(x|z) - \text{SDF}(x)\|_1 + \beta(D_{KL}(q_\phi(z|\pi)\|p(z))). \tag{2}$$

The first term of the RHS of Eq. (2) is an $\mathcal{L}_1$ loss between the predicted and ground truth signed distance values of our query points x. Here, $\text{SDF}(\cdot)$ denotes the ground-truth SDF operator that is defined for all $x \in \mathbb{R}^3$. The second term is our KL-divergence loss [21] that regularizes the generated latent space to approach a target distribution. For given point cloud features π, we describe the inferred posterior of the latent vectors z by a probability distribution $q_\phi(z|\pi)$. We regularize the posterior to match a prior $p(z)$, which we set to be a Gaussian with zero-mean and standard deviation 0.25. Diffusion processes converge toward Gaussian distributions so modeling data to approximate this distribution results in faster and more stable training. We also add a con-

stant β to control the strength of regularization, which we set to $1e$-5. We do not use a VAE reconstruction loss.

Empirically our modulation method for implicit functions is capable of representing substantially more complex and diverse geometries compared to existing methods [11, 40]. We provide comparisons in the supplement.

4.2. Diffusing Modulation Vectors

Next, we use our sampled latent vectors z from the previous step as sample space for the proposed diffusion probabilistic model Ω, illustrated in Fig. 2. In every iteration, Gaussian noise is added to the latent vectors at random timesteps, and the model learns to denoise the vectors. Instead of predicting the added noise ϵ as in the original DDPM [18], we follow Aditya *et al*. [44] and predict z_0, the original, denoised vector. In other words, after we sample a timestep t and noise ϵ to obtain z_t from input latent vector z_0, the model learns to reconstruct z_0. The loss function is

$$\mathcal{L}_{\text{diff}} = \|\Omega(z_t, \gamma(t)) - z_0\|_2, \tag{3}$$

where $\gamma(\cdot)$ is a positional embedding and $\|\cdot\|_2$ is MSE loss.

We concatenate z_t and $\gamma(t)$ as input into the model, which has layers each consisting of attention [58], a fully connected layer, and normalization. We use the architecture of DALLE-2 [44] because they use 1D vectors as input, similar to our case. In contrast, the standard DDPM [18] architecture is a UNet designed for images. During test time, our diffusion model performs generation iteratively as

$$z' = (f \circ ... \circ f)(z_T, T), \ f(x_t, t) = \Omega(x_t, \gamma(t)) + \sigma_t\epsilon, \tag{4}$$

where $z_T \sim \mathcal{N}(0, I)$, σ_t is the fixed standard deviation at the given timestep, and $\epsilon \sim \mathcal{N}(0, I)$. We iteratively denoise z_T until we obtain the final output z'. Then, we pass

the generated latent vectors z' back into the joint SDF-VAE model for marching cubes reconstruction.

4.3. Conditioning Mechanisms

One advantage of SDFs is their ability to represent 3D geometries from different modalities, such as point clouds and images [8, 57]. Given some input y, we can train a custom encoder Υ to extract shape features $\pi = \Upsilon(y)$ to guide training of the diffusion model. We primarily experiment with partial point clouds, but in Sec. 5.3, we show conditioning on real-scanned point clouds and 2D images.

We use the same architecture for our diffusion model described in the previous subsection, but add Υ and an additional cross-attention layer to each block. Our cross-attention layer is the same as that used in [46] and is defined as

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \quad (5)$$

where $Q = W_Q^{(i)} \cdot \Omega_i(z_t, \gamma(t)), K = W_K^{(i)} \cdot \pi, V = W_V^{(i)} \cdot \pi$. $\Omega_i(\cdot)$ is the output of an intermediate layer of Ω and W_Q, W_K, W_V are learnable matrices. The cross-attention mechanism learns the mapping between the conditioned input and the geometry implicitly represented by the latent code. Eq. (3) is now conditioned on π and we have

$$\mathcal{L}_{\text{c-diff}} = \|\Omega(z_t, \gamma(t)|\pi) - z_0\|_2. \quad (6)$$

Generation steps are the same as in Eq. (4) but in each step π is given as a condition

$$z' = (g \circ ... \circ g)(z_T, T, \pi), \ g(x_t, t, \pi) = \Omega(x_t, \gamma(t)|\pi) + \sigma_t \epsilon \quad (7)$$

By conditioning our diffusion model during training, we can guide reconstructions during test time. Furthermore, we can generate *multi-modal* reconstructions due to the generative nature of diffusion models. Finally, to increase diversity and prevent overfitting, we follow [19]; every training iteration, with a certain probability we use a zero-mask instead of the shape feature as condition. In practice, we use the zero-mask with probability 80%.

