

1 FullFormer: Generating Shapes Inside Shapes

2 GCPR 2023 - Fast Review Track

3 Paper ID 111

4 **Abstract.** Implicit generative models have gained significant popularity for mod-
5 eling 3D data and have recently proven to be successful in generating high-quality
6 3D shapes. However, existing research predominantly concentrates on generating
7 outer shells of 3D shapes, ignoring the representation of internal details. In this
8 work, we alleviate this limitation by presenting an implicit generative model that
9 facilitates the generation of complex 3D shapes with rich internal geometric de-
10 tails. Our proposed model utilizes unsigned distance fields, enabling the represen-
11 tation of nested 3D shapes by learning from watertight and non-watertight data.
12 Furthermore, We employ a transformer-based auto-regressive model for shape
13 generation that leverages context-rich tokens from vector quantized shape em-
14 beddings. The generated tokens are decoded into unsigned distance field values
15 which further render into novel 3D shapes exhibiting intrinsic details. We demon-
16 strate that our model achieves state-of-the-art point cloud generation results on
17 the popular ShapeNet classes 'Cars', 'Planes', and 'Chairs'. Further, we curate a
18 dataset that exclusively comprises shapes with realistic internal details from the
19 'Cars' class of ShapeNet, denoted *FullCars*. This dataset allows us to demonstrate
20 our method's efficacy in generating shapes with rich internal geometry.

21 **Keywords:** Implicit Generative Models · Unsigned Distance Field.

22 1 Introduction

23 Continuous representations of data in the form of implicit functions are revolutionizing
24 many research areas of computer vision and graphics. The idea of having a continuously
25 learned implicit function to represent 3D data is efficient since these functions can rep-
26 resent diverse topologies while being agnostic to resolution [12]. Recently, neural net-
27 works have been successfully utilized to parameterize such implicit functions, leading
28 to a wide range of applications for example in geometry representation [29,1,36], image
29 super-resolution [10] or generative modeling [33,47,58].

30 Implicit representations for 3D shapes are mainly categorized into two types. The
31 first type represents the outer surface of a 3D shape as occupancy grids and the latter
32 as distance fields. Occupancy networks [29] define the surface as a continuous deci-
33 sion boundary of a deep neural network classifier whereas DeepSDF [36] represents a
34 3D surface using a signed distance field (SDF). A significant benefit of using SDF is
35 its easy extraction of the surface using the marching cubes algorithm [26]. However,
36 many implicit neural networks based on SDF or Occupancy fields require 3D shapes
37 to be watertight which are often not readily available. Atzmon et al. [1] propose a sign
38 agnostic loss function to learn an SDF from non-watertight data; however, their model
39 requires careful initialization of the neural network parameters and often misses thin

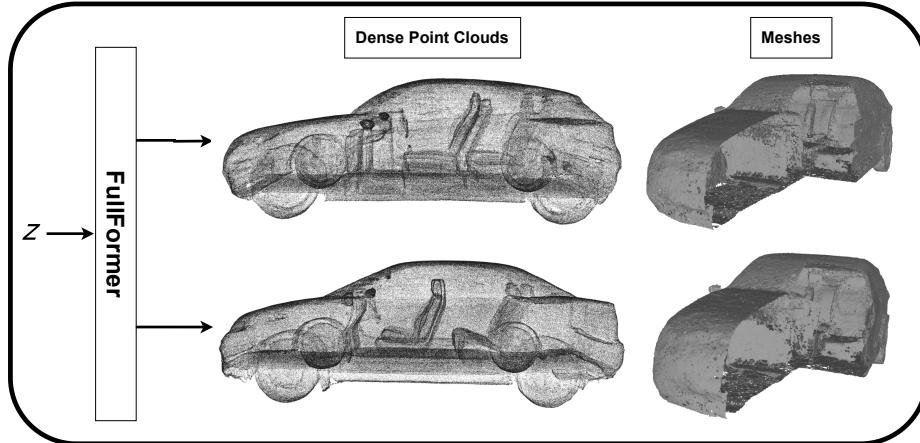


Fig. 1: This paper addresses generating 3D objects with rich internal geometric details.

40 structures. Another drawback of SDFs stems from their inherent nature, i.e., 3D shapes
 41 are modeled as inside and outside. The recently proposed works 3PSDF [8], NeAT [28]
 42 introduce the ‘null’ sign along with conventional in and out labels of SDF. This addition
 43 enables the representation of both watertight and open surfaces. However, this approach
 44 needs denser sampling in order to insert a null layer in between the multi-layer surfaces
 45 to prevent surface artifacts.

46 A simpler implicit representation for complex, potentially non-watertight surfaces
 47 can be given by unsigned distance fields (UDFs). In UDFs, the 3D shape is delineated
 48 through a regressive function that predicts the unsigned distance of a given point in
 49 space to the nearest surface of the 3D shape. This representation is capable of encoding
 50 multiple layers of internal 3D structures since distance values are not limited to only
 51 capturing inside or outside. However, the signless property of UDF’s makes it difficult
 52 to extract surfaces from the implicit fields. The standard marching cubes algorithm [26]
 53 cannot be used, as finding a zero-level set by detecting the flips between inside and
 54 outside is not possible with UDFs. Chibane et al. [13] proposed algorithms to extract
 55 point clouds comprising internal geometries from UDFs. Alongside, few works further
 56 demonstrated the use of UDFs for the task of shape reconstruction [12,57]. Nonetheless,
 57 shape completion/synthesis or novel shape generation with UDFs remain unexplored. In
 58 this paper, we present an approach which leverages UDFs’ capability to represent nested
 59 3D shapes to learn and generate rich internal details of 3D shapes, while ensuring the
 60 high quality and diversity of the generated samples.

