NeuSDFusion: A Spatial-Aware Generative Model for 3D Shape Completion, Reconstruction, and Generation

Ruikai Cui[1] *, Weizhe Liu[2] †, Weixuan Sun[2], Senbo Wang[2], Taizhang Shang[2], Yang Li[2], Xibin Song[2], Han Yan[3], Zhennan Wu[4], Shenzhou Chen[2], Hongdong Li[1], and Pan Ji[2]

[1]Australian National University [2]Tencent XR Vision Labs
[3]Shanghai Jiao Tong University [4]The University of Tokyo

Abstract. 3D shape generation aims to produce innovative 3D content adhering to specific conditions and constraints. Existing methods often decompose 3D shapes into a sequence of localized components, treating each element in isolation without considering spatial consistency. As a result, these approaches exhibit limited versatility in 3D data representation and shape generation, hindering their ability to generate highly diverse 3D shapes that comply with the specified constraints. In this paper, we introduce a novel spatial-aware 3D shape generation framework that leverages 2D plane representations for enhanced 3D shape modeling. To ensure spatial coherence and reduce memory usage, we incorporate a hybrid shape representation technique that directly learns a continuous signed distance field representation of the 3D shape using orthogonal 2D planes. Additionally, we meticulously enforce spatial correspondences across distinct planes using a transformer-based autoencoder structure, promoting the preservation of spatial relationships in the generated 3D shapes. This yields an algorithm that consistently outperforms state-of-the-art 3D shape generation methods on various tasks, including unconditional shape generation, multi-modal shape completion, single-view reconstruction, and text-to-shape synthesis. Our project page is available at `https://weizheliu.github.io/NeuSDFusion/`.

Keywords: 3D Shape Generation · 3D Object Representation

1 Introduction

In the fast-paced realm of computer vision and graphics, generative modeling has emerged as a crucial aspect of 3D shape creation, propelling advancements in 3D content generation. Recently, the remarkable achievements of generative models in 2D image synthesis and video production have inspired significant interest in

*The contribution of Ruikai Cui, Han Yan and Zhennan Wu was made during an internship at Tencent XR Vision Labs.

†Corresponding author

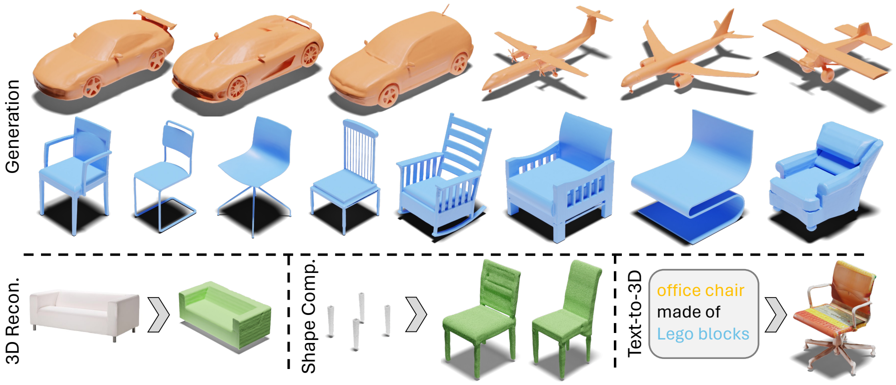


Fig. 1: NeuSDFusion demonstrates exceptional performance in generating high-quality, diverse shapes with smooth surfaces and detailed structures, showcasing its capability in 3D shape synthesis across various tasks including unconditional generation, single-view reconstruction, shape completion, and text-to-3D synthesis.

exploring generative techniques, such as diffusion models [40,41], in the creation of 3D content. This development reflects a broader transition towards more sophisticated and flexible 3D shape generation methods, fueled by the growing demand for high-quality 3D content across diverse industry applications.

Recent studies [15, 19, 36] have focused on understanding 3D geometry by leveraging multi-view supervision with predefined camera positions and incorporating inductive biases through neural rendering [31] techniques. While these approaches yield impressive results, the training process can be time-consuming and often overlooks accessible 3D data that could be harnessed to derive more effective shape priors. To directly exploit such shape priors from 3D data, contemporary methods [8, 24] typically model the distribution of 3D objects using explicit 3D representations. These techniques frequently employ Truncated Signed Distance Function (T-SDF) as the 3D object representation and utilize an encoder to embed an object directly into a latent representation. However, the T-SDF representation usually employs a 3D voxel representation to store the distance field, which is generally memory-intensive, hindering the capture of detailed shape characteristics due to memory constraints. Inspired by recent efforts [4] in using 2D plane representations for 3D shape modeling, some works [9,17] opt to encode an object as a tri-plane representation for embedding and employ point clouds as an intermediate representation. However, such an intermediate representation necessitates extra encoding networks, resulting in reduced efficiency. NFD [43], a work closely related to ours, utilizes tri-planes as the 3D object representation. However, their tri-planes are learned from occupancy grids, so that the representation capability is constrained by the grid resolution. This limitation highlights the need for more advanced 3D shape representations that can effectively capture spatial coherence and generate high-quality

3D shapes, paving the way for further progress in the field of 3D generative models.

Furthermore, while the tri-plane representation has been proposed to preserve 3D information of objects by utilizing three orthogonal 2D planes, existing approaches often overlook the 3D correlations among these planes. Previous works [9, 43] concatenate the three planes in the channel dimension and treat them as RGB images, even though there is no explicit relationship between the same coordinates on different planes. These methods neglect the synergy between planes, leading to disorder in the feature learning process, inferior shape details, and artifacts. An attempt to address cross-plane correlation is the 3D-aware convolution proposed by Rodin [47]. However, this method relies on an approximated feature aggregation through a pooling operation to tackle the computational cost, which results in displaced shape details and hampers in-plane communication.

To this end, we present NeuSDFusion, a framework for generating high-fidelity 3D objects, as depicted in Fig. 1. Unlike previous methods that either rely on rendering-based techniques, ignoring available explicit 3D priors, or employ raw 3D data (point clouds [46], mesh [26], occupancy grid [35,43], or SDF [24,32]) as 3D representations, limited by memory constraints, we introduce a hybrid tri-plane SDF representation named NeuSDF. This representation expressively embeds 3D objects as orthogonal 2D planes, preserving 3D structural information and generating smooth surfaces, while also being computationally efficient and capable of handling high-resolution details.