4.4. End-to-End Training for Geometry Constraints

Our model consists of the creation of latent vectors through jointly training a conditional SDF and a VAE, and training the diffusion model using the latent vectors as input. These two modules can be trained end-to-end. As shown by the gray arrow in Fig. 2, the output of the VAE can directly be used as input to the diffusion model, whose output can then be fed into the VAE decoder for calculating its SDF loss. In practice, we found that training end-to-end from scratch took longer than training the modules separately since there are many building blocks. After the two modules complete training, however, we fine-tune them

end-to-end. During training of the diffusion model individually, it can overfit to the input latent vectors since they are saved and fixed. When training end-to-end, the inputs are from the output of $(\Theta_{enc} \circ \Psi)(P)$ instead, which slightly vary each iteration, increasing generalization capabilities. Furthermore, the loss is originally based solely on the diffusion loss of the latent vectors, which does not have explicit geometrical constraints. By connecting the two modules, we introduce another SDF loss for the denoised latent vector, which guides SDF information to the diffusion model. This allows the model to generate more complex geometries. During this final stage of fine-tuning, we continue to use all the loss functions used in the separate modules, and add this additional constraint for end-to-end optimization

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{mod}} + \mathcal{L}_{\text{c-diff}} + \|\Phi(x|z') - \text{SDF}(x)\|_1, \quad (8)$$

where $z' = \Omega(z_t, \gamma(t)|\pi)$. We did not find it necessary to add weighing constants to the loss terms.

5. Experiments

Next, we validate the proposed method for generating shapes. In Sec. 5.1, we report results of unconditional generation initialized from Gaussian noise. In Sec. 5.2 we perform shape completion of sparse, partial point clouds, and, in this context, we compare and analyze existing related methods. In Sec. 5.3, we demonstrate different applications of our method by generating samples from real scanned, noisy point clouds, and 2D images. We end with an ablation study validating the design choices in Sec. 5.5.

Implementation We train our method as follows. First, we train our joint SDF-VAE model on full point clouds and corresponding query points and ground truth signed distance values. We combine the architecture of GenSDF [8] with a VAE [21] consisting of a 5-layer encoder and 5-layer decoder. The PointNet in GenSDF outputs three 2D plane features from the point cloud, which we concatenate and pass as input to the VAE. The bottleneck of the VAE (z in Fig. 2) is a 1D latent embedding of an SDF, and we save them after training is complete. Next, we use the latent vectors as training data for the diffusion model. We follow the architecture of DALLE-2 [44, 59]. There are six blocks each consisting of a self-attention layer and a fully-connected layer. For conditional training, we introduce custom encoders for different inputs. We use a PointNet [43] for point clouds and ResNet 18 [17] for 2D images. We also add another cross-attention layer to each block of the diffusion model that learns key and value pairs from extracted shape features. Finally, we fine-tune both modules end-to-end by connecting them as illustrated in Fig. 2. We provide a full architecture description and training details in the supplement.
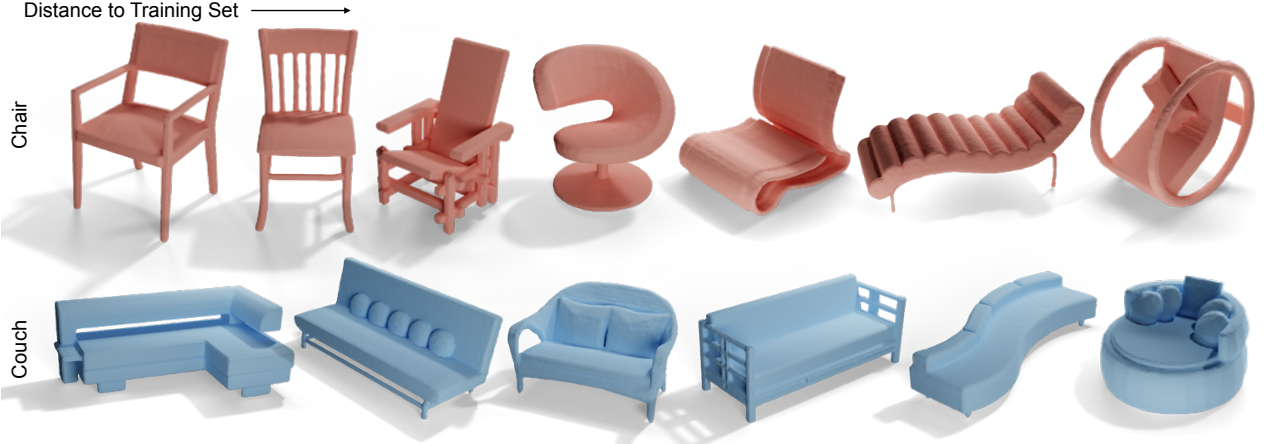
Figure 3. Samples from unconditional generation. Our method produces clean meshes with thin structures and diverse geometries. We also calculate their average CD to each object in the training set to confirm that our model is capable of producing unique shapes.