61 To facilitate the learning of complex shapes requires a suitable encoding of distant
 62 shape contexts. This is especially true when shapes with internal structures are consid-
 63 ered, local shape context is not sufficient to model long-range relationships for example
 64 between the overall height of a car and the shape or tilting of its seats (for example, the
 65 shape of seats in a sports car is quite specific). To facilitate the encoding of relation-
 66 ships at varying spatial distances, transformer-based models that leverage self-attention
 67 are the method of choice [53,14]. Transformers are proven to be effective in modeling
 68 data distributions and generating realistic samples in image generation [16], 3D shape

69 completion[55] and 3D generation tasks [31,11,59]. Unfortunately, transformers can
 70 not directly learn from UDF representations since they rely on discrete token represen-
 71 tations. Leveraging the advantages of transformers for shape generation with internal
 72 structure is therefore non-trivial. In this paper, we contribute the following:
 73

- 74 • We present an implicit neural network-based generative framework to properly
 75 learn to generate 3D shapes with internal details while modeling long-range shape
 76 dependencies in the form of a sequence. This type of shape-dependent sequencing
 77 effectively integrates transformer-based shape learning with UDFs.
 78 • Our generative model can learn from both watertight and non-watertight 3D data.
 79 Also, it is capable of generating diverse topologies, while focusing on external
 80 shapes as well as internal details.
 81 • We demonstrate that our method outperforms previous point cloud generation ap-
 82 proaches in terms of qualitative and quantitative results on different ShapeNet cat-
 83 egories as well as on the *FullCars* dataset, a dataset curated from ShapeNet 'Cars'
 83 with internal geometric details and non-watertight surfaces.

84 2 Related Work

85 *Generative Adversarial Networks* A standard generative model used in computer vision
 86 applications is the generative adversarial network (GAN)[17]. Recent works [10,24]
 87 have shown 3D shape generation combining implicit neural networks and generative
 88 adversarial networks. However, the quality of output suffers from mode collapse and
 89 catastrophic forgetting due to the instability of GAN training [25,50].

90 *Score-based Models* Another form of generative models is denoising diffusion proba-
 91 bilistic models, also known as score matching models [22,20,49]. The main principle
 92 of these models is that they model the gradient of the log probability density function
 93 with respect to the real sample. Diffusion models have achieved state-of-the-art in many
 94 downstream tasks such as super-resolution, and data generation [45,3,6,58].

95 *Likelihood-based Models* Variational autoencoders (VAEs) and auto-regressive models
 96 (ARs) are two commonly used likelihood-based models. Both aim to learn a probability
 97 distribution over the input data. While VAEs are fast at inference time, their generation
 98 quality is often inferior compared to that of GANs[23,44]. Conversely, auto-regressive
 99 models (ARs) can represent data distribution with high fidelity but generate samples
 100 slowly [35,43,38,5]. To overcome the limitations of these two model types, hybrid
 101 models combining auto-regressive transformer models and vector quantized VAEs have
 102 been proposed [16,55,31,59,11]. Our proposed method builds upon this hybrid model
 103 setup and focuses on generating 3D shapes with internal structures. Our generation ap-
 104 proach is related to previous works like ShapeFormer[55] and Pointcloud VQVAE [11].
 105 ShapeFormer[55] utilizes a latent transformer architecture to learn from compact and
 106 discretely encoded sequences that approximate 3D shapes, specifically for 3D shape
 107 completion utilizing occupancy fields. However, ShapeFormer does not address the task
 108 of unconditional shape generation and works on only watertight data. Moreover, they

109 also employ a local pooled PointNet model [42] for feature extraction, which can limit
 110 the expressiveness of the feature embeddings. Conversely, Pointcloud VQVAE[11] uses
 111 a learned canonical space to align semantically similar point cloud categories into se-
 112 quences and employ a latent transformer model similar to ShapeFormer to learn these
 113 point cloud sequences. However, this method is restricted to point-cloud generation with
 114 a fixed number of points and lacks an implicit representation of 3D shapes, limiting their
 115 ability to generate arbitrary resolution shapes or shapes with internal structures. In con-
 116 trast, our method utilizes implicit representation of 3D shapes along with incorporating
 117 locality inductive biases, as in CNNs, in extracted features that allow for tractable fea-
 118 ture embeddings. Therefore, we opt for using an IF-Net-based [7] encoder. Also due to
 119 our representation of 3D shapes using UDF, our method offers the ability to generate
 120 novel shapes with internal structures and is not constrained by watertight-only models.

121 *Implicit Neural Generative Models* In recent years, neural implicit networks have
 122 gained significant attention for their efficacy in 3D representational learning, as for
 123 example in [37,30,1,41,48,46,61,21,19,8,57]. While several models have explored im-
 124 plicit representation for 3D surface reconstruction, only a few have used it for 3D model
 125 generation [58,19,59,31]. In general, these works rely on a type of neural representation
 126 that encapsulates a 3D surface by taking a spatial coordinate value as input and outputs
 127 a parameter, ones or zeros for points inside or outside the surface [30] or a signed dis-
 128 tance from the surface [37]. However, as mentioned before, these representations do
 129 not preserve the multi-layer geometry of 3D shapes. Recently, NDF [13] and GIFS [57]
 130 have demonstrated that UDFs are capable of representing inner details within 3D mod-
 131 els. Despite its advantages in representation power, learning UDF is more challenging
 132 than learning SDFs. UDF prediction is a regression problem while SDF and occupancy
 133 field prediction are usually cast as classification problems. This makes the training us-
 134 ing UDFs more difficult, requiring more sophisticated regression algorithms. Replacing
 135 SDF with UDF is not expected to work right away. Additionally, due to the lack of sign
 136 in the UDF representation, the model requires a sign-agnostic loss function along with
 137 careful initialization of neural network parameters and is, therefore, harder to learn than
 138 SDF [1,8]. In this paper, we propose a deep implicit generative framework that utilizes
 139 UDFs to generate high-quality 3D models with internal geometric structures. Our work
 140 highlights the potential of UDFs in generating rich 3D models. This has significant im-
 141 plications for various applications, such as product design, robotics, CAD designs, and
 142 medical imaging, whereby internal geometries are crucial for accurate modeling and
 143 simulation.