Our proposed approach consists of a three-stage pipeline. In the first stage, we fit each object in a dataset with a NeuSDF representation. To achieve generative modeling of this representation and retain the topological structure, we propose a specially designed autoencoder in the second stage. This autoencoder compresses the raw NeuSDF representation into a compact latent representation that preserves spatial correspondence among different planes. In the final stage, we adapt a diffusion model to synthesize these compressed latent representations of objects with various conditioning signals. The generated latent representations can then be further decoded into a new NeuSDF representation and ultimately transformed into a novel 3D object via marching cubes shape extraction [29].

We demonstrate the capability of our framework in various settings, including unconditional generation, multi-modal shape completion, single-view reconstruction, and text-guided generation. In summary, our contributions are:

- A novel pipeline capable of generating high-fidelity 3D shapes under various conditions.
- A hybrid 3D representation that captures highly detailed surface shapes with minimal memory consumption.
- A spatial-aware autoencoder structure designed to maintain spatial coherence for our proposed hybrid 3D representation.
- Extensive experiments showing that our method achieves state-of-the-art performance on various 3D shape generation benchmarks, surpassing previous methods in terms of both generation quality and diversity.

2 Related Works

2.1 Generative Modeling of 3D Shapes

In recent years, numerous works have been proposed to generate 3D shapes, with existing 3D generative models built on various frameworks. This includes generative adversarial networks (GANs) [1,48,61], variational autoencoders (VAEs) [16, 22, 32, 45], normalizing flows [54], autoregressive models [57], energy-based models [11,51], and more recently, denoising diffusion probabilistic models (DDPMs) [6, 13, 20, 49, 63]. Luo *et al.* [30] pioneered the application of DDPMs for modeling raw point clouds. Building on the success of latent diffusion models [41] in 2D image generation, several works [8, 24, 46] explored generative modeling of 3D shapes in latent space to reduce computational complexity and enhance generation quality. Within this context, LION [46] utilized point clouds to represent 3D objects, while 3DQD [24], SDFusion [8], and DiffusionSDF [9] employed the Signed Distance Function (SDF) for 3D shape representation. Concurrently, other research efforts have investigated DDPMs for 3D shape generation with varied representations, such as mesh [26], occupancy grid [43], and neural radiance fields [33]. Despite these advancements, a common limitation among most approaches is their inadvertent neglect of spatial coherence during the generation process. Addressing this gap, a recent study by Wang *et al.* [47] introduces a method that maintains spatial correspondence through the use of 3D-aware convolution on rolled-out tri-planes. Nevertheless, this technique employs axis-wise pooling to downsample plane features to a vector and only aggregates such downsampled features without enforcing in-plane communication. As a result, it lacks context information and fails to generate smooth shape surfaces. This highlights the need for more advanced methods that can effectively capture spatial coherence and generate high-quality 3D shapes.

2.2 Representation of 3D Shapes

Various 3D shape representations have been investigated to facilitate the 3D synthesis task. We categorize them into two primary classes: rendering-based and rendering-free methods. Rendering-based approaches [19,21], such as LRM [19], Zero123 [25], employ multi-view images to learn object geometry via volume rendering [31]. However, these methods overlook the available 3D objects as priors, leading to coarse shapes that often lack fine geometry details. On the other hand, rendering-free methods model the distribution of 3D objects using raw 3D representations, such as point clouds [12], mesh [3], binary occupancy [53, 58], and raw SDF [55,59,60]. These methods necessitate careful network design (*e.g.*, treating point clouds as sets [30] or handling mesh edges [26]), and may struggle to represent intricate object structures. SDFusion [8] and 3DQD [24] employ T-SDF to represent 3D objects. However, direct use 3D representation requires substantial memory, while most of the 3D space remains devoid of shape surfaces. Consequently, these 3D representation approaches are generally less effective compared to 2D plane formulations [4]. NDF [43] was the first to introduce

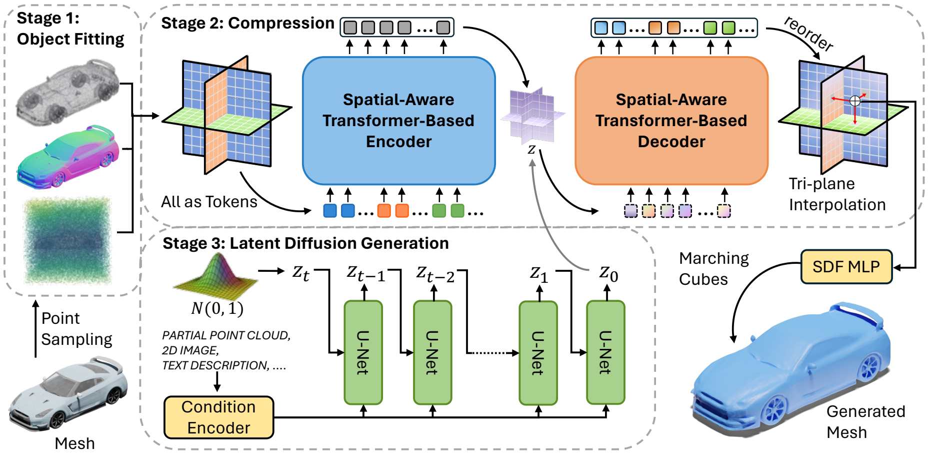


Fig. 2: Our method follows a pipeline consisting of three stages. Given a raw mesh, we first sample surface points and space-filling points to adapt each mesh to a NeuSDF representation. In the second stage, we compress the raw tri-plane representation into latent tri-planes z with a spatial-aware autoencoder. In the third stage, we train a latent diffusion model capable of generating tri-plane latent z_0 from a standard Gaussian under flexible conditions. During the inference phase, we input the generated latent z_0 into the decoder, and generate a mesh using the Marching Cubes algorithm by querying the signed distance value of any position via interpolating the reconstructed tri-plane.

a tri-plane representation for modeling raw 3D data. However, their tri-plane representation was derived from a discrete occupancy grid, constraining shape details by the resolution of planes.