Table 1. Metrics for unconditional shape generation. ↑ means higher is better and ↓ means lower is better. MMD is scaled up by 10^2. COV and 1-NNA are measured in percentages (%).

Shape	Model	MMD (↓)	COV (↑)	1-NNA (↓)
Chair	ShapeGAN [23]	7.738	8.661	99.80
	PVD [69]	0.342	39.43	86.56
	DPM3D [28]	0.130	56.69	53.54
	Ours	**0.129**	**65.35**	**51.18**
Couch	ShapeGAN [23]	6.527	1.923	99.84
	PVD [69]	0.145	49.45	56.83
	DPM3D [28]	0.108	48.72	62.82
	Ours	**0.106**	**61.22**	**54.97**
Multi-Class	ShapeGAN [23]	4.659	5.280	99.99
	PVD [69]	0.350	12.36	93.33
	DPM3D [28]	0.150	45.40	68.36
	Ours	**0.131**	**57.06**	**67.38**

Table 2. Metrics for multi-modal shape completion of sparse, partial point clouds (128 points, 50% cropped). ↑ means higher is better and ↓ means lower is better. All values are scaled up 10^2.

Shape	Model	MMD (↓)	TMD (↑)	UHD (↓)
Chair	cGAN [62]	0.193	2.663	7.804
	PVD [69]	0.504	9.163	**3.917**
	SFormer [63]	0.278	4.820	17.76
	Ours	**0.036**	**14.22**	12.56
Couch	cGAN [62]	0.145	2.231	7.251
	PVD [69]	0.350	7.920	**6.134**
	SFormer [63]	0.103	1.567	7.270
	Ours	**0.041**	**13.53**	10.37
Multi-Class	cGAN [62]	0.225	1.994	**7.162**
	PVD [69]	0.412	10.16	8.368
	SFormer [63]	0.208	9.523	14.98
	Ours	**0.035**	**20.11**	14.86

Datasets For unconditional generation and partial point cloud completion, we train and evaluate using Acronym [12]. Acronym is a processed subset of the popular ShapeNet [6] dataset and contains watertight, synthetic 3D meshes across 262 shape categories. We use three training splits. The first two are single categories: Chair and Couch. The third split uses all classes that have at least 20 objects, providing us with 106 classes in total. From each of them, we take at most 50 objects to prevent the model from overfitting to larger categories. For single-view reconstruction, we use the Airplane and Couch categories from ShapeNet [6] and their rendered 2D images as conditioning input. For real-scanned point clouds, we use YCB [4], a collection of point clouds acquired from multi-view RGBD captures.