144 3 Method

145 The objective of this work is to leverage the representational power of unsigned distance
 146 fields (UDF) in order to implicitly model 3D shapes while retaining their internal geo-
 147 metric details. To achieve this goal, we utilize the learning capabilities of transformers
 148 and incorporate UDF-based implicit function learning to develop an autoregressive gen-
 149 erative model capable of generating 3D shapes with internal structures. However, the
 150 complexity of the auto-regressive generation model increases considerably with the in-
 151 put sequence length [53]. This problem is exasperated when the data representation is a

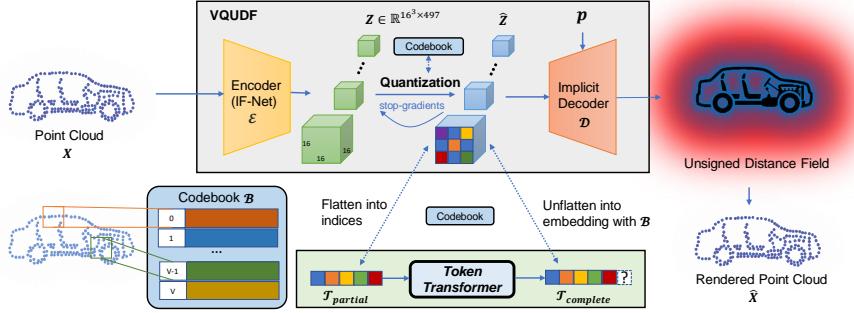


Fig. 2: Approach: Key ingredients of our pipeline are vector quantized autoencoder, unsigned distance field (UDF), and latent transformer. The first stage is learning VQUDF which is a vector quantized autoencoder model that takes voxelized point clouds as input to a CNN-based encoder and utilizes an implicit decoder to output a UDF of the 3D shape. UDF ensures rich internal details are retained in a continuous data representation. Latent codes from the learned VQUDF are used to train an autoregressive transformer. This transformer learns to generate novel latent codes at test time. An implicit decoder then decodes generated latent codes to output a UDF. A 3D shape is then rendered from the UDF as a more tractable data format such as a point cloud.

152 dense 3D model. Therefore, instead of representing a 3D model as voxels, point clouds,
 153 or discrete patches directly, we learn a compact and discrete representation whereby a
 154 shape is encoded using a codebook of context-rich parts. This allows an auto-regressive
 155 transformer model to capture long-range interactions between these contextual parts
 156 and effectively model the distributions over the full shapes. Figure 2 details the com-
 157 plete framework of our approach. Our method can be sectioned into two parts. First,
 158 we describe a form of an autoencoder, namely Vector Quantized Unsigned Distance
 159 Field (VQUDF), which learns a context-rich codebook, as detailed in Sec. 3.1. Then we
 160 present the latent transformer architecture as a generative model capable of producing
 161 novel shapes, as outlined in Sec. 3.2.

162 3.1 Sequential Encoding with VQUDF

163 A 3D shape is represented as a point cloud input denoted by $\mathbf{X} \in \mathbb{R}^{N \times 3}$. To harness
 164 the power of transformers in the generation, we encode \mathbf{X} into a discrete *sequence* of
 165 tokens. This discrete *sequence* must encapsulate the complete geometric information of
 166 the 3D shape. Inspired by ideas from [52,55,31], we formalize the encoder, codebook,
 167 and decoder architecture for generating 3D shapes with internal geometry using UDFs.

168 *Encoder:* To generate 3D shapes with internal structures using transformers, we require
 169 a compact and discrete representation of the input shape that maintains high geometric
 170 resolution. The input to our encoder is a sparse voxelized point cloud defining a 3D
 171 shape. When dealing with voxel data representations, capturing local spatial context
 172 is essential since the correlation between neighboring voxels significantly impacts the

152
153
154
155
156
157
158
159
160
161

162
163
164
165
166
167
168
169
170
171
172

173 overall shape of the object. CNNs are well-suited for capturing prior inductive bias
 174 of strong spatial locality within the images [15]. By incorporating local priors from
 175 CNNs, we can effectively capture the spatial context of the input data and encode it into
 176 a compact feature grid utilizing ideas from neural discrete representation learning [52].
 177 To achieve this, the first step is to employ a CNN-based feature extractor \mathcal{E} called IF-Net
 178 [13]. IF-Net takes a sparse voxelized point cloud \mathbf{X} and maps it to a set of *multi-scale*
 179 grid of deep features $\mathbf{F}_1, \dots, \mathbf{F}_m$ s.t. $\mathbf{F}_k \in \mathcal{F}_k^{K^3}$ and $\mathcal{F}_k \in \mathbb{R}^c$. Note that the resolution
 180 K reduces, and the number of channels c increases as k increases. For tractability, we
 181 interpolate feature grids $\mathbf{F}_1, \dots, \mathbf{F}_{m-1}$ to the scale of final feature grid \mathbf{F}_m using trilinear
 182 interpolation. This provides us with a good trade-off between model complexity and
 183 shape details. A concatenation of $\mathbf{F}_1, \dots, \mathbf{F}_m$ along the channel dimension results in a
 184 compact feature grid $\mathbf{Z} \in \mathbb{R}^{K^3 \times C}$, i.e. \mathbf{Z} is a continuous latent feature representation.
 185

185 *Quantization:* A discrete description of the world can aid learning by compressing in-
 186 formation in many domains, such as language or images [52,32,9]. We posit that 3D
 187 models are no exception and can greatly benefit from discrete representations. In addi-
 188 tion, to utilize the generative transformer model, the input shape is preferably a discrete
 189 *sequence*. Therefore, we employ vector quantization to transform the continuous la-
 190 tent feature representation \mathbf{Z} into a sequence of tokens \mathcal{T} using a learned codebook
 191 \mathcal{B} of context-rich codes $\mathcal{B} = \{\mathbf{b}_i\}_{i=1}^V \subset \mathbb{R}^{n_z}$ where n_z is the length $K \times C$ of a
 192 code. Following a row-major ordering [16], each feature slice $\mathbf{z}_i \in \mathbf{Z}$ is clamped to the
 193 nearest code in the codebook \mathcal{B} using equation 1, fig. 2, which results in a quantized
 194 feature grid $\hat{\mathbf{Z}}$. A sequence of tokens \mathcal{T} is then defined as the ordered set of indices
 195 $(t_i) \forall i \in \{1, \dots, |\mathcal{T}|\}$.

$$t_i = \operatorname{argmin}_{j \in \{1, \dots, V\}} \|\mathbf{z}_i - \mathbf{b}_j\| \quad (1)$$

196 *Decoder:* As stated earlier, we aspire to learn an implicit representation of shapes to
 197 benefit from properties of such models, for example, no watertight shape restrictions,
 198 arbitrary resolution, and encoding internal structures. To achieve this, we train a decoder
 199 to output an unsigned distance field $\text{UDF}(\mathbf{p}, \mathcal{S}) = \min_{\mathbf{q} \in \mathcal{S}} \|\mathbf{p} - \mathbf{q}\|$ which is a function
 200 that approximates the unsigned distances between the sample points \mathbf{p} and the surface of
 201 the shape \mathcal{S} . Formally, the decoder is defined as a neural function $\mathcal{D}(\hat{\mathbf{Z}}, \mathbf{p}) : \mathbb{R}^{K^3 \times C} \times$
 202 $\mathbb{R}^3 \mapsto \mathbb{R}^+$ that regresses the UDF from a set of point \mathbf{p} conditioned on the latent discrete
 203 feature grid $\hat{\mathbf{Z}}$. The dense point cloud algorithm provided by Chibane et al. [12] is used
 204 further to convert UDF to a final point cloud denoted by $\hat{\mathbf{X}}$.