3 Methodology

Our pipeline comprises three stages, as illustrated in Fig. 2. Initially, we model each 3D object with a novel hybrid representation named NeuSDF, mapping the signed distance function into three orthogonal 2D planes (Sec. 3.1).

In the second stage, we compress this 2D plane representation into latent features. To this end, we propose a transformer-based autoencoder, which efficiently encodes the tri-planes into a compact representation while preserving the 3D correlations between planes (Sec. 3.2). In the final stage, we train a diffusion model capable of performing unconditional or guided generation with prompts from various modalities, including images, text, and point clouds (Sec. 3.3).

3.1 Representing objects as NeuSDF

Formulation. A signed distance function is a function that maps a spatial coordinate (x, y, z) to a real value s, indicating the distance of the spatial position to the object surface. This function is commonly represented with a network or a vector representation [32, 34]. Instead, inspired by the recent success of the tri-plane representation [4], we represent the geometry of a 3D shape using three axis-aligned planes, *i.e.* the XY, YZ, and XZ planes. We then use a multi-layer perceptron (MLP) to decode the tri-plane into signed distance values. Formally, we query the signed distance value of any 3D position $p \in \mathbb{R}^3$ by projecting it onto each of the three feature planes, retrieving the corresponding feature vector F_{xy}, F_{xz}, F_{yz} via bilinear interpolation and aggregating the three feature vectors to obtain the feature vector for the query position. Then, we use an MLP ϕ to interpret the aggregated feature as a real value. The signed distance of any query position p is acquired as follows:

$$\varPhi(p) = \mathrm{MLP}_{\phi}(F_{xy} \oplus F_{xz} \oplus F_{yz}), \tag{1}$$

where $\oplus$ denotes element-wise summation.

Point Sampling. To effectively represent a 3D object as a NeuSDF, we sample SDF values to train the tri-planes. Given a 3D object mesh, we first preprocess it to transform it to a watertight shape using Blender's *voxel remeshing* tool. Then, we normalize the shape within the $[-1,1]^3$ box. Subsequently, we randomly sample on-surface points $\varOmega_0$ from the shape surface and uniformly sample off-surface points $\varOmega$ with ground truth SDF values in the $[-1,1]^3$ space. In addition to the signed distance supervision, we leverage normal direction as extra guidance to achieve detailed surface modeling. Consequently, we also sample the normal vector for each on-surface point.

Fitting. We employ an optimization-based approach to convert each 3D object into tri-planes, which will then be used for training our generative model. There are two learnable competent, *i.e.*, the tri-planes and the MLP with parameters ϕ. We jointly optimize both using the following geometry loss:

$$\mathcal{L}_{geo} = \mathcal{L}_{sdf} + \mathcal{L}_{normal} + \mathcal{L}_{eikonal}. \tag{2}$$

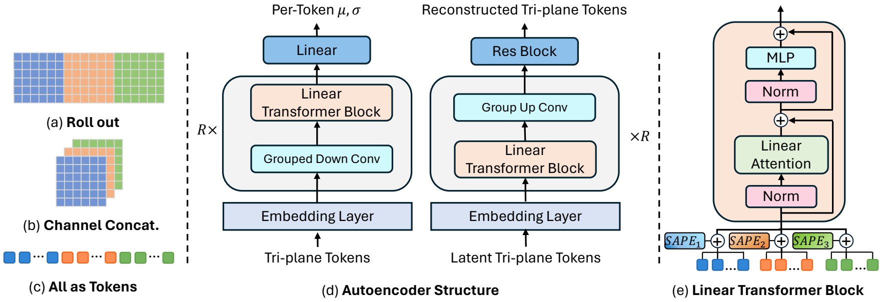
These three terms are defined as:

$$\mathcal{L}_{sdf} = \lambda_1 \sum_{p \in \varOmega_0} ||\varPhi(p)|| + \lambda_2 \sum_{p \in \varOmega} ||\varPhi(p) - d_p||, \tag{3}$$

$$\mathcal{L}_{normal} = \lambda_3 \sum_{p \in \varOmega_0} ||\nabla_p \varPhi(p) - n_p||, \tag{4}$$

$$\mathcal{L}_{eikonal} = \lambda_4 \sum_{p \in \varOmega_0} ||||\nabla_p \varPhi(p)|| - 1||, \tag{5}$$

where d_p and n_p are ground truth SDF value and surface normal vector, respectively. The gradient $\nabla_p \varPhi(p) = \left(\frac{\partial \varPhi(p)}{\partial X}, \frac{\partial \varPhi(p)}{\partial Y}, \frac{\partial \varPhi(p)}{\partial Z}\right)$ represents the direction of the steepest change in SDF. It can be computed using finite differences, *e.g.*, the partial derivative for the X-axis component reads $\frac{\partial \varPhi(p)}{\partial X} = \frac{\varPhi(p+(\delta,0,0)) - \varPhi(p-(\delta,0,0))}{2\delta}$ where δ is the step size. The Eikonal loss constrains $\|\nabla_p \varPhi(p)\|$ to be 1 almost everywhere, thus maintaining the intrinsic physical property of the signed distance function.



(a) **Roll out**

(b) **Channel Concat.**

(c) **All as Tokens**

(d) **Autoencoder Structure**

(e) **Linear Transformer Block**

Fig. 3: An illustration of the spatial-aware autoencoder design. Both (a) a roll-out mechanism and (b) a channel-wise concatenation strategy utilize a convolutional neural network to manipulate a 2D feature map, which leads to a contextual disorder. To address this issue, we propose the (c) *all as tokens* operation which is designed to preserve spatial coherence. This operation is facilitated by (d) a transformer-based autoencoder structure and the implementation of (e) a spatial-aware position embedding (SAPE) technique.

Our NeuSDF formulation learns continuous SDF values using a compact 2D plane representation, which stands in contrast to previous work. Existing methods either directly leverage T-SDF representation [8,24], which is memory-intensive and sensitive to noise, or utilize additional point cloud encoding networks [9], resulting in reduced efficiency.