Evaluation For unconditional generation, we follow Yang *et al*. [64] and use minimum matching distance (MMD), coverage (COV), and 1-nearest neighbor accuracy (1-NNA). MMD measures quality, COV measures diversity,

and 1-NNA uses a classifier to measure the similarity of the reference and generated distributions, where 50% accuracy means the generated set is indistinguishable from the reference set. We generate the same number of samples as the reference set. For conditional generation (shape completion), we follow Wu *et al*. [62] and evaluate MMD, total mutual difference (TMD), and unidirectional Hausdorff distance (UHD). TMD measures diversity and UHD measures fidelity to the input partial point cloud. We generate 10 samples for every input partial point cloud in the reference set. For all metrics except UHD, we use Chamfer Distance (CD) [43] as the distance measure. We extract 2,048 points from each sample to calculate these metrics. Additionally, since generated results are random, we run evaluation 5 times and report the best set of metrics. We provide detailed formulations of all metrics in the supplement. For visualization, we run Marching Cubes [26] and render the resulting meshes.
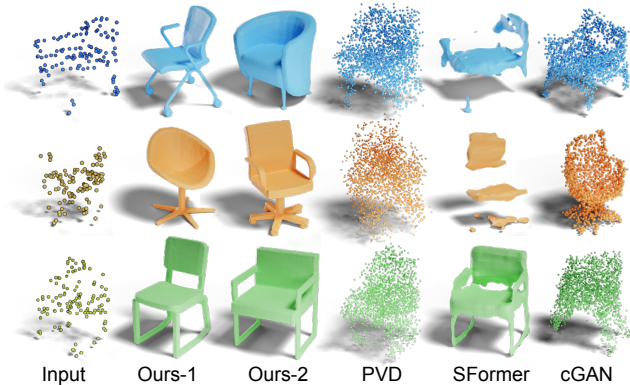
Figure 4. Shape completion results from sparse, partial point clouds. Reconstructions from the proposed method represent details such as the legs of the chair, whether they are separated (top), branched out (middle), or connected (bottom).

Figure 5. Reconstructing scanned point clouds and single images. Our method captures details of conditioned geometry, such as the curves of the drill, engines of the plane, and pillows on the couch.

5.1. Unconditional Generation

We train unconditional models on the three data splits mentioned above: Chair, Couch, and Multi-class. We compare to ShapeGAN [23], which generates SDFs, and PVD [69] and DPM3D [28], which both diffuse point clouds. Our method outperforms baselines in all metrics as reported in Tab. 1. Furthermore, our method has substantially higher diversity measured by coverage (COV), surpassing the second-best result by roughly 10% in all experiments. This is due to our regularized latent space, which allows the model to learn and interpolate from a continuous distribution. Our visualizations (Fig. 1 and Fig. 3) validate that our model generates clean 3D surfaces. We also calculate the average distance between the generations and each object in the reference set to confirm that our model is capable of producing diverse and unique shapes.

5.2. Conditional Generation for Shape Completion

Next, we assess the proposed method for shape completions of sparse, partial point clouds. During both training and testing, we randomly sample 128 points from a full point cloud, then crop 50% of points. We randomly select a viewpoint and remove the 64 furthest points from the viewpoint to obtain a partial point cloud, following Yu *et al.* [66]. We perform this cropping during training online in each iteration. We report quantitative results in Tab. 2 and show visualizations in Fig. 4. Previous works perform well on dense partial point clouds and we show results in our supplement. However, completing a sparse and partial point cloud is challenging. Methods such as ShapeFormer (SFormer) [63] and AutoSDF [35] fail under this setting because they quantize shapes into patches. Sparse point clouds mean there are very few patches to extract information from. PVD [69] produces noisy samples because they operate on discrete points and cannot interpolate smoothly

from a learned prior distribution. cGAN [62] learns a regularized latent space so its generations are relatively clean and complete but are less diverse because its priors are less expressive due to the "prior hole problem" of VAEs, which diffusion models solve [68, 56, 1, 49].

Our method outperforms all baselines in MMD (quality) and TMD (diversity) but not UHD. The UHD metric measures fidelity by finding the largest distance between any partial input point and its nearest neighbor to the completed shape, so outliers determine the UHD value. We note that PVD [69] performs well under the UHD metric but generations are noisy and less realistic. Our visualizations in Fig. 4 show that our completions match the input well. Given indication of the style of the legs of a chair, our method produces plausible shapes accordingly.

5.3. Other Modalities for Conditioning

In our method, we formulate shape completion, single-view reconstruction, and reconstruction of real-scanned point clouds as a unified task. Essentially, we are learning a distribution over 3D shapes that we can sample from, given a condition. In Fig. 1 and Fig. 5, we show two additional modalities: real scanned, noisy point clouds and 2D images. A sample image for extracting the scanned point cloud is shown for reference, but was not used during training. For the former, we train from scratch using the YCB [4] dataset, a collection of point clouds acquired from multiview RGBD captures. The point clouds are noisy and incomplete (e.g., the bottom of each object is on a table and not captured). For conditioning on 2D images, we use a pretrained ResNet 18 [17] as our encoder Υ.