205 *Training VQUDF:* The training process involves learning the encoder \mathcal{E} , codebook \mathcal{B} ,
 206 and the decoder \mathcal{D} simultaneously. The overall loss function is denoted in equation (2).
 207

$$\mathcal{L}_{\text{VQUDF}}(\mathcal{E}, \mathcal{B}, \mathcal{D}) = \|\text{UDF}(\mathbf{p}, \mathcal{S}) - \text{UDF}_{gt}(\mathbf{p}, \mathcal{S})\|_2^2 + \mathcal{L}_c \quad (2)$$

208 The first term denotes the reconstruction loss, which is computed as the difference be-
 209 tween predicted and ground truth UDFs. This method is different from the commonly
 210 utilized approach of computing loss between predicted and true point clouds. The sec-
 211 ond term \mathcal{L}_c denotes the commitment loss in equation (3).
 212

$$\mathcal{L}_c = \|\operatorname{sg}[\mathcal{E}(\mathbf{X})] - \hat{\mathbf{Z}}\|_2^2 + \|\operatorname{sg}[\hat{\mathbf{Z}}] - \mathcal{E}(\mathbf{X})\|_2^2 \quad (3)$$

209 Different from vanilla NDF training, our pipeline has a non-differentiable quantization
 210 operation. Following previous works [2,52], we utilize a straight-through gradient es-
 211 timator to circumvent this problem. Under this approach, gradients are simply copied
 212 over from the decoder to the encoder. This method ensures joint training of the code-
 213 book, the encoder, and the decoder.

214 3.2 Generating a Sequence of Latent Vectors

215 *Latent Transformer:* Transformers have shown tremendous performance in generat-
 216 ing images by modeling them as a sequence of tokens and learning to generate such
 217 sequences [39,34]. Transformers are unconstrained by the locality bias of CNNs al-
 218 lowing them to capture long-range dependencies in images. 3D models with internal
 219 structures also exhibit long-range dependencies, for example, the number and shape
 220 of seats in a car depend on the body being either a sedan or a sports car. Previous
 221 works [60,18,54,55,31,11] have successfully demonstrated capturing these depen-
 222 dencies using transformers for 3D models. We represent 3D shapes as a sequence of to-
 223 kens $\mathcal{T} = (t_1, \dots, t_{|\mathcal{T}|})$ resulting from our trained VQUDF framework. Recall that
 224 each token t_i is an index of the closest codebook latent embedding to the continu-
 225 ous latent feature grid. The generation of shapes is modeled as an autoregressive pre-
 226 diction of these indices. A transformer learns to predict the distribution of the next
 227 indices given prior ones. The likelihood of the complete sequence \mathcal{T} is described as
 228 $p(\mathcal{T}) = \prod_{i=1}^{|\mathcal{T}|} p(t_i | t_{1 \dots i-1})$.

Transformer Training: The generation of latent codes as a sequence of tokens using
 transformers is highlighted in Fig. 2. The learned weights of the trained VQUDF au-
 toencoder are frozen before the training of the transformer. VQUDF is first used to
 create a training dataset of 3D shape latent embeddings. These latent embeddings are
 used in the training of the transformer. The training objective for generation is maxi-
 mizing the log-likelihood of tokens in a randomly sampled sequence to represent the
 3D shape $p(\mathcal{T})$:

$$\mathcal{L}_{\text{Transformer}} = \mathbb{E}_{x \sim p(x)}[-\log p(\mathcal{T})] \quad (4)$$

229 After training, this model starts with the [START] token and predicts the next indices
 230 forming a complete sequence \mathcal{T} until a [END] token is predicted. By mapping indices
 231 in the sequence \mathcal{T} back to the corresponding codebook entries, a discrete latent feature
 232 grid $\hat{\mathbf{Z}}$ is recovered. The 3D shape is then reconstructed using the implicit decoder \mathcal{D} ,
 233 which results in a UDF from which point cloud $\hat{\mathbf{X}}$ is extracted as in [13].

234 4 Experiments

235 This section thoroughly evaluates our proposed approach on the standard object cate-
 236 gories of *Cars*, *Planes*, and *Chairs* from ShapeNetCore [4] dataset. Additionally, we
 237 curate a new dataset named 'Full Cars', which constitutes a subset of the *Cars* cate-
 238 gory of the ShapeNetCore v2 dataset, on which we evaluate our proposed approach

239 and competing methods on their ability to generate shapes with internal structures. Our
 240 experiments demonstrate our methods' effectiveness in generating high-quality shapes
 241 with internal structures. We compare our point cloud generation results against multi-
 242 ple SOTA point cloud generation baselines and show good qualitative and quantitative
 243 results on shape generation. More qualitative results as well as an ablation on the use of
 244 UDF versus SDF in our approach are given in the supplementary material.

245 4.1 Implementation Details

246 We train our models in two stages. First, we train the VQUDF module, followed by a la-
 247 tent transformer module. For training, we utilize stock hardware comprising one Nvidia
 248 RTX Quadro GPU with 48GB of VRAM. All code is written in PyTorch [40] whereby
 249 a portion is acquired from open repositories of [13,16]. For training both modules, we
 250 use a batch size of 1 and the Adam optimizer. For VQUDF training, we employ a learn-
 251 ing rate of 1e-6 and ReLU activation, whereas the transformer's training uses a learning
 252 rate of 4.5e-6. Furthermore, the transformer has 12 layers and 8 attention heads. The
 253 length of the input sequence to the transformer model is set as 7952; the codebook size
 254 is 8192, with each codebook having a dimensionality of 512.