3.2 Compression with Spatial-Aware Autoencoder

Given that we individually embed each object as a tri-plane, directly training a diffusion model on these tri-planes would lead to two significant drawbacks: 1) an excessive number of parameters to diffuse, resulting in the model's inability to generalize and synthesize novel tri-planes, and 2) substantial model complexity, necessitating vast computational resources [41]. To address these limitations, we introduce a spatial-aware autoencoder to compress tri-planes into latent representations. Specifically, given a tri-plane $x \in \mathbb{R}^{3\times C\times H\times W}$, the encoder $\mathcal{E}$ encodes x into a latent representation $z = \mathcal{E}(x)$, and the decoder $\mathcal{D}$ reconstructs the tri-plane from this latent representation. Thus, we have $\tilde{x} = \mathcal{D}(z) = \mathcal{D}(\mathcal{E}(x))$, where $z \in \mathbb{R}^{3\times c\times h\times w}$, and c is the feature dimension.

Model Structure. A tri-plane data structure differs from a grid-like image that can be efficiently processed via convolutional neural networks, such as the autoencoder used in Stable Diffusion [41]. We aim to model and compress the global 3D information of tri-planes into compact latent representations while preserving the 3D correlations between planes. However, most existing methods overlook these correlations. For instance, *roll out* used in Rodin [47], and [9,43] naively concatenate tri-planes either horizontally or in the channel dimension to form a 2D feature map. Such approaches result in disorder in the feature learning process and generate artifacts. Moreover, Wang *et al*. [47] attempts to address cross-plane correlation using 3D-aware convolution, but it relies on a feature aggregation using a pooling operation to tackle the computational cost and generate inferior results in our experiments. In summary, existing tri-plane autoencoders, which all adopt convolution, do not guarantee meaningful spatial correspondence among different planes.

To effectively compress the tri-planes, an efficient and expressive structure capable of handling tri-plane features while maintaining spatial consistency is essential. We propose a transformer-based autoencoder designed to process and progressively downsample tri-plane features in accordance with their 3D relationship, as depicted in Fig. 3.

Our transformer-based autoencoder adheres to a U-shaped encoder-decoder structure. Both the encoder and decoder comprise several stages, where the downsampling operation and cross-plane correlation are separately handled within each stage. This design ensures a balanced approach to feature compression and spatial comprehension. Within each stage, tri-plane features are first concatenated in the channel dimension. Subsequently, we use a group convolution to downsample each plane individually with a separate kernel, allowing for more parameter-efficient compression while preserving the spatial relationships among the tri-plane features. Following this, the tri-plane is flattened into a 1D sequence of tokens $x \in \mathbb{R}^{C \times 3HW}$ and input into a transformer block. This transformer block attends to different parts of the tri-plane representations and learn global relationships between three planes. However, a prominent limitation of the transformer structure is the computational overhead of the self-attention module, as its complexity grows quadratically with respect to the sequence length ($3 \times 64 \times 64 = 12288$ in the first stage in our setting). This complexity poses a challenge to the scalability of the transformer-based model to handle tri-planes. For example, LRM [19] only processes tri-planes with a resolution of 32, leading to unsatisfactory details. To overcome this difficulty, we utilize the linear attention mechanism proposed in [37,38]. Linear attention considerably reduces computational complexity, enabling us to directly operate on high-resolution tri-planes and achieve high-quality results.

Furthermore, naively flattening a tri-plane into a 1D sequence of tokens can result in the loss of 3D relative correlations between three planes. To preserve the cross-plane correlations in the attention operation, we propose *spatial-aware position embedding* (SAPE). Specifically, we create three orthogonal learnable position embeddings and add them to each flattened plane respectively. This approach introduces an inductive bias, allowing each token in the flattened tri-plane sequence to differentiate whether other tokens belong to the same plane or two other distinct planes. In this way, we maintain the 3D relative correlations between planes throughout the model. More details of linear attention and *spatial-aware position embedding* are introduced in the supplementary material.

Learning. Our training objective of the spatial-aware auto-encoder can be formulated as follows:

$$\mathcal{L}_{ae} = \mathcal{L}_{rec}(x, \mathcal{D}(\mathcal{E}(x))) + \mathcal{L}_{KL}(x, \mathcal{D}, \mathcal{E}) + \mathcal{L}_{geo}, \tag{6}$$

where $\mathcal{L}_{rec}$ is a L_1 norm applied between input x and its reconstruction $\mathcal{D}(\mathcal{E}(x))$. To avoid arbitrarily high-variance latent spaces, we introduce a slight KL penalty $\mathcal{L}_{KL}$ towards a standard normal on the learned latent, similar to a VAE [23]. Furthermore, we additionally add geometry loss $\mathcal{L}_{geo}$ as defined in Eq. 2 to ensure the faithful representation of the shape in the learned latent tri-plane.

To be noted, the latent tri-plane representation z retains a tri-plane structure with $z = \{z^{(i)} | z^{(i)} \in \mathbb{R}^{c \times n \times n}, i \in \{1, 2, 3\}\}$. Unlike previous work such as DiffusionSDF [9] that employs an arbitrary one-dimensional latent vector, our method upholds the inherent 3D structure of the tri-plane representation. By preserving this 3D structure, our compression model is capable of more effectively capturing the details of the input information, contributing to the generation of high-quality 3D shapes.

3.3 Generative Modeling of Latent Tri-Planes

By employing our innovative spatial-aware tri-plane autoencoder, we can now encode raw tri-planes into a compact, low-dimensional latent tri-plane space. This latent representation is more advantageous for likelihood-based generative modeling compared to the raw tri-plane space, as it enables them to focus on the fundamental, semantic aspects of the data in a reduced-dimensional, computationally more manageable space [41].

Diffusion Model Formulation. Diffusion Models are probabilistic constructs designed to learn a data distribution $q(z_0)$ by progressively denoising a normally distributed variable. The learning process is equivalent to performing the reverse operation of a fixed Markov Chain with a length of T. The diffusion process transforms latent z_0 into purely Gaussian noise $z_T \sim \mathcal{N}(0, I)$ over T time steps. The forward step in this process is defined as:

$$q(z_t | z_{t-1}) = \mathcal{N}(z_t; \sqrt{1 - \beta_t} z_{t-1}, \beta_t I), \tag{7}$$

where noisy variable z_t is derived by scaling the previous noise sample z_{t-1} with $\sqrt{1 - \beta_t}$ and adding Gaussian noise following a variance schedule $\beta_1, \beta_2, \ldots, \beta_T$.

