

FullFormer: Generating Shapes Inside Shapes

GCPR 2023 - Fast Review Track

Paper ID 111

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

Keywords: Implicit Generative Models · Unsigned Distance Field.

1 Introduction

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

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

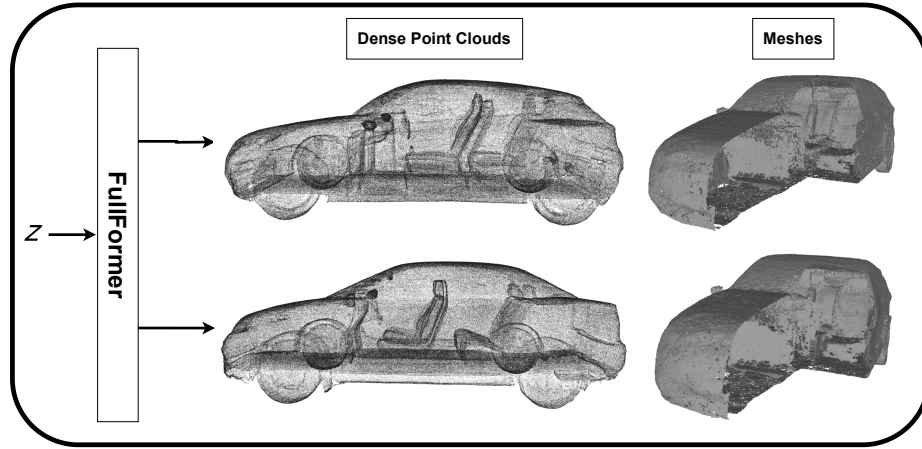


Fig. 1: This paper addresses generating 3D objects with rich internal geometric details.

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

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

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

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

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

2 Related Work

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

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

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

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

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

3 Method

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

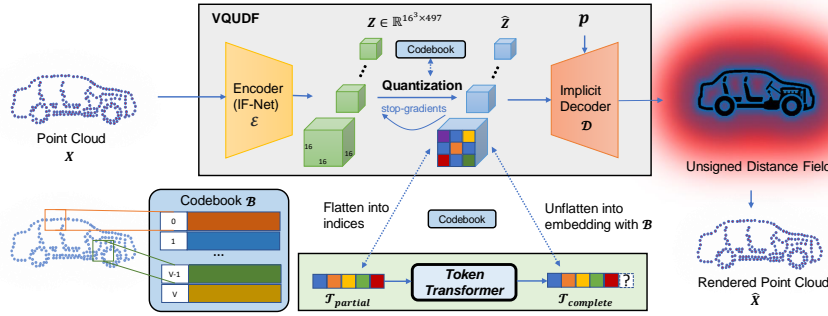


Fig. 2: **Approach:** Key ingredients of our pipeline are vector quantized autoencoder, unsigned distance field (UDF), and latent transformer. The first stage is learning VQUDF which is a vector quantized autoencoder model that takes voxelized point clouds as input to a CNN-based encoder and utilizes 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.

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

3.1 Sequential Encoding with VQUDF

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

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

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

Quantization: A discrete description of the world can aid learning by compressing information in many domains, such as language or images [52,32,9]. We posit that 3D models are no exception and can greatly benefit from discrete representations. In addition, to utilize the generative transformer model, the input shape is preferably a discrete *sequence*. Therefore, we employ vector quantization to transform the continuous latent feature representation \mathbf{Z} into a sequence of tokens \mathcal{T} using a learned codebook \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 code. Following a row-major ordering [16], each feature slice $\mathbf{z}_i \in \mathbf{Z}$ is clamped to the nearest code in the codebook \mathcal{B} using equation 1, fig. 2, which results in a quantized feature grid $\hat{\mathbf{Z}}$. A sequence of tokens \mathcal{T} is then defined as the ordered set of indices $(t_i) \forall i \in \{1, \dots, |\mathcal{T}|\}$.

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

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

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

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

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

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

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

3.2 Generating a Sequence of Latent Vectors

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

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 autoencoder 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 maximizing the log-likelihood of tokens in a randomly sampled sequence to represent the 3D shape $p(\mathcal{T})$:

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

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

4 Experiments

This section thoroughly evaluates our proposed approach on the standard object categories of *Cars*, *Planes*, and *Chairs* from ShapeNetCore [4] dataset. Additionally, we curate a new dataset named 'Full Cars', which constitutes a subset of the *Cars* category of the ShapeNetCore v2 dataset, on which we evaluate our proposed approach

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

4.1 Implementation Details

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

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

4.2 VQUDF Reconstruction Performance

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

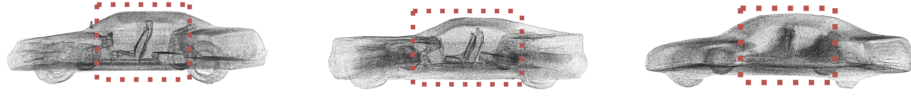


Fig. 3: **Reconstruction Results:** Our model reconstruction results with different latent space resolutions 64^3 , 16^3 and 8^3 respectively (left to right).