255 *Datasets* We conduct experiments on the standard object categories of *Cars*, *Planes*,
 256 and *Chairs* from ShapeNetCore [4] dataset. Additionally, we curate a new dataset named
 257 'Full Cars', which constitutes a subset of the *Cars* category of the ShapeNetCore v2
 258 dataset. The 'Full Cars' dataset includes cars with diverse and realistic internal geome-
 259 try such as seats, steering wheels, shift sticks, and other internal structures. The primary
 260 objective of this curation of the dataset is to demonstrate the capability of our model
 261 in generating novel and realistic shape interiors. It is also essential to note that there
 262 is a strong interdependence between such internal structures and outer car shapes: for
 263 example, sports cars are expected to have quite specific types of seats. Further descrip-
 264 tions of datasets and additional training details, including the architecture of our model,
 265 are presented in the supplementary material.

266 4.2 VQUDF Reconstruction Performance

267 The input point cloud is sampled and voxelized before feeding into the VQUDF en-
 268 coder. The number of points sampled from different datasets and considered voxel res-
 269 olution during training of the VQUDF module are presented in Table 1. Recall that the
 270 input 3D shape is encoded into a feature grid $\hat{\mathbf{Z}}$ where each channel comprises a feature
 271 block of resolution K^3 . The quality of encoded information and generation capability
 272 depends on the dimensionality and resolution K of the 3D latent feature grid $\hat{\mathbf{Z}}$. Fig.3
 273 shows reconstruction results of the VQUDF module on the Full Cars dataset with dif-
 274 ferent values of K such that resolution of the 3D latent feature becomes $\hat{\mathbf{Z}} \in \mathbb{R}^{64^3 \times C}$,
 275 $\hat{\mathbf{Z}} \in \mathbb{R}^{16^3 \times C}$ and $\hat{\mathbf{Z}} \in \mathbb{R}^{8^3 \times C}$ respectively, where C is the number of channels. Note
 276 that the fidelity of internal geometries increases progressively with the dimensionality
 277 K of $\hat{\mathbf{Z}}$. However, increased K results in a large quantized sequence length \mathcal{T} making
 278 transformer training difficult. Hence, a good trade-off between geometrical fidelity and
 279 memory footprint is achieved by selecting $\hat{\mathbf{Z}} \in \mathbb{R}^{16^3 \times C}$ which is then processed into a
 280 tractable sequence of tokens to generate shapes with internal details.



Fig. 3: **Reconstruction Results:** Our model reconstruction results with different latent space resolutions 64^3 , 16^3 and 8^3 respectively (left to right).

Table 1: Number of points sampled and voxel resolution considered for VQUDF training for different datasets. The *Full Cars* dataset used for evaluating the ability of models to generate shapes with internal structures is curated from ShapeNet Cars by us.

Dataset	Points Sampled	Voxel resolution
ShapeNet <i>Cars</i>	10000	256^3
ShapeNet <i>Planes</i>	5000	32^3
ShapeNet <i>Chairs</i>	4000	32^3
Full Cars	10000	256^3

281 4.3 Baselines

282 We use the following baselines which generate novel 3D point clouds to compare with
 283 our point cloud generation. The first baseline is Graph Convolution GAN [51], which
 284 relies on standard GAN-based generation and employs localized operations in the form
 285 of graph convolutions to generate point clouds. Another baseline is Diffusion Model.
 286 Luo et al. [27] employs denoising diffusion probabilistic models for point cloud gener-
 287 ation. Lastly, we also compare against Pointflow [56], which utilizes normalizing flows
 288 for the point cloud generation. These models naturally carry the ability to learn inside
 289 details of 3D models, provided that they have been trained on datasets with internal
 290 structures. However, they do not utilize an implicit continuous representation to capture
 291 internal details. Therefore, these approaches are not only limited to a fixed number of
 292 points generation but also their ability to model insides in predicted 3D shapes.

293 4.4 Metrics

294 For quantitative evaluation, we use three different metrics following previous works.

295 *MMD*: Minimum matching distance (MMD) indicates the faithfulness of generated
 296 samples with real data. A lower MMD indicates that generated samples are realistic
 297 towards ground truth samples

298 *COV*: Diversity is an important aspect of generative models. A high coverage score
 299 (*COV*) indicates that the model does not suffer from mode collapse and has high sample
 300 diversity.

301 *JSD*: Jenson-Shannon divergence (JSD) computes the symmetric similarity between
 302 distributions of generated samples and reference samples. A lower value of JSD is de-
 303 sirable. However, this metric is dependent on the selection of the reference set.

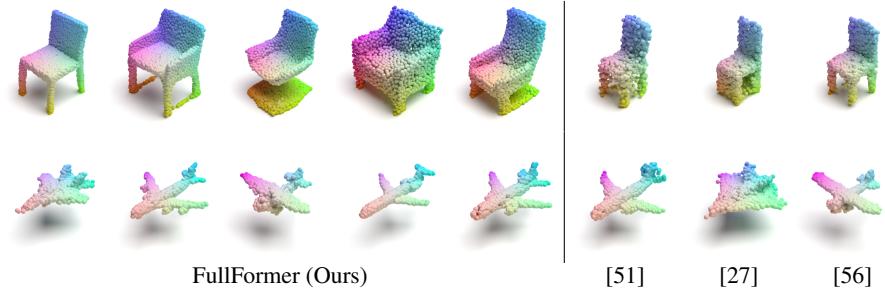


Fig. 4: Outer Hull Generation: Our models show high-quality point cloud generation results when trained on object categories of chairs, aeroplanes of ShapeNet dataset and visually improve over previous methods such as GraphCNN-GAN [51], Diffusion [27] and PointFlow [56].