Learning. The goal of training a diffusion model is to learn to reverse the diffusion process. To achieve this, we adopt the approach proposed by Aditya *et al.* [40], wherein a neural network is used to directly predict z_0. Given a uniformly sampled time step t from the set $\{1, ..., T\}$, we generate z_t by sampling noise from the input latent vector z_0. A time-conditioned denoising autoencoder [41], denoted by Ψ, is employed to reconstruct z_0 from z_t. The objective of the latent tri-plane diffusion is given by:

$$\mathcal{L}_{ldm} = ||\Psi(z_t, \gamma(t)) - z_0||^2, \tag{8}$$

where $\gamma(\cdot)$ represents a positional embedding and $||\cdot||^2$ denotes the mean squared error (MSE) loss.

During the testing phase, we iteratively denoise z_T to obtain the final output z'. This output can be decoded into the raw tri-plane x' with a single pass through the decoder $\mathcal{D}$. Finally, a pretrained MLP decodes x' into a dense signed distance volume using marching cube-shape extraction.

Condition Injection. To generate novel object with given conditions from various modalities, including point clouds, 2D images, and text descriptions, we inject a latent representation of conditions derived from these modalities. Specifically, for a given input y, we utilize a domain-specific encoder Υ to extract shape features $\pi = \Upsilon(y)$, which subsequently guide the sampling or training process of the diffusion model.

We employ the denoising U-Net architecture described in Rombach *et al.* [41], augmenting it with an additional cross-attention layer within each block of the U-Net backbone to incorporate condition prompts. The cross-attention layer is defined as: $\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V$, with:

$$Q = W_Q^{(i)} \cdot \psi_i(z_t, \gamma(t)), \quad K = W_K^{(i)} \cdot \pi, \quad V = W_V^{(i)} \cdot \pi. \tag{9}$$

Here, $\psi_i(\cdot)$ denotes the output of an intermediate layer of Ψ, and W_Q, W_K, W_V represent learnable parameters.

Furthermore, we adopt the classifier-free guidance paradigm [18] at each training iteration. That is, we substitute the shape feature with a *zero-mask* as a conditioning input with 10% probability to enhance sample diversity.

4 Experiments

Our NeuSDFusion offers a flexible and efficient way to generate high-quality 3D shapes, either unconditionally or guided by different conditions, as demonstrated in the following sections. We evaluate our approach in various experimental settings, encompassing unconditional content generation, multi-modal shape completion, single-view 3D reconstruction and language-guided generation.

4.1 Unconditional Generation

Adhering to the precedent established by previous work [24,46], we employ three categories from ShapeNet [5], namely Airplane, Chair and Car, to assess the unconditional generation capability of our proposed method. We adopt 1-Nearest Neighbour Accuracy (1-NNA) [28] for evaluating unconditional generation. 1-NNA is a direct measure of distributional similarity, accounting for both diversity and quality. We measure 1-NNA using both Chamfer Distance (CD) [10] and Earth Mover Distance (EMD) [14] to provide a comprehensive assessment of the evaluation metric.

As shown in Tab. 1, our method outperforms existing techniques, achieving state-of-the-art results in unconditional generation. Our approach consistently excels across all categories, demonstrating the significant superiority of our proposed pipeline in the detailed modeling of 3D shapes. Notably, we markedly surpass both LION and 3DQD, which also utilize DDPMs but with different object representations. This performance enhancement is largely attributed to our

Table 1: Results on *Airplane*, *Chair*, *Car* categories from ShapeNet using 1-NNA↓.

		Airplane		Chair		Car	
Method	Rep.	CD	EMD	CD	EMD	CD	EMD
IM-GAN [7]	Occupancy	79.48	82.94	58.59	69.05	95.69	94.79
SDF StyleGAN [61]	SDF	85.48	87.08	63.25	67.80	88.34	88.31
GET3D [15]	SDF	-	-	75.26	72.49	75.26	72.49
MeshDiffusion [26]	Mesh	66.44	76.26	53.69	57.63	81.43	87.84
LION [46]	Point Cloud	67.41	61.23	53.70	52.34	53.41	51.14
3DQD [24]	TSDF	56.29	54.78	55.61	52.94	55.75	52.80
Ours	NeuSDF	**52.33**	**52.47**	**51.95**	**52.60**	**53.06**	**51.11**

novel object representation method, NeuSDF, and the spatial-aware tri-plane autoencoding scheme. As a result, our samples are diverse and visually appealing, as illustrated in Fig. 1.

4.2 Multi-Modal Shape Completion

In accordance with standard practices [24,32], we evaluate the shape completion capability of our proposed method using the ShapeNet dataset, comprising 13 categories and follow the train/test splits provided by DISN [52]. The partial shapes encompass two settings: 1) the bottom half of the ground truth, and 2) the octant with front, left, and bottom half of the ground truth. For evaluation purposes, we compute the Total Mutual Difference (TMD) of $N = 10$ generated shapes for each input, indicative of completion diversity. We report the Minimum Matching Distance (MMD) and Average Matching Distance (AMD), which gauge the minimum and average Chamfer Distances from the ground truth to the 10 generated shapes, thereby demonstrating completion quality.

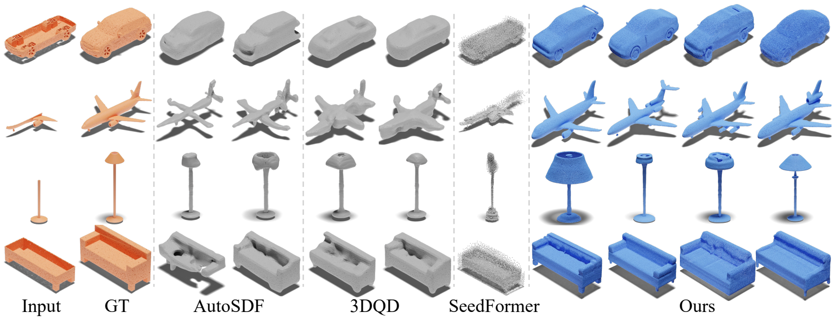


Fig. 4: Multi-modal shape completion results. Our NeuSDFusion method generates shapes with superior quality and diversity compared to previous state-of-the-art approaches, while remaining consistent with the input partial shapes.