5.4. Scalability and Interpolation

Large Training Datasets We investigate whether our VAE poses a bottleneck to learning large datasets. The single-category experiments (Chair, Couch) are performed on 558 and 366 meshes, respectively. Our multi-class split contains 4230 meshes. As such, without adjusting the architecture or number of parameters, our method scales without degradation in quality or creating artifacts. To further validate scalability, we experiment with 90% of the entire

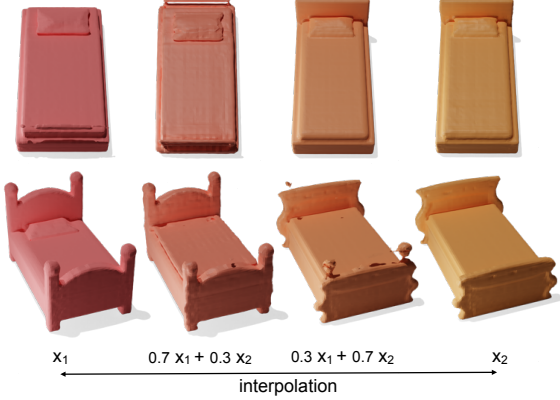x_1 $0.7\,x_1 + 0.3\,x_2$ $0.3\,x_1 + 0.7\,x_2$ x_2

interpolation

Figure 6. Interpolation of generations x_1, x_2 by performing a linear combination of their respective latent vectors. The resulting vectors are decoded and meshed.

Acronym dataset, 7148 meshes in total. The average Chamfer Distance (CD) of the SDF reconstructions of all 7148 meshes is 0.92×10^{-3}, and 0.87×10^{-3} when reconstructing only the Couch category. This value is *lower* than training on the single Couch category, where the reconstruction CD is 1.04×10^{-3}. This validates that our approach scales gracefully, learns better when we introduce more training data, and generalizes to out-of-distribution shapes as many categories have very few data (just 1-10 meshes) compared to the larger classes (300-500 meshes). See visualizations in the supplement.

Latent Interpolation In Fig. 6, we interpolate between generated samples by performing a linear combination of their respective latent vectors. Gradual shift in shape features, such as the headboards, validate that our learned latent space is continuous and the latents control semantic geometry. Thus, during generation, our model randomly samples from the latent space to form novel shapes. We provide further discussion in the supplement.

5.5. Ablation Experiments

We analyze design choices that affect conditional generations. In Tab. 3, we report shape completion metrics for the Couch training split. We also define a *consistency (CONS)* metric for measuring fidelity, following our observation in Sec. 5.2 that UHD does not correlate closely with visual results. For CONS, we evaluate all points in the input partial point cloud using the generated SDF and take the average of the predicted signed distance values. If the points are to be present on the reconstructed surface, then by definition, the values of each point are close to 0. This metric also allows us to filter generations before running marching cubes, which reduces inference time while maintaining high-quality samples. For filtering, we sample the maximum number of meshes that can fit into one sampling batch (30 in our case), and keep the 10 meshes with the lowest consistency scores. Note that the CONS filter is only used

Table 3. Ablation study based on conditional generation using the Couch split. ↑ means higher is better and ↓ means lower is better. All values are scaled up by 10^2.

	MMD (↓)	TMD (↑)	CONS (↓)
No end-to-end	0.096	8.292	5.346
$\omega = 4$	0.044	7.251	**0.822**
$\omega = 1$	0.049	11.46	1.594
Concatenation	0.049	12.22	2.068
Ours	**0.041**	13.53	1.967
Ours (no filt)	**0.041**	**17.06**	3.545

for conditional generation, and not during training.