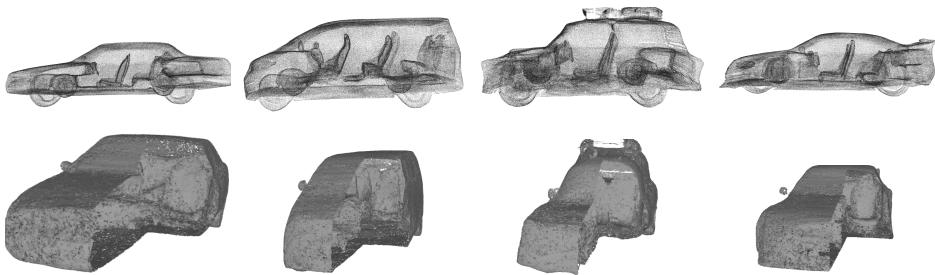


Fig. 5: Generation: Diverse generation results from our FullFormer model on the Full Cars dataset with internal structures. The high degree of detail of generated shapes is clearly visible in the dense point clouds. Note that, not only seats specific to car type, but also minute details such as steering wheels are generated. High point clouds quality even allows to compute surface meshes (bottom) of the non-watertight shapes with internal structures.

304 4.5 Qualitative Results 304

305 In this section, we show the qualitative performance of our generative model on the 305
306 considered datasets. 306

307 *ShapeNet*: The samples of point cloud generation results with 2048 points of our model 307
308 against baseline models for the classes *chairs* and *airplanes* are presented in Fig. 4. We 308
309 highlight that our model does not rely on any priors in the form of preset tokens in the 309
310 input sequence, thus ensuring the complete unconditioned generation of the results. The 310
311 performance of our method is apparent with less noisy and realistic shape generations. 311
312 We further note that immense diversity is present in the shapes generated, whereby all 312
313 generated samples in Fig. 4 are of distinct visual designs. High fidelity is also percep- 313
314 tible across the generated examples. More results of generated mesh samples of *Planes* 314
315 and *Chairs* are provided in the supplementary material. 315

316 *Full Cars:* We use the Full Cars dataset to showcase the veracity of our approach’s key
 317 feature to generate high-fidelity outer shells with intricate internal geometric details.
 318 The qualitative results of randomly generated cars are presented in Fig. 5 demonstrating
 319 the efficacy of our model in generating samples with rich internal geometric structures.
 320 Additionally, generated cars in Fig. 5 demonstrate a remarkable level of diversity, for
 321 example, varied genres of cars with different numbers of seats. We also present in Fig.
 322 6 the comparative point cloud generation results with uniformly sampled 2048 points
 323 of randomly generated cars from Diffusion [27], PointFlow [56] and our FullFormer.
 324 We retrain other comparative methods on the ‘Full Cars’ dataset by processing input
 325 data as required for the methods. Our approach achieves a clear visual superiority over
 326 comparative methods, which fail to generate any discernible internal structures. It is
 327 also important to note that shapes in the training data lack dense internal geometries of
 328 high fidelity. Despite this limitation, our method is able to learn a general model which
 329 is capable of generating shapes with internal structures given noisy real-world raw data.



Fig. 6: **Generation Comparison:** From left to right (Diffusion [27], Point Flow [56], Graph-CNN GAN [51], FullFormer (Ours)). Our model (with 16^3 latent space resolution) shows high-quality internal structure generation results compared to other mentioned models. It is apparent that other comparative models do not achieve discernable internal structures in generation results. All point clouds in this figure are sampled to 2048 points.

330 4.6 Quantitative Results

331 In this section, we present a quantitative evaluation of our model’s performance in
 332 point cloud generation. The metrics discussed in section 4.4 are tabulated in Table
 333 2. Our method achieves state-of-the-art performance on all the metrics for the ‘Full
 334 Cars’ dataset, validating the capability of FullFormer in generating complete shapes
 335 with rich insides. High coverage and low JSD further demonstrate that generated mod-
 336 els exhibit high diversity which we also observe visually. Moreover, we achieve the
 337 best performance in MMD and coverage across all classes of cars, chairs, and planes
 338 of the ShapeNet dataset compared with other baselines. While it is true that FullFormer
 339 appears to achieve higher JSD values than PointFlow [56] and Diffusion [27] for the
 340 ShapeNet dataset, however qualitative results continue to show diversity in all the con-
 341 sidered datasets. Therefore the lower score of JSD for the ShapeNet dataset is hypoth-
 342 esized to be a cause of reference set selection.

343 4.7 Limitations

344 Unlike the high-fidelity achieved on outer shells, generated internal details exhibit lower
 345 quality. A sampling of the feature space limits the details of the shape’s geometry. Our
 346 model evaluation is also constrained by the scarcity of available shape datasets with rich
 347 internal structures. Furthermore, we used off-the-rack methods to mesh our dense point
 348 cloud results which degraded the quality of our results, as there is no direct algorithm
 349 to extract the surface of 3D shapes from unsigned distance fields. Especially on fine
 350 details and thin structures, the quality of generated shapes is not easy to assess from
 point clouds.

Table 2: We quantitatively compare the point cloud generation results of our method with GraphCNN-GAN [51], Diffusion [27] and PointFlow [56]. We report minimum matching distance (MMD), coverage score (COV), and Jenson and Shannon divergence (JSD) for comparison. We use Chamfer distance (CD) for MMD and COV calculations. MMD scores are multiplied by 10^3 and JSD are multiplied by 10^{-1} . Our proposed FullFormer improves consistently over all previous methods in terms of MMD and COV. It also improves over previous methods in terms of JSD on the Full Cars dataset.

Dataset	GraphCNN-GAN [51]			Diffusion [27]			PointFlow [56]			Ours (FullFormer)		
	MMD↓	COV↑	JSD↓	MMD↓	COV↑	JSD↓	MMD↓	COV↑	JSD↓	MMD↓	COV↑	JSD↓
ShapeNet Cars	3.18	16	4.67	1.4	17.7	2.21	1.28	29.67	3.16	1.13	29.72	2.29
ShapeNet Planes	1.1	31.09	1.75	0.98	36.73	0.65	1.41	35.87	1.06	0.92	37.37	0.83
ShapeNet Chairs	4.213	33.5	1.24	3.79	36.2	0.42	4.19	33.23	0.82	3.79	37	1.06
Full Cars	2.32	20	3.81	1.24	21.23	2.83	1.18	24.85	3.39	0.93	25.07	2.72

351

351

352 5 Conclusion

353 In this work, we present FullFormer, a novel two-stage generative model designed to
 354 generate 3D objects with intricate internal structures. Our approach employs a vector
 355 quantized autoencoder (VQUDF) to learn 3D shape geometry in the first stage and
 356 employ a latent transformer model in the second stage for shape generation. This la-
 357 tent transformer is trained autoregressively on indices of quantized shape embeddings
 358 learned by the VQUDF, making it computationally efficient. Consequently, the trained
 359 transformer can generate latent codes unconditionally. Generated codes are fed into a
 360 learned decoder (VQUDF) to output UDF representation from which 3D shapes are
 361 retrieved ensuring that generated shapes have details of internal structure and high-
 362 fidelity outer surface at arbitrary resolution. We further demonstrate superior qualita-
 363 tive and quantitative point cloud results compared to previous state-of-the-art methods.
 364 The ability to generate high-quality 3D shapes has implications across various domains,
 365 from computer graphics and virtual reality to manufacturing and design, paving the way
 366 for exciting future research in the field.