Table 2: Results of multi-modal shape completion, metrics are multiplied by 10^2.

	Bottom Half			Octant		
Method	MMD↓	AMD↓	TMD↑	MMD↓	AMD↓	TMD↑
PoinTr [56]	5.32	N/A	N/A	21.57	N/A	N/A
SeedFormer [62]	4.97	N/A	N/A	23.99	N/A	N/A
AutoSDF [32]	3.51	8.20	4.66	5.72	12.79	8.26
3DQD [24]	2.93	6.30	**4.78**	4.69	10.93	**9.60**
Ours	**2.29**	**5.90**	4.76	**3.03**	**9.59**	8.32

We compare our approach with both state-of-the-art shape generation [24, 32] and point cloud completion [56, 62] methods. The results are presented in Tab. 2. Notably, our method outperforms the previously best-performing method, 3DQD, in terms of quality metrics for bottom half and octant shape completion, while maintaining a competitive diversity score. Qualitative results in Fig. 4 demonstrate that our method exhibits superior performance in diversity while preserving the high-quality geometries. This suggests that our proposed autoencoder effectively embeds raw tri-plane representations into a structural latent space, enabling the subsequent generation process to produce shapes with descriptive conditions.

4.3 Single-View 3D Reconstruction

Following the precedent set by previous methods [8,32], we evaluate our approach for 3D shape reconstruction from a single image using the real-world Pix3D [44] benchmark dataset. We employ the provided train/test splits for the chair category and, in the absence of official splits for other categories, follow previous methods [8, 32] to split samples into train/test sets. We adopt the official evaluation script [44] to assess our method and compare it with prior approaches. We contrast our approach with recent state-of-the-art generative single-view reconstruction methods, namely AutoSDF [32] and SDFusion [8]. The evaluation results are reported in Tab. 3. Our approach consistently outperforms prior work by a substantial margin, achieving approximately 50% improvements in both Chamfer Distance and F-Score. Visualizations of our results compared to previous work are presented in Fig. 5.

Our proposed NeuSDF can effectively represent 3D object shapes, as our pipeline is capable of modeling fine shape details with minimal memory usage, while previous methods are hindered by computational cost. Consequently, the 3D shapes generated by our approach are significantly more detailed than those produced by prior state-of-the-art methods.

4.4 Language-Guided Generation

To quantitatively evaluate our method on text-guided shape generation, we adopt the approach used in previous work [24, 32] and utilize the ShapeGlot [2]

Table 3: Quantitative results for single-view reconstruction.

Method	CD↓	F-Score↑
Pix2Vox [50]	3.00	0.39
AutoSDF [32]	2.28	0.42
SDFusion [8]	1.85	0.43
Ours	**0.92**	**0.61**

Table 4: Results for text-guided generation. PMMD, CLIP-S, and TMD are scaled by 10^2.

Method	PMMD↓	CLIP-S↑	FPD↓	TMD↑
Shape-IMLE [27]	1.68	31.42	82.34	0.54
AutoSDF [32]	1.96	31.65	141.87	1.30
3DQD [24]	1.49	32.11	59.00	2.80
Ours	**1.49**	**32.52**	**55.01**	**3.20**

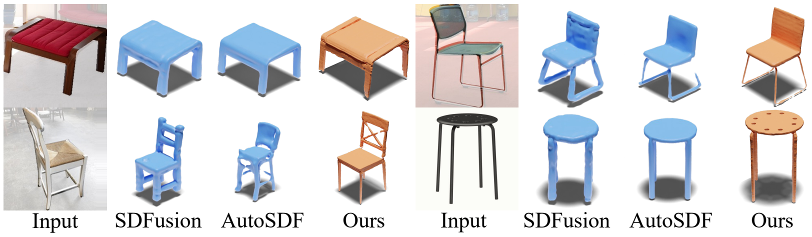


Fig. 5: Single-view reconstruction on the Pix3D dataset. Note that our approach generates significantly more detailed shapes compared to previous works, demonstrating the effectiveness of our method in capturing intricate shape properties.

dataset, which provides text utterances describing the differences between a target shape and two distractors based on the ShapeNet dataset. We employ the same train/test splits provided by AutoSDF [32]. To measure the similarity between text and shape modalities, we follow 3DQD [24] and use CLIP-S as the metric. The metric computes the maximum score of cosine similarity between $N = 9$ generated shapes and their corresponding text prompts using a pre-trained CLIP [39]. Since CLIP cannot directly process 3D shape inputs, we render each generated shape into 20 2D images from various viewpoints to compute CLIP-S. Additionally, we employ Fréchet-Pointcloud Distance (FPD) [42] and Pairwise Minimum Matching Distance (PMMD) to measure the distance between ground truth and samples.

Quantitative results in Tab. 4 demonstrate that our technique consistently outperforms the baseline works across all evaluated metrics, particularly in FPD and PMMD. This indicates that our approach adheres more consistently to text prompts than previous work. We also present text-guided generation results in Fig. 6, illustrating that our method can achieve effective modality alignment between text and 3D objects.