No end-to-end refers to skipping the fine-tuning stage explained in Sec. 4.4. Generations are clean and realistic but lack diversity and complexity. This confirms our end-to-end training scheme improves generalization and introduces geometrical constraints. Next, ω refers to the ratio of guidance strength during generation: the final output is a linear combination of two generations with and without guidance: $(\omega + 1)z_c - (\omega)z_u$, where z_c is guided and z_u is unconditional [19]. In our experiments, we guide generations every iteration (i.e., $\omega = 0$) but depending on the use case, one can adjust this hyperparameter to determine the tradeoff between diversity and fidelity. See supplement for details. Then, we experiment with *Concatenation* instead of cross-attention for conditioning, following [44]. We diffuse the concatenation of the conditioned feature and noisy vector and rely on self-attention. We find both conditioning mechanisms lead to similar outputs, but concatenation increases dimensions and memory consumption significantly. Finally, we report quantitative values without filtering in *no filt*. We show that our CONS filter does not inflate MMD (quality) and TMD (diversity). We note that our model prioritizes generation quality and diversity, at the cost of the UHD metric. In future work, we plan to enforce the relationship between latent vectors of partial shapes and those of complete shapes to improve consistency.

6. Conclusion

We devise a probabilistic diffusion model that generates diverse shapes from a distribution of learned SDFs. We validate the proposed method for shape generation and completion of various input modalities. To further improve the method, we could speed up inference time of the diffusion model with techniques such as DDIM sampling [52], and we could enforce the relationship between the latents of partial shapes and those of complete shapes to improve interpretability and fidelity. There are also many avenues for exciting future work. Besides exploring other conditioning approaches, e.g., text-to-shape, we could learn appearance for generating realistic assets. We would also be interested in expanding Diffusion-SDF to full scene synthesis.

Acknowledgments

References

[1] Jyoti Aneja, Alex Schwing, Jan Kautz, and Arash Vahdat. A contrastive learning approach for training variational autoencoder priors. *Advances in neural information processing systems*, 34:480–493, 2021. 7

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 1, 2

[3] M Lo Brutto and Paola Meli. Computer vision tools for 3d modelling in archaeology. *International Journal of Heritage in the Digital Era*, 1(1_suppl):1–6, 2012. 1

[4] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 international conference on advanced robotics (ICAR)*, pages 510–517. IEEE, 2015. 1, 2, 6, 7

[5] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5799–5809, 2021. 1, 3

[6] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2, 6

[7] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6970–6981, 2020. 2

[8] Gene Chou, Ilya Chugunov, and Felix Heide. Gensdf: Two-stage learning of generalizable signed distance functions. In *Proc. of Neural Information Processing Systems (NeurIPS)*, 2022. 1, 3, 5

[9] Thomas Davies, Derek Nowrouzezahrai, and Alec Jacobson. On the effectiveness of weight-encoded neural implicit 3d shapes. *arXiv preprint arXiv:2009.09808*, 2020. 1, 3

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 1, 2

[11] Emilien Dupont, Hyunjik Kim, SM Ali Eslami, Danilo Jimenez Rezende, and Dan Rosenbaum. From data to functa: Your data point is a function and you can treat it like one. In *International Conference on Machine Learning*, pages 5694–5725. PMLR, 2022. 1, 2, 3, 4

[12] Clemens Eppner, Arsalan Mousavian, and Dieter Fox. ACRONYM: A large-scale grasp dataset based on simulation. In *2021 IEEE Int. Conf. on Robotics and Automation, ICRA*, 2020. 2, 6

[13] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2463–2471, 2017. 1

[14] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. In *Advances In Neural Information Processing Systems*, 2022. 2

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1

[16] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5, 7

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1, 2, 3, 4

[19] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 5, 8

[20] Ka-Hei Hui, Ruihui Li, Jingyu Hu, and Chi-Wing Fu. Neural wavelet-domain diffusion for 3d shape generation. *arXiv preprint arXiv:2209.08725*, 2022. 2

[21] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations*, 2014. 3, 4, 5

[22] Kilian Kleeberger, Richard Bormann, Werner Kraus, and Marco F Huber. A survey on learning-based robotic grasping. *Current Robotics Reports*, 1(4):239–249, 2020. 1

[23] Marian Kleineberg, Matthias Fey, and Frank Weichert. Adversarial generation of continuous implicit shape representations. *arXiv preprint arXiv:2002.00349*, 2020. 6, 7

[24] Liang Liang, Fanwei Kong, Caitlin Martin, Thuy Pham, Qian Wang, James Duncan, and Wei Sun. Machine learning–based 3-d geometry reconstruction and modeling of aortic valve deformation using 3-d computed tomography images. *International journal for numerical methods in biomedical engineering*, 33(5):e2827, 2017. 1