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

4.3 Baselines

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

4.4 Metrics

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

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

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

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

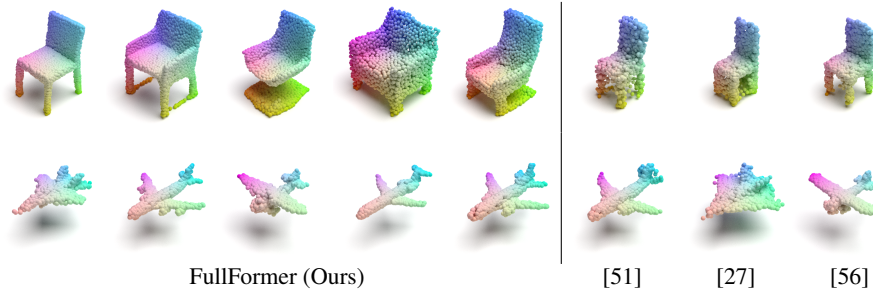


Fig. 4: **Outer Hull Generation:** Our models show high-quality point cloud generation results when trained on object categories of chairs, aeroplanes of ShapeNet dataset and visually improve over previous methods such as GraphCNN-GAN [51], Diffusion [27] and PointFlow [56].

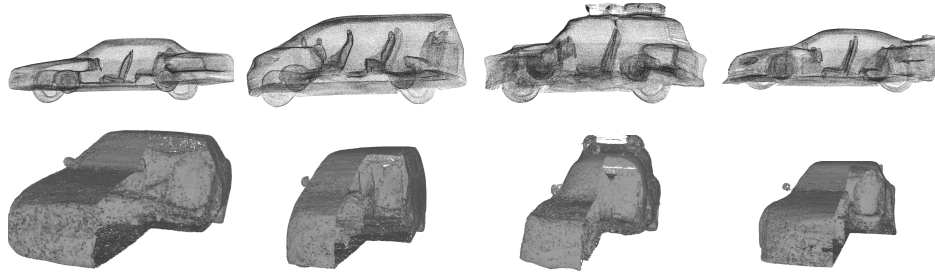


Fig. 5: **Generation:** Diverse generation results from our FullFormer model on the Full Cars dataset with internal structures. The high degree of detail of generated shapes is clearly visible in the dense point clouds. Note that, not only seats specific to car type, but also minute details such as steering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

4.5 Qualitative Results

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

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

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



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 330

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

4.7 Limitations

Unlike the high-fidelity achieved on outer shells, generated internal details exhibit lower quality. A sampling of the feature space limits the details of the shape’s geometry. Our model evaluation is also constrained by the scarcity of available shape datasets with rich internal structures. Furthermore, we used off-the-rack methods to mesh our dense point cloud results which degraded the quality of our results, as there is no direct algorithm to extract the surface of 3D shapes from unsigned distance fields. Especially on fine 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 Jensen 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 <i>Cars</i>	3.18	16	4.67	1.4	17.7	2.21	1.28	29.67	3.16	1.13	29.72	2.29
ShapeNet <i>Planes</i>	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 <i>Chairs</i>	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

5 Conclusion

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

References

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)
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>
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)
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>
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)
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)
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>
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)
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)
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)
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)
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)
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)
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)
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)
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)
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)

18. Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M.: Pct: Point cloud trans-
former. *Computational Visual Media* **7**, 187–199 (2021)
19. Hertz, A., Perel, O., Giryès, R., Sorkine-Hornung, O., Cohen-Or, D.: Spaghetti: Edit-
ing implicit shapes through part aware generation. *ACM Trans. Graph.* **41**(4) (jul 2022).
<https://doi.org/10.1145/3528223.3530084>, <https://doi.org/10.1145/3528223.3530084>
20. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. *Advances in Neural
Information Processing Systems* **33**, 6840–6851 (2020)
21. 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)
22. Hyvärinen, A., Dayan, P.: Estimation of non-normalized statistical models by score match-
ing. *Journal of Machine Learning Research* **6**(4) (2005)
23. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. *arXiv preprint*
arXiv:1312.6114 (2013)
24. Kleineberg, M., Fey, M., Weichert, F.: Adversarial generation of continuous implicit shape
representations. *arXiv preprint arXiv:2002.00349* (2020)
25. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Te-
jani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a gener-
ative adversarial network. In: *Proceedings of the IEEE conference on computer vision and
pattern recognition*. pp. 4681–4690 (2017)
26. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction
algorithm. *ACM siggraph computer graphics* **21**(4), 163–169 (1987)
27. Luo, S., Hu, W.: Diffusion probabilistic models for 3d point cloud generation. In: *Proceed-
ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 2837–
2845 (2021)
28. Meng, X., Chen, W., Yang, B.: Neat: Learning neural implicit surfaces with arbitrary topolo-
gies from multi-view images. In: *Proceedings of the IEEE/CVF Conference on Computer
Vision and Pattern Recognition (CVPR)*. pp. 248–258 (June 2023)
29. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Oc-
cupancy networks: Learning 3d reconstruction in function space (2018).
<https://doi.org/10.48550/ARXIV.1812.03828>, <https://arxiv.org/abs/1812.03828>
30. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks:
Learning 3d reconstruction in function space. In: *Proceedings of the IEEE/CVF conference
on computer vision and pattern recognition*. pp. 4460–4470 (2019)
31. Mittal, P., Cheng, Y.C., Singh, M., Tulsiani, S.: AutoSDF: Shape priors for 3d completion,
reconstruction and generation. In: *CVPR* (2022)
32. Mnih, A., Gregor, K.: Neural variational inference and learning in belief networks. In: *Inter-
national Conference on Machine Learning*. pp. 1791–1799. PMLR (2014)
33. Niemeyer, M., Geiger, A.: Giraffe: Representing scenes as compositional generative neural
feature fields. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
Recognition*. pp. 11453–11464 (2021)
34. Oord, A.v.d., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks (2016).
<https://doi.org/10.48550/ARXIV.1601.06759>, <https://arxiv.org/abs/1601.06759>
35. Oord, A.v.d., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A., Kavukcuoglu, K.: Con-
ditional image generation with pixelcnn decoders. *arXiv preprint arXiv:1606.05328* (2016)
36. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: DeepSDF:
Learning continuous signed distance functions for shape representation (2019).
<https://doi.org/10.48550/ARXIV.1901.05103>, <https://arxiv.org/abs/1901.05103>
37. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: DeepSDF: Learning contin-
uous signed distance functions for shape representation. In: *Proceedings of the IEEE/CVF
conference on computer vision and pattern recognition*. pp. 165–174 (2019)

38. Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, Ł., Shazeer, N., Ku, A., Tran, D.: Image transformer. *International Conference on Machine Learning* pp. 4055–4064 (2018)
39. Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, Ł., Shazeer, N., Ku, A., Tran, D.: Image transformer (2018). <https://doi.org/10.48550/ARXIV.1802.05751>, <https://arxiv.org/abs/1802.05751>
40. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: *Advances in Neural Information Processing 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>
41. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A.: Convolutional occupancy networks. In: *European Conference on Computer Vision*. pp. 523–540. Springer (2020)
42. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation (2016). <https://doi.org/10.48550/ARXIV.1612.00593>, <https://arxiv.org/abs/1612.00593>
43. Razavi, A., van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems* pp. 14866–14876 (2019)
44. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. In: Xing, E.P., Jebara, T. (eds.) *Proceedings of the 31st International Conference on Machine Learning*. pp. 1278–1286. No. 2 in *Proceedings of Machine Learning Research*, PMLR, Beijing, China (22–24 Jun 2014), <https://proceedings.mlr.press/v32/rezende14.html>
45. Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D.J., Norouzi, M.: Image super-resolution via iterative refinement. *arXiv preprint arXiv:2104.07636* (2021)
46. Sarmad, M., Ruspini, L., Lindseth, F.: Photo-realistic continuous image super-resolution with implicit neural networks and generative adversarial networks. In: *Proceedings of the Northern Lights Deep Learning Workshop*. vol. 3 (2022)
47. Schwarz, K., Liao, Y., Niemeyer, M., Geiger, A.: Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems* **33**, 20154–20166 (2020)
48. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems* **33**, 7462–7473 (2020)
49. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems* **32** (2019)
50. Thanh-Tung, H., Tran, T.: Catastrophic forgetting and mode collapse in gans. In: *2020 International Joint Conference on Neural Networks (IJCNN)*. pp. 1–10. IEEE (2020)
51. Valsesia, D., Fracastoro, G., Magli, E.: Learning localized generative models for 3d point clouds via graph convolution. In: *International conference on learning representations* (2019)
52. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. *Advances in neural information processing systems* **30** (2017)
53. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. *Advances in neural information processing systems* **30** (2017)
54. Xiang, P., Wen, X., Liu, Y.S., Cao, Y.P., Wan, P., Zheng, W., Han, Z.: Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 5499–5509 (2021)
55. 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)

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