4.5 Ablation Study

We evaluate the effectiveness of our spatial-aware transformer-based autoencoder design by comparing it with two related methods: roll out [17, 47] and channel-

[*] https://www.meshy.ai/

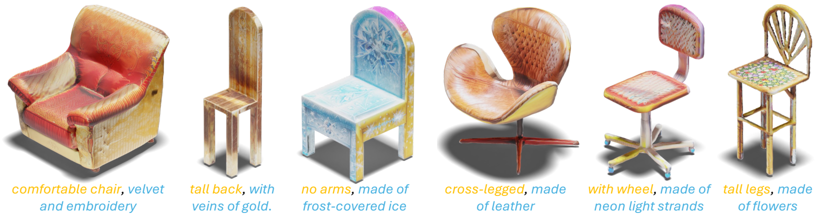

comfortable chair, velvet and embroidery
tall back, with veins of gold.
no arms, made of frost-covered ice
cross-legged, made of leather
with wheel, made of neon light strands
tall legs, made of flowers

Fig. 6: Text-guided shape generation. We showcase the ability of our method w.r.t. generating shapes based on text prompts. The orange text represents the geometry prompt and we employ Meshy*to generate texture using the blue prompt.

wise concatenation [9, 43]. Specifically, we train both our autoencoder and a CNN-based autoencoder from Stable Diffusion [41] on a chair-class subset of ShapeNet. As illustrated in Fig. 7, the reconstructed results of channel-wise concatenation or roll out exhibit defects due to spatial misalignment. The roll out method suffers from defects at the border of the tri-plane representation, as the convolution operation overlaps over two planes, while the values at the edge of two adjacent planes are not continuous in space, as shown in Fig. 7 (b). Channel concatenation, in contrast, presents some reconstruction defects due to the lack of an explicit relationship between the same spatial locations across different planes, thereby limiting the reconstruction capability. Our proposed approach employs group convolution to downsample each plane individually and introduces a specially designed attention mechanism to achieve 3D-aware tri-plane interaction. Consequently, our generated objects are integral with a smooth surface. Furthermore, we provide additional evaluation and implementation details of our method in the supplementary material.

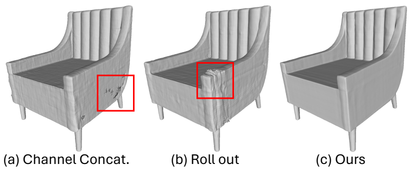


Fig. 7: Comparison of different tri-plane autoencoding methods.

5 Conclusion

In this paper, we introduce a novel approach to 3D shape synthesis by leveraging a hybrid 3D shape representation and enforcing spatial consistency within the autoencoder module. Our method enables the generation of high-fidelity 3D shapes that are consistent with various condition modalities. Through extensive evaluation, we demonstrate that our proposed approach consistently outperforms previous state-of-the-art methods across multiple evaluation metrics and diverse settings, verifying the effectiveness and robustness of our method. Furthermore, our approach exhibits superior performance in terms of generation diversity and shape quality, highlighting its potential for practical applications in computer vision, graphics, and related fields.

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds. In: International conference on machine learning. pp. 40–49. PMLR (2018)
2. Achlioptas, P., Fan, J., Hawkins, R., Goodman, N., Guibas, L.J.: Shapeglot: Learning language for shape differentiation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8938–8947 (2019)
3. Alliegro, A., Siddiqui, Y., Tommasi, T., Nießner, M.: Polydiff: Generating 3d polygonal meshes with diffusion models. arXiv preprint arXiv:2312.11417 (2023)
4. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., De Mello, S., Gallo, O., Guibas, L.J., Tremblay, J., Khamis, S., et al.: Efficient geometry-aware 3d generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16123–16133 (2022)
5. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
6. Chen, H., Gu, J., Chen, A., Tian, W., Tu, Z., Liu, L., Su, H.: Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 2416–2425 (2023)
7. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5939–5948 (2019)
8. Cheng, Y.C., Lee, H.Y., Tulyakov, S., Schwing, A.G., Gui, L.Y.: Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4456–4465 (2023)
9. Chou, G., Bahat, Y., Heide, F.: Diffusion-sdf: Conditional generative modeling of signed distance functions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2262–2272 (2023)
10. Cui, R., Qiu, S., Anwar, S., Liu, J., Xing, C., Zhang, J., Barnes, N.: P2c: Self-supervised point cloud completion from single partial clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14351–14360 (2023)
11. Cui, R., Qiu, S., Anwar, S., Zhang, J., Barnes, N.: Energy-based residual latent transport for unsupervised point cloud completion. arXiv preprint arXiv:2211.06820 (2022)
12. Cui, R., Song, X., Sun, W., Wang, S., Liu, W., Chen, S., Shang, T., Li, Y., Barnes, N., Li, H., et al.: Lam3d: Large image-point-cloud alignment model for 3d reconstruction from single image. arXiv preprint arXiv:2405.15622 (2024)
13. Erkoç, Z., Ma, F., Shan, Q., Nießner, M., Dai, A.: Hyperdiffusion: Generating implicit neural fields with weight-space diffusion. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 14300–14310 (2023)
14. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 605–613 (2017)
15. Gao, J., Shen, T., Wang, Z., Chen, W., Yin, K., Li, D., Litany, O., Gojcic, Z., Fidler, S.: Get3d: A generative model of high quality 3d textured shapes learned from images. Advances In Neural Information Processing Systems **35**, 31841–31854 (2022)

16. Gao, L., Wu, T., Yuan, Y.J., Lin, M.X., Lai, Y.K., Zhang, H.: Tm-net: Deep generative networks for textured meshes. ACM Transactions on Graphics (TOG) **40**(6), 1–15 (2021)
17. Gupta, A., Xiong, W., Nie, Y., Jones, I., Oğuz, B.: 3dgen: Triplane latent diffusion for textured mesh generation. arXiv preprint arXiv:2303.05371 (2023)
18. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)
19. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: Lrm: Large reconstruction model for single image to 3d. arXiv preprint arXiv:2311.04400 (2023)
20. Hui, K.H., Li, R., Hu, J., Fu, C.W.: Neural wavelet-domain diffusion for 3d shape generation. In: SIGGRAPH Asia 2022 Conference Papers. pp. 1–9 (2022)
21. Karnewar, A., Vedaldi, A., Novotny, D., Mitra, N.J.: Holodiffusion: Training a 3d diffusion model using 2d images. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18423–18433 (2023)
22. Kim, J., Yoo, J., Lee, J., Hong, S.: Setvae: Learning hierarchical composition for generative modeling of set-structured data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15059–15068 (2021)
23. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
24. Li, Y., Dou, Y., Chen, X., Ni, B., Sun, Y., Liu, Y., Wang, F.: Generalized deep 3d shape prior via part-discretized diffusion process. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16784–16794 (2023)
25. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9298–9309 (2023)
26. Liu, Z., Feng, Y., Black, M.J., Nowrouzezahrai, D., Paull, L., Liu, W.: Meshdiffusion: Score-based generative 3d mesh modeling. In: The Eleventh International Conference on Learning Representations (2022)
27. Liu, Z., Wang, Y., Qi, X., Fu, C.W.: Towards implicit text-guided 3d shape generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17896–17906 (2022)
28. Lopez-Paz, D., Oquab, M.: Revisiting classifier two-sample tests. In: International Conference on Learning Representations (2016)
29. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: Seminal graphics: pioneering efforts that shaped the field, pp. 347–353. ACM SIGGRAPH Computer Graphics (1998)
30. Luo, S., Hu, W.: Diffusion probabilistic models for 3d point cloud generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2837–2845 (2021)
31. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
32. Mittal, P., Cheng, Y.C., Singh, M., Tulsiani, S.: Autosdf: Shape priors for 3d completion, reconstruction and generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 306–315 (2022)
33. Müller, N., Siddiqui, Y., Porzi, L., Bulo, S.R., Kontschieder, P., Nießner, M.: Diffrf: Rendering-guided 3d radiance field diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4328–4338 (2023)

34. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: 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)
35. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A.: Convolutional occupancy networks. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16. pp. 523–540. Springer (2020)
36. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv preprint arXiv:2209.14988 (2022)
37. Qin, Z., Han, X., Sun, W., Li, D., Kong, L., Barnes, N., Zhong, Y.: The devil in linear transformer. arXiv preprint arXiv:2210.10340 (2022)
38. Qin, Z., Li, D., Sun, W., Sun, W., Shen, X., Han, X., Wei, Y., Lv, B., Yuan, F., Luo, X., et al.: Scaling transnormer to 175 billion parameters. arXiv preprint arXiv:2307.14995 (2023)
39. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
40. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 **1**(2), 3 (2022)
41. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
42. Shu, D.W., Park, S.W., Kwon, J.: 3d point cloud generative adversarial network based on tree structured graph convolutions. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3859–3868 (2019)
43. Shue, J.R., Chan, E.R., Po, R., Ankner, Z., Wu, J., Wetzstein, G.: 3d neural field generation using triplane diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20875–20886 (2023)
44. Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J.B., Freeman, W.T.: Pix3d: Dataset and methods for single-image 3d shape modeling. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2974–2983 (2018)
45. Tan, Q., Gao, L., Lai, Y.K., Xia, S.: Variational autoencoders for deforming 3d mesh models. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5841–5850 (2018)
46. Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., Kreis, K., et al.: Lion: Latent point diffusion models for 3d shape generation. Advances in Neural Information Processing Systems **35**, 10021–10039 (2022)
47. Wang, T., Zhang, B., Zhang, T., Gu, S., Bao, J., Baltrusaitis, T., Shen, J., Chen, D., Wen, F., Chen, Q., et al.: Rodin: A generative model for sculpting 3d digital avatars using diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4563–4573 (2023)
48. Wu, J., Zhang, C., Xue, T., Freeman, B., Tenenbaum, J.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. Advances in neural information processing systems **29** (2016)
49. Wu, Z., Li, Y., Yan, H., Shang, T., Sun, W., Wang, S., Cui, R., Liu, W., Sato, H., Li, H., et al.: Blockfusion: Expandable 3d scene generation using latent tri-plane extrapolation. arXiv preprint arXiv:2401.17053 (2024)

50. Xie, H., Yao, H., Sun, X., Zhou, S., Zhang, S.: Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 2690–2698 (2019)
51. Xie, J., Xu, Y., Zheng, Z., Zhu, S.C., Wu, Y.N.: Generative pointnet: Deep energy-based learning on unordered point sets for 3d generation, reconstruction and classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14976–14985 (2021)
52. Xu, Q., Wang, W., Ceylan, D., Mech, R., Neumann, U.: Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. Advances in neural information processing systems **32** (2019)
53. Yan, X., Lin, L., Mitra, N.J., Lischinski, D., Cohen-Or, D., Huang, H.: Shapeformer: Transformer-based shape completion via sparse representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6239–6249 (2022)
54. Yang, G., Huang, X., Hao, Z., Liu, M.Y., Belongie, S., Hariharan, B.: Pointflow: 3d point cloud generation with continuous normalizing flows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4541–4550 (2019)
55. Yariv, L., Puny, O., Gafni, O., Lipman, Y.: Mosaic-sdf for 3d generative models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4630–4639 (2024)
56. Yu, X., Rao, Y., Wang, Z., Liu, Z., Lu, J., Zhou, J.: Pointr: Diverse point cloud completion with geometry-aware transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 12498–12507 (2021)
57. Zhang, B., Nießner, M., Wonka, P.: 3dilg: Irregular latent grids for 3d generative modeling. Advances in Neural Information Processing Systems **35**, 21871–21885 (2022)
58. Zhang, B., Tang, J., Niessner, M., Wonka, P.: 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. ACM Transactions on Graphics (TOG) **42**(4), 1–16 (2023)
59. Zhang, B., Wonka, P.: Functional diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4723–4732 (2024)
60. Zheng, X.Y., Pan, H., Wang, P.S., Tong, X., Liu, Y., Shum, H.Y.: Locally attentional sdf diffusion for controllable 3d shape generation. ACM Transactions on Graphics (ToG) **42**(4), 1–13 (2023)
61. Zheng, X., Liu, Y., Wang, P., Tong, X.: Sdf-stylegan: Implicit sdf-based stylegan for 3d shape generation. In: Computer Graphics Forum. vol. 41, pp. 52–63. Wiley Online Library (2022)
62. Zhou, H., Cao, Y., Chu, W., Zhu, J., Lu, T., Tai, Y., Wang, C.: Seedformer: Patch seeds based point cloud completion with upsample transformer. In: European conference on computer vision. pp. 416–432. Springer (2022)
63. Zhou, L., Du, Y., Wu, J.: 3d shape generation and completion through point-voxel diffusion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5826–5835 (2021)