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367	References	367
368	1. Atzmon, M., Lipman, Y.: Sal: Sign agnostic learning of shapes from raw data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2565–2574 (2020)	368 369 370
371	2. Bengio, Y., Léonard, N., Courville, A.: Estimating or propagating gradients through stochastic neurons for conditional computation (2013). https://doi.org/10.48550/ARXIV.1308.3432 , https://arxiv.org/abs/1308.3432	371 372 373
374	3. Cai, R., Yang, G., Averbuch-Elor, H., Hao, Z., Belongie, S., Snavely, N., Hariharan, B.: Learning gradient fields for shape generation. In: European Conference on Computer Vision. pp. 364–381. Springer (2020)	374 375 376
377	4. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: Shapenet: An information-rich 3d model repository (2015). https://doi.org/10.48550/ARXIV.1512.03012 , https://arxiv.org/abs/1512.03012	377 378 379 380
381	5. Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., Sutskever, I.: Generative pretraining from pixels. In: International Conference on Machine Learning. pp. 1691–1703. PMLR (2020)	381 382 383
384	6. Chen, N., Zhang, Y., Zen, H., Weiss, R.J., Norouzi, M., Chan, W.: Wavegrad: Estimating gradients for waveform generation. arXiv preprint arXiv:2009.00713 (2020)	384 385
386	7. Chen, P.H., Luo, Z.X., Huang, Z.K., Yang, C., Chen, K.W.: If-net: An illumination-invariant feature network (2020). https://doi.org/10.48550/ARXIV.2008.03897 , https://arxiv.org/abs/2008.03897	386 387 388
389	8. Chen, W., Lin, C., Li, W., Yang, B.: 3psdf: Three-pole signed distance function for learning surfaces with arbitrary topologies. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (June 2022)	389 390 391
392	9. Chen, X., Kingma, D.P., Salimans, T., Duan, Y., Dhariwal, P., Schulman, J., Sutskever, I., Abbeel, P.: Variational lossy autoencoder. arXiv preprint arXiv:1611.02731 (2016)	392 393
394	10. 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)	394 395 396
397	11. Cheng, A.C., Li, X., Liu, S., Sun, M., Yang, M.H.: Autoregressive 3d shape generation via canonical mapping. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III. pp. 89–104. Springer (2022)	397 398 399
400	12. Chibane, J., Pons-Moll, G.: Implicit feature networks for texture completion from partial 3d data. In: European Conference on Computer Vision. pp. 717–725. Springer (2020)	400 401
402	13. Chibane, J., Pons-Moll, G., et al.: Neural unsigned distance fields for implicit function learning. Advances in Neural Information Processing Systems 33 , 21638–21652 (2020)	402 403
404	14. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)	404 405 406
407	15. d’Ascoli, S., Touvron, H., Leavitt, M.L., Morcos, A.S., Biroli, G., Sagun, L.: Convit: Improving vision transformers with soft convolutional inductive biases. In: International Conference on Machine Learning. pp. 2286–2296. PMLR (2021)	407 408 409
410	16. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12873–12883 (2021)	410 411 412
413	17. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems 27 (2014)	413 414 415

- 416 18. Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M.: Pct: Point cloud trans- 416
417 former. Computational Visual Media **7**, 187–199 (2021) 417
418 19. Hertz, A., Perel, O., Giryes, R., Sorkine-Hornung, O., Cohen-Or, D.: Spaghetti: Editing 418
419 implicit shapes through part aware generation. ACM Trans. Graph. **41**(4) (jul 2022). 419
420 <https://doi.org/10.1145/3528223.3530084> 420
421 20. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in Neural 421
422 Information Processing Systems **33**, 6840–6851 (2020) 422
423 21. Hui, K.H., Li, R., Hu, J., Fu, C.W.: Neural wavelet-domain diffusion for 3d shape generation. 423
424 In: SIGGRAPH Asia 2022 Conference Papers. pp. 1–9 (2022) 424
425 22. Hyvärinen, A., Dayan, P.: Estimation of non-normalized statistical models by score match- 425
426 ing. Journal of Machine Learning Research **6**(4) (2005) 426
427 23. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint 427
428 arXiv:1312.6114 (2013) 428
429 24. Kleineberg, M., Fey, M., Weichert, F.: Adversarial generation of continuous implicit shape 429
430 representations. arXiv preprint arXiv:2002.00349 (2020) 430
431 25. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Te- 431
432 jani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a gen- 432
433 erative adversarial network. In: Proceedings of the IEEE conference on computer vision and 433
434 pattern recognition. pp. 4681–4690 (2017) 434
435 26. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction 435
436 algorithm. ACM siggraph computer graphics **21**(4), 163–169 (1987) 436
437 27. Luo, S., Hu, W.: Diffusion probabilistic models for 3d point cloud generation. In: Proceed- 437
438 ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2837– 438
439 2845 (2021) 439
440 28. Meng, X., Chen, W., Yang, B.: Neat: Learning neural implicit surfaces with arbitrary topo- 440
441 logies from multi-view images. In: Proceedings of the IEEE/CVF Conference on Computer 441
442 Vision and Pattern Recognition (CVPR). pp. 248–258 (June 2023) 442
443 29. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy 443
444 networks: Learning 3d reconstruction in function space (2018). 444
445 <https://doi.org/10.48550/ARXIV.1812.03828> 445
446 30. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: 446
447 Learning 3d reconstruction in function space. In: Proceedings of the IEEE/CVF conference 447
448 on computer vision and pattern recognition. pp. 4460–4470 (2019) 448
449 31. Mittal, P., Cheng, Y.C., Singh, M., Tulsiani, S.: AutoSDF: Shape priors for 3d completion, 449
450 reconstruction and generation. In: CVPR (2022) 450
451 32. Mnih, A., Gregor, K.: Neural variational inference and learning in belief networks. In: Inter- 451
452 national Conference on Machine Learning. pp. 1791–1799. PMLR (2014) 452
453 33. Niemeyer, M., Geiger, A.: Giraffe: Representing scenes as compositional generative neural 453
454 feature fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern 454
455 Recognition. pp. 11453–11464 (2021) 455
456 34. Oord, A.v.d., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks (2016). 456
457 <https://doi.org/10.48550/ARXIV.1601.06759>, <https://arxiv.org/abs/1601.06759> 457
458 35. Oord, A.v.d., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A., Kavukcuoglu, K.: Con- 458
459 ditional image generation with pixelcnn decoders. arXiv preprint arXiv:1606.05328 (2016) 459
460 36. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: 460
461 Learning continuous signed distance functions for shape representation (2019). 461
462 <https://doi.org/10.48550/ARXIV.1901.05103>, <https://arxiv.org/abs/1901.05103> 462
463 37. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning 463
464 continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF 464
465 conference on computer vision and pattern recognition. pp. 165–174 (2019) 465