[25] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15651–15663. Curran Associates, Inc., 2020. 1

[26] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In

Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '87, page 163–169, New York, NY, USA, 1987. Association for Computing Machinery. 6

[27] Shitong Luo, Jiaqi Guan, Jianzhu Ma, and Jian Peng. A 3d generative model for structure-based drug design. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 1

[28] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2021. 2, 6, 7

[29] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Neural-pull: Learning signed distance function from point clouds by learning to pull space onto surface. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7246–7257. PMLR, 18–24 Jul 2021. 3

[30] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015. 1

[31] Ishit Mehta, Michaël Gharbi, Connelly Barnes, Eli Shechtman, Ravi Ramamoorthi, and Manmohan Chandraker. Modulated periodic activations for generalizable local functional representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14214–14223, 2021. 1, 3

[32] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[33] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 3

[34] Christopher S Milligan. Facial recognition technology, video surveillance, and privacy. *S. Cal. Interdisc. LJ*, 9:295, 1999. 1

[35] Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. Autosdf: Shape priors for 3d completion, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 306–315, 2022. 2, 7

[36] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 1, 2

[37] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. *arXiv preprint arXiv:2103.03841*, 2021. 1, 2

[38] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 1, 2

[39] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2856–2865, 2021. 1

[40] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 1, 3, 4

[41] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision*, pages 523–540. Springer, 2020. 3

[42] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 2

[43] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1, 3, 5, 6

[44] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1, 2, 4, 5, 8

[45] Yuchen Rao, Yinyu Nie, and Angela Dai. Patchcomplete: Learning multi-resolution patch priors for 3d shape completion on unseen categories. *arXiv preprint arXiv:2206.04916*, 2022. 2

[46] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1, 2, 5

[47] Alize Scheenstra, Arnout Ruifrok, and Remco C Veltkamp. A survey of 3d face recognition methods. In *International Conference on Audio-and Video-based Biometric Person Authentication*, pages 891–899. Springer, 2005. 1

[48] Ygor Rebouças Serpa and Maria Andréia Formico Rodrigues. Towards machine-learning assisted asset generation for games: a study on pixel art sprite sheets. In *2019 18th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, pages 182–191. IEEE, 2019. 1

[49] Abhishek Sinha, Jiaming Song, Chenlin Meng, and Stefano Ermon. D2c: Diffusion-decoding models for few-shot conditional generation. *Advances in Neural Information Processing Systems*, 34:12533–12548, 2021. 7

[50] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020. 3

[51] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 1, 2, 3

[52] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 8

[53] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 3

[54] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 1

[55] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11358–11367, 2021. 1, 3

[56] Jakub Tomczak and Max Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pages 1214–1223. PMLR, 2018. 7

[57] Kalyan Alwala Vasudev, Abhinav Gupta, and Shubham Tulsiani. Pre-train, self-train, distill: A simple recipe for supersizing 3d reconstruction. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 5

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[59] Phil Wang. Dalle2-pytorch, 2022. 5

[60] Xiaogang Wang, Marcelo H. Ang Jr. , and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[61] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. *arXiv*, 2022. 2

[62] Rundi Wu, Xuelin Chen, Yixin Zhuang, and Baoquan Chen. Multimodal shape completion via conditional generative adversarial networks. In *European Conference on Computer Vision*, pages 281–296. Springer, 2020. 2, 6, 7

[63] Xingguang Yan, Liqiang Lin, Niloy J Mitra, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Shapeformer: Transformer-based shape completion via sparse representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6239–6249, 2022. 2, 6, 7

[64] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4541–4550, 2019. 6

[65] Lin Yen-Chen, Pete Florence, Jonathan T. Barron, Tsung-Yi Lin, Alberto Rodriguez, and Phillip Isola. NeRF-Supervision: Learning dense object descriptors from neural radiance fields. In *IEEE Conference on Robotics and Automation (ICRA)*, 2022. 1

[66] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *ICCV*, 2021. 2, 7

[67] Wentao Yuan, Zhaoyang Lv, Tanner Schmidt, and Steven Lovegrove. Star: Self-supervised tracking and reconstruction of rigid objects in motion with neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13144–13152, 2021. 1

[68] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2, 7

[69] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5826–5835, 2021. 2, 6, 7