- 466 38. Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., Tran, D.: Image
467 transformer. International Conference on Machine Learning pp. 4055–4064 (2018) 466
468 39. Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., Tran, D.: Image
469 transformer (2018). <https://doi.org/10.48550/ARXIV.1802.05751>, <https://arxiv.org/abs/1802.05751> 467
470 40. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z.,
471 Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Te-
472 jani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative
473 style, high-performance deep learning library. In: Advances in Neural Information Process-
474 ing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf> 475
476 41. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A.: Convolutional occupancy
477 networks. In: European Conference on Computer Vision, pp. 523–540. Springer (2020) 477
478 42. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d
479 classification and segmentation (2016). <https://doi.org/10.48550/ARXIV.1612.00593>, <https://arxiv.org/abs/1612.00593> 479
480 43. Razavi, A., van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with vq-
481 vae-2. Advances in neural information processing systems pp. 14866–14876 (2019) 480
482 44. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate in-
483 ference in deep generative models. In: Xing, E.P., Jebara, T. (eds.) Proceedings of the 31st
484 International Conference on Machine Learning. pp. 1278–1286. No. 2 in Proceedings of Ma-
485 chine Learning Research, PMLR, Beijing, China (22–24 Jun 2014), <https://proceedings.mlr.press/v32/rezende14.html> 485
486 45. Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D.J., Norouzi, M.: Image super-resolution
487 via iterative refinement. arXiv preprint arXiv:2104.07636 (2021) 486
488 46. Sarmad, M., Rusipini, L., Lindseth, F.: Photo-realistic continuous image super-resolution with
489 implicit neural networks and generative adversarial networks. In: Proceedings of the North-
490 ern Lights Deep Learning Workshop. vol. 3 (2022) 489
491 47. Schwarz, K., Liao, Y., Niemeyer, M., Geiger, A.: Graf: Generative radiance fields for 3d-
492 aware image synthesis. Advances in Neural Information Processing Systems **33**, 20154–
493 20166 (2020) 493
494 48. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representa-
495 tions with periodic activation functions. Advances in Neural Information Processing Systems
496 **33**, 7462–7473 (2020) 496
497 49. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution.
498 Advances in Neural Information Processing Systems **32** (2019) 498
499 50. Thanh-Tung, H., Tran, T.: Catastrophic forgetting and mode collapse in gans. In: 2020 Inter-
500 national Joint Conference on Neural Networks (IJCNN). pp. 1–10. IEEE (2020) 500
501 51. Valsesia, D., Fracastoro, G., Magli, E.: Learning localized generative models for 3d point
502 clouds via graph convolution. In: International conference on learning representations (2019) 502
503 52. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. Advances in
504 neural information processing systems **30** (2017) 504
505 53. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł.,
506 Polosukhin, I.: Attention is all you need. Advances in neural information processing systems
507 **30** (2017) 507
508 54. Xiang, P., Wen, X., Liu, Y.S., Cao, Y.P., Wan, P., Zheng, W., Han, Z.: Snowflakenet: Point
509 cloud completion by snowflake point deconvolution with skip-transformer. In: Proceedings
510 of the IEEE/CVF international conference on computer vision. pp. 5499–5509 (2021) 509
511 55. Yan, X., Lin, L., Mitra, N.J., Lischinski, D., Cohen-Or, D., Huang, H.: Shapeformer:
512 Transformer-based shape completion via sparse representation. In: Proceedings of the
513 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6239–6249 (2022) 513
514 515 516

- 517 56. Yang, G., Huang, X., Hao, Z., Liu, M.Y., Belongie, S., Hariharan, B.: Pointflow: 3d point
518 cloud generation with continuous normalizing flows. In: Proceedings of the IEEE/CVF Interna-
519 tional Conference on Computer Vision. pp. 4541–4550 (2019) 517
520 57. Ye, J., Chen, Y., Wang, N., Wang, X.: Gifs: Neural implicit function for general shape rep-
521 resentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
522 Recognition (CVPR). pp. 12829–12839 (June 2022) 518
523 58. Zeng, X., Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., Kreis, K.: Lion: Latent
524 point diffusion models for 3d shape generation. In: Advances in Neural Information Process-
525 ing Systems (NeurIPS) (2022) 519
526 59. Zhang, B., Nießner, M., Wonka, P.: 3DILG: Irregular latent grids for 3d generative modeling.
527 In: Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) Advances in Neural Information
528 Processing Systems (2022), <https://openreview.net/forum?id=RO0wSr3R7y> 520
529 60. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Proceedings of the
530 IEEE/CVF international conference on computer vision. pp. 16259–16268 (2021) 521
531 61. Zheng, X., Liu, Y., Wang, P., Tong, X.: Sdf-stylegan: Implicit sdf-based stylegan for 3d shape
532 generation. In: Computer Graphics Forum. vol. 41, pp. 52–63. Wiley Online Library (2022) 522